

fintech  OS

# FintechOS Studio 24.4

## User Guide

# TOC

FintechOS Studio .....	42
Dex .....	45
Dex Context Management .....	45
Chat History .....	46
Evolutionary Data Core .....	48
Data Models .....	48
Entities .....	48
Attributes .....	48
Relationships .....	49
Extensibility .....	49
Building with Data at the Core .....	50
Data Integration .....	50
Data Model Designer .....	51
Access the Data Model Designer .....	51
Entities Panel .....	51
Toolbar .....	52
Work Area Panel .....	52
1 Add Data Model .....	53
2 Add Entities to Data Model .....	54
3 Define Relationships .....	54
Load Data Models .....	55
Edit Data Model Details .....	56
Edit Data Models .....	56
Delete Data Models .....	57
Data Model Explorer .....	57

Business Entities .....	58
Types of entities .....	58
View existing business entities .....	59
Create Entities .....	60
Edit Entities .....	69
Delete Entities .....	70
Referential Integrity Check .....	70
Attributes .....	71
System-generated attributes .....	72
Types of Attributes .....	72
Add Attributes .....	85
Reorder Entity Attributes .....	88
Add Option Set Attributes .....	89
STEP 1. Define the option set (picklist) .....	89
STEP 2. Add new option set items .....	89
Change the items order .....	92
Edit option set items .....	92
Delete Option Set Items .....	92
STEP 3. Define the option set attribute .....	92
Entity Unique Constraints .....	94
Create a Unique Constraint for an Entity .....	94
Add Unique Constraint Attributes .....	95
Enable/Disable a Unique Constraint .....	96
Entity Relationships .....	97
1:N Entity Relationships .....	97
Introduction .....	97
Create a one-to-many relationship .....	98
N:N Entity Relationships .....	100
Introduction .....	100
Create a many-to-many relationship .....	101

Advanced Entity Find .....	105
1 Select the attributes you wish to display .....	105
2 Apply filtering conditions .....	106
3 Order results .....	110
4 Validate the Fetch Results .....	111
Data Views .....	112
View basic action handlers .....	112
Create and Design Views .....	114
Introduction .....	114
Add Views .....	114
Provide View General Information .....	115
Define View Data .....	115
1 Fetch Entity Data .....	116
Choose Entity Links .....	117
Apply Filtering Conditions .....	120
Validate the Fetch Results .....	126
2 Set the default sorting of the view records .....	131
3 Define the View Columns .....	131
Generate View Columns .....	133
The view has no fetch data .....	133
The view has fetch data .....	134
Inline Editing for View Records .....	134
Create Views using Cell Template .....	137
Creating views default fetch .....	137
Creating views with custom fetch .....	138
Customize the Delete Confirmation .....	140
Show Loading Panel .....	142
Data Forms .....	142
Create an Entity Form .....	144
Edit an Entity Form .....	147

General .....	147
UI .....	147
Steps .....	149
Field Options .....	149
Filtered Fields .....	150
Advanced .....	150
Security Roles .....	150
Transient Data Entities .....	150
Create transient data entities .....	150
Define the automation script for load .....	151
Examples .....	152
Define the automation script for save .....	155
Extend platform data entities with transient data entities .....	156
Step 1. Create a transient data entity extension .....	156
Step 2. Add virtual attributes (only for transient data entities with single instance outputs) .....	157
Step 3. Bind entity attributes to the automation script for load input parameters .....	158
Step 4. Bind entity attributes to the automation script for save input parameters .....	159
Display transient data entity attributes in form driven flows .....	160
Display transient data entity attributes for single instance outputs .....	160
Display transient data entity attributes for collection outputs .....	162
Sample API Calls .....	163
Data Import Templates .....	165
Data Governance .....	168
Sensitive Context And Sensitive Data .....	169
Create a Sensitive Context .....	169
Sensitive Data .....	170
Data Anonymization Jobs And Requests .....	176

1 Set up the Anonymization Service .....	176
2 Set up the Anonymization Service Trigger .....	177
3 Restart the FintechOS Platform Job Server .....	177
Data Anonymization Requests .....	178
1 Create a Sensitive Request .....	178
2 Check the Returned Sensitive Data .....	180
3 Approve the Records for Data Anonymization .....	182
<b>Digital Experience .....</b>	<b>186</b>
Customer Journeys .....	186
Digital Journeys .....	186
Form Driven Flows .....	186
Custom Flows .....	187
Differences between the Form Driven Flow, Custom Flow and Digital Journey .....	187
UI Designer .....	187
Built-in Analytics .....	188
Digital Frontends .....	188
Customer Journeys .....	188
Digital Journeys .....	189
Create a Digital Journey .....	189
Journey Designer .....	191
Journey Settings .....	192
Add Journey Elements .....	193
Screen .....	193
Decision .....	194
Condition Editor .....	196
Integration .....	196
Service .....	199
Services Orchestration .....	201
Message .....	202

Document .....	203
Task .....	203
Asynchronous Tasks .....	204
Stage .....	204
Connect To .....	205
Move Journey Elements .....	205
Advanced Interface .....	205
Data Domain .....	206
Technical Variables .....	206
Data Entities .....	206
Dictionaries .....	207
Service Orchestration .....	207
Edit Specifications and Generate the FSD .....	207
Publish the Journey .....	208
Preview the Journey .....	209
General .....	209
Digital Journey Map .....	210
Add a flow to the map .....	211
Edit a step .....	211
Built-in Analytics .....	212
Form Driven Flows .....	221
Prerequisite .....	221
Create a form driven flow .....	222
3 Design the flow UI .....	227
4 Divide the flow in steps .....	228
Define access to the flow .....	229
Add Code to the Flow .....	229
Clone a form driven flow .....	229
Add and Configure Steps .....	231
1 Add a step .....	231

2 Design the step layout .....	233
3 Set advanced flow control rules .....	235
4 Configure event handlers .....	236
5 Provide the code to be executed after the step is generated (optional) .....	236
6 Define who has access to the step .....	236
7 Actions .....	236
Custom Processor Steps .....	238
Action Steps .....	240
Flow Control .....	241
Flow Execution Sequence .....	241
Override the Default Navigation .....	242
Apply Form Actions .....	243
Apply Conditional Navigation Rules .....	243
Checking with Custom Processor .....	246
Flow Map .....	247
Configure Field Options .....	248
Customize Form Fields .....	249
Configure Field Rules .....	257
1 Add field for action .....	257
2 Add field for condition .....	257
Form Actions .....	258
Create a form action .....	259
Available form action commands .....	260
Add an action to a specific step in a Form Driven Flow .....	261
Attach an endpoint in Form Action .....	262
Defining Action Groups .....	263
Prerequisite .....	263
1 Add action group .....	263
2 Add endpoints .....	264

Hide the action button .....	265
Defining Filtered Fields .....	266
Set up a Filtered Field in a Data Form, Form Driven Flow, or App Data Form .....	266
Filtered Fields on Editable Views .....	268
Header Items .....	269
Add a new header item .....	269
Link Labels to Attributes (deprecated) .....	270
Link labels to attributes using the HTML editor .....	271
Link labels to attributes using the Source code .....	271
Display View from Another Entity .....	272
Prerequisites .....	272
Display a view from another entity .....	272
Filter the view results .....	275
Pass default value .....	275
Refresh the view .....	275
Render Custom Data Extensions .....	276
Example .....	277
Create Custom Search Forms .....	278
Form Driven Mock-up Flows .....	280
Create a form driven mock-up flow .....	281
Convert a form driven mock-up flow into a regular form driven flow .....	282
Custom Flows .....	283
Create a Custom Flow .....	283
1 Provide custom flow general information .....	283
2 Design the custom flow layout .....	284
3 Define the custom flow .....	285
4 Define who has access to the journey .....	287
5 Save the journey .....	287
6 Display the Custom Flow on the Portal .....	287

Create Custom Controls .....	289
Create Custom Controls Using DevExtreme Widgets .....	289
Display Tabular Data Using DevExtreme Widgets .....	290
Modify Controls' Advanced Properties .....	292
Create custom controls using JQuery .....	293
Custom UI .....	293
Configuration .....	294
Create a Custom UI .....	294
Clone a Custom UI .....	296
Associate a Custom UI with a Form/Flow/Step .....	296
Events flow execution .....	298
Custom UI on Custom Flows (B2C) .....	299
Custom UI on B2C journeys .....	300
Custom UI on Automation Blocks .....	300
Custom UI on the entire UI (B2C) .....	300
Scenarios .....	300
Code Execution Sequence .....	301
Data Forms .....	302
Data Form Steps .....	303
Form Driven Flows .....	304
Form Driven Flow Steps .....	305
Good Coding Practices .....	306
Use Case Scenarios .....	307
App Data Forms .....	308
App Data Form Steps .....	309
Custom Flows .....	310
Custom UIs in Forms/Steps .....	310
Entity Views .....	311
Charts .....	312
Logic Blocks .....	313

Add, Rename, Find, and Delete Logic Blocks .....	313
Configure a Logic Block's Executable Logic .....	313
Configure a Logic Block's Trigger(s) .....	319
Preview the Logic Block's Code .....	319
Review the Logic Block Triggers .....	320
Digital Journey Context .....	320
Apply flow control rules only for specific digital journeys .....	321
UI Designer .....	323
Adding UI elements to the user interface .....	323
Moving and deleting UI elements .....	323
Where can you access the UI Designer .....	324
Design the UI layout with containers .....	325
Collapsible Containers .....	327
Add attributes to the UI .....	327
Custom UI Controls .....	331
Custom UI Controls for Boolean Attributes (Non-Mockup) .....	331
Custom UI Controls for Numeric and Whole Number Attributes (Non-Mockup) .....	332
Custom UI Controls for Option Set Attributes (Non-Mockup) .....	333
Add relations to the UI .....	333
Add transient data entities to the UI .....	337
Add views to the UI .....	340
Use predefined UI templates .....	342
Use UI components (headings, paragraphs, charts, widgets, buttons) .....	343
Headings .....	344
Paragraphs .....	345
Widgets .....	345
Charts .....	346
Buttons .....	347
Code Snippets Support .....	354

Code Snippets Placeholders .....	355
Navigating placeholders .....	355
Replacing placeholders .....	355
Deactivating placeholders .....	356
Series of placeholders .....	356
Nested code snippets .....	356
Examples of code snippets .....	356
Examples of code snippets for attributes and relations. ....	357
Code Snippets .....	358
Code Snippets Placeholders .....	358
Navigating placeholders .....	359
Replacing placeholders .....	359
Deactivating placeholders .....	359
Series of placeholders .....	360
Nested code snippets .....	360
Code snippets for entities and attributes .....	360
Digital Frontends .....	361
Dashboards .....	362
1 Create a Dashboard .....	362
2 Add Widgets to Dashboards .....	363
Customize Widgets .....	364
Resize Widgets .....	367
Move Widgets .....	367
Delete Widgets .....	368
3 Attach Security Roles to Dashboards .....	368
4 Add Dashboards to Portal Profiles .....	368
Widgets .....	370
Menu Items .....	371
Create Menu Items .....	371
Manage Menu Items in Menu Sections .....	373

Custom Themes .....	373
Create a Custom Theme .....	373
Apply the Custom Theme to a FintechOS Portal Instance .....	376
Apply a Custom CSS Style Sheet to the Portal UI .....	376
Portal Profiles .....	377
Create a Portal Profile .....	377
Assign a Portal Profile to a FintechOS Portal Instance .....	380
B2C Portals .....	381
Expose a Digital Journey to Anonymous Users .....	381
Activate B2C Portals .....	384
Initiate an Anonymous User Journey .....	384
Request .....	384
Response .....	385
Resume an Anonymous User Journey .....	385
Request .....	385
Response .....	385
Initiate/Resume Anonymous User Journeys with Digital Journey APIs .....	386
Resume External Journeys in an Anonymous B2C Environment .....	386
Use Custom Parameters in Anonymous User Journeys .....	386
Retrieve Custom Parameters in Anonymous User Journeys .....	387
Manage Style Sheets for Anonymous User Journeys .....	387
Login Screen Themes .....	388
Edit the Theme .....	388
Update and Upgrade .....	389
Anonymous Frontends (deprecated) .....	391
Anonymous Frontends Security .....	391
Set Up a B2C Environment .....	392
1 Install the FTOS Reverse Proxy .....	394

2 Configure the FTOS Reverse Proxy .....	394
3 Allow Access to the Journey Through the Reverse Proxy .....	396
4 Override the Default Save Mechanism .....	397
Enable Caching on the FTOS Reverse Proxy (optional) .....	397
Customize the Styles Sheets (optional) .....	398
Customize the Language (optional) .....	398
Reset an Anonymous Frontend Session .....	398
Override the Default Save Mechanism (deprecated) .....	399
1 Create an On-Demand Automation Script .....	399
2 Create an Endpoint and Attach the Script to it .....	400
3 Call the Endpoint on the Form Driven Flow .....	401
Serving User Journeys in a Specific Language (deprecated) .....	401
Digital Experience Portals .....	402
Using Custom Icons .....	404
What files do you need? .....	404
Files Location .....	407
Use custom icons .....	408
Setting Sticky Header .....	408
Keyboard Shortcuts .....	408
General keyboard shortcuts .....	409
Shortcuts for date/ date time fields .....	409
Shortcuts for drop-down fields .....	410
Shortcuts for radio buttons .....	410
Shortcuts for check boxes .....	410
<b>Automation Blocks .....</b>	<b>411</b>
Business Workflows .....	411
Business Formulas .....	412
Scheduled Jobs .....	412
Digital Documents .....	413

Scheduled Jobs .....	413
1 Add a scheduled job .....	413
2 Add schedule services .....	416
3 Set the execution order .....	416
Assign Job Server User .....	418
Business Workflows .....	418
Set Up Business Workflows .....	419
Create Business Workflows .....	419
Add statuses from the Designer .....	421
Add statuses from the Configurations tab .....	422
Add a transition from the Designer tab .....	423
Add a transition from the Configurations tab .....	425
Select business workflow on entity .....	427
Manage Business Workflows .....	428
Edit workflow statuses .....	429
Edit workflow transitions .....	429
How workflow transitions are processed in the database .....	429
Delete workflow statuses and transitions .....	429
Configure Business Workflows .....	430
Edit Workflow Transition Configuration .....	438
Define User Competences .....	442
1 Define User Competence Types .....	442
2 Define User Competences .....	443
Define Workflow Transition Approvals .....	445
Approve Workflow Transitions .....	447
Analytics .....	449
Business Formulas .....	450
Formula Inputs .....	451
Add Arguments to a Formula Input .....	452

Clone Formula Inputs .....	455
Define Formula Expressions .....	455
Add steps to a formula .....	457
Test Your Formula .....	461
Formula Versioning .....	463
Activate a Formula .....	463
Clone a Formula .....	464
Create a New Formula Version Draft .....	465
Activate a Formula Version Draft .....	466
Formula Editor .....	469
Syntax .....	470
Formula Arguments .....	471
Dynamic Collections Definitions .....	472
IntelliSense Support .....	473
Built-in Functions .....	473
For Simple types .....	473
For Collection Types .....	482
For Simple Collections Types .....	490
Formula Built-in Excel Functions .....	496
Formula Built-in DateTime Functions .....	500
Data Set Calls .....	512
Data Sets .....	513
Create a Data Set .....	513
Define Data Set Discriminants (non Single Value data sets) .....	515
Add Data Set Values (single value data sets) .....	516
Add Data Set Values (non Single Value data sets) .....	516
Data Set Versioning .....	518
Activate a Data Set .....	518
Create a New Data Set Version Draft .....	518
Activate a Data Set Version Draft .....	519

Clone Data Sets .....	519
Formula Parameter Mapping .....	520
Use Business Formulas in a Digital Journey .....	521
Call Formulas in a form-driven flow (no code) .....	521
Track form driven flows that are using a specific formula .....	522
Call formulas using server side scripts .....	523
Export a Formula or Data Set .....	523
<b>Products .....</b>	<b>526</b>
Product Templates .....	526
Select a Product Template .....	527
Use Dex to Select a Product Template .....	527
Built-In Product Templates .....	528
Personal Buy Now Pay Later .....	528
Main Info .....	528
Interest .....	530
Fees .....	533
Insurances .....	534
Repayment .....	535
Discounts .....	537
Underwriting .....	538
Documents .....	539
Service .....	541
Personal Current Account .....	543
Main Info .....	544
Interest .....	544
Fees .....	545
Discounts .....	546
Underwriting .....	547
Documents .....	548
Personal Debit Card .....	551

Main Info .....	552
Interest .....	552
Fees .....	553
Discounts .....	554
Underwriting .....	555
Documents .....	556
Personal Credit Card .....	558
Main Info .....	558
Interest .....	560
Fees .....	563
Repayment .....	564
Discounts .....	566
Underwriting .....	567
Documents .....	568
Service .....	570
Personal Deposits .....	572
Main Info .....	572
Interest .....	573
Fees .....	574
Payment .....	575
Bonuses .....	576
Discounts .....	577
Underwriting .....	577
Documents .....	579
Personal Secured Loan .....	581
Main Info .....	581
Interest .....	584
Fees .....	587
Insurances .....	588
Repayment .....	589

Discounts .....	591
Collateral .....	592
Underwriting .....	593
Documents .....	594
Service .....	596
Personal Bike Insurance .....	596
Main Info .....	596
Coverages .....	598
Underwriting .....	600
Pricing .....	601
Plans .....	602
Override the Default Claim Rules .....	603
Add Dynamic Dimensions to an Insurance Plan .....	605
Payment .....	606
Documents .....	606
Service .....	607
Home Insurance .....	609
Main Info .....	610
Coverages .....	612
Underwriting .....	614
Pricing .....	615
Plans .....	616
Override the Default Claim Rules .....	617
Add Dynamic Dimensions to an Insurance Plan .....	619
Payment .....	620
Documents .....	620
Service .....	621
Individual Healthcare .....	626
Main Info .....	626
Coverages .....	628

Underwriting .....	630
Pricing .....	631
Plans .....	632
Override the Default Claim Rules .....	633
Add Dynamic Dimensions to an Insurance Plan .....	635
Payment .....	636
Documents .....	636
Service .....	637
Personal Mortgage .....	639
Main Info .....	640
Interest .....	642
Fees .....	645
Insurances .....	646
Repayment .....	647
Discounts .....	649
Collateral .....	650
Underwriting .....	651
Documents .....	652
Service .....	654
Motor Insurance .....	657
Main Info .....	658
Coverages .....	660
Underwriting .....	662
Pricing .....	663
Plans .....	664
Override the Default Claim Rules .....	665
Add Dynamic Dimensions to an Insurance Plan .....	667
Payment .....	668
Documents .....	668
Service .....	669

Personal Overdraft .....	671
Main Info .....	671
Interest .....	672
Fees .....	673
Discounts .....	674
Underwriting .....	675
Documents .....	676
Pet Insurance .....	678
Main Info .....	678
Coverages .....	680
Underwriting .....	682
Pricing .....	683
Plans .....	684
Override the Default Claim Rules .....	685
Add Dynamic Dimensions to an Insurance Plan .....	687
Payment .....	688
Documents .....	688
Service .....	689
Personal Savings .....	691
Main Info .....	691
Interest .....	692
Fees .....	693
Discounts .....	694
Underwriting .....	695
Documents .....	696
Personal Split Payments .....	698
Main Info .....	698
Interest .....	701
Fees .....	704
Insurances .....	705

Repayment .....	706
Discounts .....	708
Underwriting .....	709
Documents .....	710
Service .....	712
Travel Insurance .....	712
Main Info .....	712
Coverages .....	714
Underwriting .....	716
Pricing .....	717
Plans .....	718
Override the Default Claim Rules .....	719
Add Dynamic Dimensions to an Insurance Plan .....	721
Payment .....	722
Documents .....	722
Service .....	723
Personal Unsecured Loan .....	725
Main Info .....	726
Interest .....	728
Fees .....	731
Insurances .....	732
Repayment .....	733
Discounts .....	735
Underwriting .....	736
Documents .....	737
Service .....	739
Personal Accidents .....	742
Main Info .....	743
Coverages .....	745
Underwriting .....	747

Pricing .....	748
Plans .....	749
Override the Default Claim Rules .....	750
Add Dynamic Dimensions to an Insurance Plan .....	752
Payment .....	753
Documents .....	753
Service .....	754
Commercial Business Current Account .....	756
Main Info .....	756
Interest .....	757
Fees .....	758
Discounts .....	759
Underwriting .....	760
Documents .....	761
Commercial Business Loan .....	764
Main Info .....	765
Interest .....	767
Fees .....	770
Insurances .....	771
Repayment .....	772
Discounts .....	774
Underwriting .....	775
Documents .....	776
Service .....	778
Commercial Credit Line .....	778
Main Info .....	778
Interest .....	780
Fees .....	783
Insurances .....	784
Repayment .....	785

Discounts .....	787
Collateral .....	788
Underwriting .....	789
Documents .....	790
Service .....	792
Commercial Property Insurance .....	792
Main Info .....	792
Coverages .....	794
Underwriting .....	796
Pricing .....	797
Plans .....	798
Override the Default Claim Rules .....	799
Add Dynamic Dimensions to an Insurance Plan .....	801
Payment .....	802
Documents .....	802
Service .....	803
Commercial Employer's Liability .....	808
Main Info .....	808
Coverages .....	810
Underwriting .....	812
Pricing .....	813
Plans .....	814
Override the Default Claim Rules .....	815
Add Dynamic Dimensions to an Insurance Plan .....	817
Payment .....	818
Documents .....	818
Service .....	819
Commercial CMR .....	821
Main Info .....	822
Coverages .....	824

Underwriting .....	826
Pricing .....	827
Plans .....	828
Override the Default Claim Rules .....	829
Add Dynamic Dimensions to an Insurance Plan .....	831
Payment .....	832
Documents .....	832
Service .....	833
Custom Product Templates .....	835
Create a Custom Product Template .....	835
Edit a Product Template .....	836
Customize Product Template Sections .....	836
Customize the Main Info Section .....	836
Customize Regular Sections .....	838
Product Template Life Cycle .....	839
Draft .....	840
Active .....	840
Inactive .....	841
Product Analytics (Portfolio) .....	841
Product Analytics Details .....	842
Product events .....	843
Raise events .....	844
Add historic data to Product Analytics .....	845
Prerequisite .....	845
1 Populate events Excel template .....	845
2 Upload historic data .....	846
Delete imported data .....	847
Product Life Cycle .....	848
Approve a Product .....	849
Approved Products Summary .....	849

Create a New Product Version .....	850
Delete a Product Draft .....	850
Close a Product .....	851
Offers .....	852
Create Offers .....	852
Products .....	853
Product Bundles .....	853
Content .....	853
Insurance Plans .....	855
Audience .....	858
Discounts .....	858
Offer Life Cycle .....	859
Duplicate or Delete Offers .....	860
Test Products and Offers .....	860
Test Product APIs Locally .....	864
Product Formulas .....	865
Basic Product Formulas .....	865
FixedInterest data set: .....	868
InterestByCustomerRating data set: .....	868
MerchantPortfolioYield data set: .....	868
Building Flexa coefficient data set: .....	871
Building NatCat coefficient data set: .....	871
Payment frequency data set: .....	871
Risk Factor data set: .....	874
Premium rate data set: .....	874
Advanced Product Formulas .....	875
Convert a Basic Product Formula into an Advanced Product Formula .....	875
Create an Advanced Product Formula from Scratch .....	876
Reuse a Product Formula Within the Same Product .....	876

Clone a Product Formula .....	877
<b>Product Data Sets .....</b>	<b>877</b>
One-Dimensional Data Sets .....	878
Two-Dimensional Data Sets .....	878
Multi-Dimensional Data Sets .....	879
How to Fill In a Product Data Set .....	880
<b>Product Settings .....</b>	<b>881</b>
Payment Schedule Type .....	881
Loan Periodicity .....	892
Create Loan Periodicity Records .....	893
Collateral Type .....	893
Banking Product Document .....	895
Insurance Peril .....	895
Add Perils .....	896
Insurance Document .....	897
Lexicon Term .....	897
Fee Types .....	900
Interest Indexes .....	902
Insured Objects .....	902
Underwriting Data Set Values .....	903
Plan Dimension .....	903
<b>Products APIs .....</b>	<b>904</b>
Web API Client Library .....	905
Server Automation Script Library .....	905
<b>Hyper-personalization Automation .....</b>	<b>906</b>
Features .....	907
Customer Personas .....	907
Add Customer Persona Data Model .....	908
Create Customer Personas .....	910

1 Insert a new customer persona record .....	910
2 Define the customer persona .....	910
3 Define audience segments .....	911
Preview your audience segment .....	912
Customer Personas Examples .....	912
Get Records for Entity .....	916
Define Aggregate Data .....	919
Set Tokens for Personalized Content .....	923
View Audiences that Use Current Customer Persona .....	924
View Campaign Stages Excluding Segment .....	925
Audiences Management .....	926
View Audiences .....	928
Edit audiences .....	928
Preview audiences .....	928
Create Audiences .....	929
1 Insert audience .....	929
2 Define the audience .....	929
3 Set up the audience .....	930
Audience Setup Examples .....	931
View Referenced Segments .....	931
View Campaigns Referencing Current Audience .....	931
<b>Servicing .....</b>	<b>933</b>
App Data Forms .....	933
Prerequisite .....	933
Create an App Data Form .....	934
Set the App Data Form Default Type .....	940
Define Who Has Access to the Form .....	941
Clone an App Data Form .....	941

For detailed procedures on how to do extensive configuration of app data forms, expand the below sections: .....	942
App Data Mock-up Forms .....	944
Add and Configure Steps .....	945
1 Add step .....	945
2 Design the step layout .....	946
3 Configure event handlers .....	948
4 Provide the code to be executed after the step is generated (optional) .....	948
5 Define who has access to the step .....	948
6 Save step .....	948
Define Form Actions .....	949
Create a Form Action .....	949
Available Form Action Commands .....	950
Attach an endpoint in Form Action .....	950
Define Action Groups .....	952
Prerequisite .....	952
1 Add action group .....	952
2 Add endpoints .....	952
Hide the action button .....	954
Define Filtered Fields .....	954
Back-Office Flows .....	955
Create a Back-Office Flow .....	955
Differences Between Back-Office Flows and Digital Journeys .....	955
Attach a Back-Office Flow to an App Data Form .....	956
Run a Back-Office Flow from an App Data Form .....	958
<b>Ecosystem Hub .....</b>	<b>959</b>
External Services via Service Pipes .....	959
Web API Client Libraries .....	959
External APIs .....	960

Web API Client Libraries .....	960
Create a Web API client library from an OpenAPI specification file .....	961
Use a Web API client library in server automation scripts .....	962
Add certificate support to WebApi client and WCF client .....	962
OpenAPI Domains .....	964
Create an OpenAPI Domain .....	965
Assign an Entity to an OpenAPI Domain .....	966
Assign an Endpoint to an OpenAPI Domain .....	966
Call External Services via Service Pipes .....	967
Set up the endpoint specifications in your Service Pipes project .....	967
Create routes and mappings directories .....	968
Generate the data model artifacts for your Service Pipes project .....	968
Create the Service Pipes (routes) .....	969
Use Cloud Configuration Manager properties in Service Pipes .....	969
Import the Endpoint Specifications File in a Web API Client Library .....	970
Create a Server Automation Script that Invokes the Web API Client Library .....	971
Additional Information .....	972
External APIs .....	972
Create, Configure, and Call External APIs .....	973
Create External APIs .....	973
Configure External APIs .....	973
Call External APIs .....	989
External API General Settings and Data Model .....	991
Data Model .....	991

Camel DSL Best Practices .....	992
Build Camel DSL Routing Scripts .....	992
Create Connectors Helper Classes .....	994
Write Unit Tests .....	996
Create a Connector .....	1000
Prerequisites .....	1000
Use the GitHub Template .....	1001
Create the Project Structure .....	1001
Project Configurations and Settings .....	1002
Project Object Model (POM) File .....	1002
Maven setting.xml File .....	1006
Build a Connector .....	1007
Debug a Connector .....	1008
Use Service Pipes Container Apps .....	1009
Prerequisites .....	1009
Create the Service Pipes Project and start it locally .....	1010
Run the Service Pipes Container Application .....	1013
Enable Debug Mode .....	1014
<b>Analytics .....</b>	<b>1015</b>
Data Source .....	1015
Reports .....	1015
Advanced Analytics .....	1016
Charts .....	1016
Data Sources .....	1016
Add a Data Source .....	1016
Use Stored Procedures .....	1017
Use Fetch Data .....	1017
Charts .....	1019

Add a chart .....	1019
Advanced Analytics .....	1026
Register App for Power BI .....	1027
Prerequisites .....	1027
How to find the tenant URL .....	1027
Register app for Power BI .....	1029
Embed Power BI Report .....	1031
Get the Power BI report ID .....	1031
Embed the Power BI report in FintechOS Studio .....	1031
Add report parameters (Embedding authentication mode only) ..	1033
Add Power BI Reports to Dashboards .....	1034
Prerequisites .....	1034
Add a Power BI report to a dashboard .....	1034
Add Power BI Reports to Digital Journeys .....	1035
Simple Grid Reports .....	1036
1 Add Report Parameters .....	1036
2 Add a Simple Grid Report .....	1037
3 Add Report Items .....	1038
4 Define Report Access Privileges .....	1039
Custom Reports .....	1039
1 Add a report .....	1039
For scope General or Entity, and type Custom Report: .....	1040
For scope Entity and type Document or Custom Report .....	1040
2 Add report items .....	1041
3 Insert Report Parameters .....	1042
4 Define who has access to the custom report .....	1042
Observability & Telemetry .....	1043
1 Authentication & Roles .....	1043
2 Dashboards .....	1046

Default dashboard .....	1046
Monitoring Metrics .....	1046
WebApps (for FintechOS components and Business Components) .....	1046
AzureInsights .....	1047
Database .....	1047
SqlQueryPerformance .....	1047
Business dashboard .....	1048
3 Search & Filtering .....	1048
4 Examples .....	1049
AzureInsights .....	1049
SqlQueryPerformanceMonitoring .....	1050
Database_Monitoring .....	1052
WebApps_Monitoring .....	1052
<b>Configuration Management .....</b>	<b>1053</b>
The Context Digital Asset .....	1053
Digital Assets .....	1055
Create Digital Assets .....	1055
Edit Digital Assets .....	1058
General .....	1058
Set the Context Digital Asset .....	1058
Unlock Sealed Digital Assets .....	1059
Configuration Items .....	1059
Manually Add Configuration Items to the Digital Asset .....	1060
Manually Remove Configuration Items from the Digital Asset ..	1063
Related Digital Solutions .....	1064
Data Import Files .....	1064
DB objects .....	1065
Products .....	1066
Custom Files .....	1067

Custom and Custom-on-demand Folders Structure .....	1068
Custom Fonts for Document Templates .....	1069
Dependencies .....	1071
Configuration Items Migration .....	1072
Run server scripts in the Advanced tab .....	1072
Delete Digital Assets .....	1073
Import Digital Assets .....	1073
Prerequisites .....	1074
Lock and Unlock Digital Assets .....	1074
How to Lock a Digital Asset .....	1075
How to Unlock a Digital Asset .....	1075
Branch Digital Assets .....	1075
Digital Solutions .....	1077
Create Digital Solutions .....	1077
Edit Digital Solutions .....	1078
Delete Digital Solutions .....	1078
Configuration Data Definitions .....	1079
Deployment .....	1085
Digital Solutions Packages .....	1086
Create Digital Solutions Packages .....	1086
Add security configuration to the digital solution package .....	1086
Configure the custom files mappings .....	1087
Include identity provider UI themes .....	1089
Export Digital Solutions Packages .....	1089
Import Digital Solutions Packages .....	1090
Delete a Digital Solution Package .....	1094
Delete a Digital Solution Package .....	1094
Deployment Scenarios .....	1094
Configuration Data Packages .....	1099

Export a Configuration Data Package .....	1100
Import a Configuration Data Package .....	1101
Deployment Package Logs .....	1102
Manage Locked Files .....	1104
<b>Advanced Developer Tools .....</b>	<b>1105</b>
Code Editor .....	1108
Open the Code Editor .....	1109
Code Editor Layout .....	1109
Explorer Structure .....	1110
FintechOS Explorer .....	1110
Manage and Edit Files .....	1112
Source Control .....	1113
Intellisense for SDK Methods .....	1114
Code Editor in Sandbox .....	1114
Code Editor Development Guide .....	1115
Initial Preparation .....	1116
Add a Change .....	1128
Integrate changes .....	1137
Build and deploy .....	1141
Advanced Code Editor (deprecated) .....	1142
Features .....	1143
Access the Advanced Code Editor .....	1143
General Layout .....	1143
Files Explorer .....	1144
Toolbar .....	1145
Search Nodes .....	1146
Search in Files .....	1146
Properties List .....	1146
Previewing HTML Files .....	1147

Code Editing Supporting Features .....	1148
HTML Syntax Guidelines .....	1148
Debug Files using File History .....	1149
<b>Business Service Components .....</b>	<b>1149</b>
Using SDK Methods in Modules .....	1150
Create Business Service Components .....	1152
Debug Components .....	1154
Generate Endpoints .....	1155
Import or Export Components .....	1156
<b>Client Script Libraries .....</b>	<b>1156</b>
Create a Client Script Library .....	1157
Clone a Client Script Library .....	1158
Import Client Script Libraries in your Client Side Code .....	1159
<b>Server Automation Scripts .....</b>	<b>1162</b>
Event Triggered Automation Scripts .....	1163
On-Demand Automation Scripts .....	1163
Create Event Triggered Automation Scripts .....	1163
Create On-demand Server Automation Scripts .....	1168
Call Actions .....	1171
XML Support .....	1173
Load XML from a String .....	1173
Catch an XML Load Error .....	1173
Run XPath Queries .....	1174
Run XPath Queries with Namespaces .....	1175
Node API Calls .....	1176
Fluent Queries .....	1177
Execute a Fluent Query .....	1178
Comparison Operators .....	1179
Logical Operators .....	1179

Inner Joins .....	1179
Left Joins .....	1180
Attribute Aliases (Projections) .....	1180
Where Clauses .....	1182
Aggregate Functions .....	1183
Work with Fluent Query Result Sets .....	1184
Map Result Sets to POJO Objects .....	1185
Fluent Query Caching .....	1186
Assign a Cache Key to a Fluent Query .....	1187
Create Asynchronous Processing Flows .....	1188
1 Create the Asynchronous Steps .....	1189
2 Configure the Message Queues and Flow in the Configuration Manager .....	1189
Call an Asynchronous Flow via API .....	1190
Call an Asynchronous Flow via Service Pipes (Deprecated) .....	1190
Debugging .....	1190
Debugging Server .....	1191
Debugging Server Prerequisites .....	1191
Use the Debugging Server .....	1191
Debugging Log .....	1197
Logging Destinations .....	1197
Logging Context .....	1197
Custom Log Context Properties .....	1198
Throw Exceptions .....	1199
JavaScript Exceptions .....	1199
Console Debugging .....	1199
JavaScript Engine Jint V3 .....	1201
Switching to Jint V3 .....	1201
Mandatory changes with Jint v3 .....	1201
Redefine the same variable multiple times in the same script ..	1201

Using exception message in try/catch .....	1202
JavaScript ES6 Support .....	1203
Switch between the Jint (ES5) and ClearScript (ES6) Scripting Engines .....	1204
ES6 Compatibility Impact .....	1204
The JavaScript Date Object Is Always Set to UTC .....	1204
Enforced ES6 JavaScript Syntax .....	1205
Processor Settings .....	1205
Numeric Attributes and Undefined Variables .....	1205
ftos.data.query.getAlias Case Sensitive Attribute Names .....	1206
Server Automation Script Libraries .....	1206
Create Automation Script Libraries .....	1207
Style Sheets .....	1209
Create a New Style Sheet .....	1209
Clone a Style Sheet .....	1209
Apply Style Sheets to Form Driven Flows (No-Code) .....	1211
Apply Style Sheets Using Code .....	1212
Limit Style Impact to Current Form .....	1212
Overwrite Variables .....	1213
Endpoints .....	1213
Create an endpoint .....	1213
Attach a security role to an endpoint .....	1215
Customize the Input Parameters .....	1216
Customize the Output Structure .....	1219
Plugin Assemblies (Deprecated) .....	1221
1 Add the Plugin Assembly .....	1221
2 Add the IEbsPlugin Plugin .....	1222
3 Add the UI Processor .....	1222
Sequencers .....	1222

Roll the Sequencer Back or Forward .....	1225
Retrieve a Sequence Number from the Sequencer ..	1226
Code Blocks .....	1227
Define Code Blocks Categories .....	1227
Add Code Blocks .....	1228
Use Code Blocks .....	1229
DB Tasks .....	1231
1 Add DB Tasks .....	1231
2 Add Security Roles to DB Tasks .....	1232
3 Execute DB Tasks .....	1232
Syntax: .....	1232
Request Parameters .....	1233
Returns .....	1233
Operation Authorization .....	1233
1 Prerequisites .....	1234
2 Details Automation Script .....	1234
3 Pass-Code Message and Authorization Page Templates .....	1235
Template Tokens .....	1235
Pass-Code Message Template Example .....	1236
Authorization Page Template Example .....	1237
4 Authorization and Rejection Automation Scripts ...	1237
5 Action Authorization Type .....	1237
Initiate an Operation Authorization Process .....	1240
<b>Security .....</b>	<b>1241</b>
Business Units .....	1243
Create Business Units .....	1243
Remove Business Units .....	1244
Synchronization in IDP .....	1244

Security Roles .....	1245
Default Security Roles .....	1245
1 Add the security role .....	1247
2 Assign security items to the security role .....	1248
3 Edit Security Roles .....	1250
System Users and Service Accounts .....	1251
Add System Users .....	1252
Change Password .....	1253
Create Service Accounts .....	1254
Edit System Users or Service Accounts .....	1255
Service Accounts in Digital Assets .....	1256
Recover Password (for users) .....	1257
Share Access to Uploaded Files .....	1258
<b>Admin Configuration .....</b>	<b>1260</b>
Email Templates .....	1260
Entity Cloning and Versioning .....	1261
How to Set Up Cloning and Versioning on an Entity .....	1262
1 Set Up the Entity Workflow .....	1262
2 Configure the Clone and Version Settings .....	1262
(Optional) Set Up Related Versioned Entities .....	1264
How to Clone an Entity Record .....	1265
How to Version an Entity Record .....	1265
Legacy Entity Versioning (Deprecated) .....	1266
Settings .....	1267
Custom Themes .....	1269
Localization .....	1269
Available Languages .....	1270
Add Languages .....	1270

Localization Resources .....	1273
Localize Language Resources .....	1273
Export-Import Localization Resources between FintechOS Studio Instances .....	1274
Localize Metadata .....	1274
Localize HTML Templates .....	1275
Localize HTML elements on data forms .....	1276
Localize from Metadata .....	1278
Localize Relationship Labels .....	1279
Localize Option Set Items .....	1280
Localize Views .....	1280
Client-side and Server-Side Localization .....	1281
Client-side .....	1281
Server-side .....	1284
System Parameters .....	1285
Default System Parameters .....	1286
System Parameter Groups .....	1293

# FintechOS Studio

## 24.4 User Guide

FintechOS Studio is an IDE which gives both citizen developers and software engineers the tools to build, customize, and extend digital solutions on top of the FintechOS Platform.

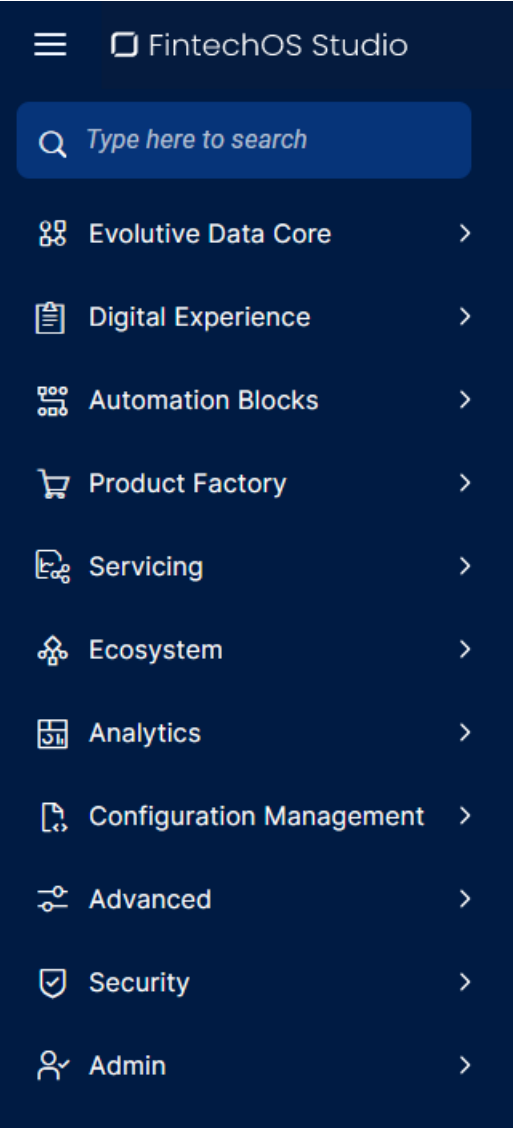
Using FintechOS Studio, you can configure all the components that make up a comprehensive digital experience, to create personalized products, digital journeys, or back-office applications.

- ["Evolutive Data Core" on page 48](#) contains tools for data modeling and data replication. You can create, browse, and edit data models by defining entities, attributes, and relationships. You can also extend the data model with data from third-party systems or other FintechOS Platform instances via data pipes and data import templates.
- ["Digital Experience" on page 186](#) allows the creation and configuration of digital journeys and of front end components such as dashboards, menus, themes, or portals.
- ["Automation Blocks" on page 411](#) are advanced pre-built features that you can customize and include in your digital experience. They cover frequent use cases, such as OCR extraction, identity validation, video streaming and collaboration, marketing campaigns management, electronic signatures, digital documents, etc.
- ["App Data Forms" on page 933](#) provides digital apps for middle and back-office operations that allow you to navigate freely between form tabs without the need of navigation buttons or a flow map.
- ["Ecosystem Hub" on page 959](#) facilitates connectivity with external resources, such as API endpoints from third parties.

- ["Analytics" on page 1015](#) provide business intelligence tools to aggregate and analyze data through charts, reports, and advanced business intelligence features.
- ["Configuration Management" on page 1053](#) allows you to control the ownership, versioning, deployment, and import/export of all components that fulfill a specific business need (such as a digital journey, a common data model, or an app data form).
- ["Server Automation Scripts" on page 1162](#) and other ["Advanced Developer Tools" on page 1105](#) provide developers with advanced coding tools such as scripts, libraries, sequencers, code blocks, database tasks, as well as a powerful code editor.
- ["Security" on page 1241](#) covers the organization structure, security roles, and management of both internal and external users.

In addition to the development tools described above, the ["Admin Configuration" on page 1260](#) provides a set of administrative features that allow you to manage the FintechOS Platform.

The above concepts are also reflected in how the FintechOS Studio's main menu is organized. As you explore the menu, you can use the search box at the top of the menu to quickly find the menu items you are looking for.



# Dex

Dex is an AI assistant which can provide answers to platform related questions, troubleshoot issues, suggest documentation resources, or jump-start your work by suggesting relevant templates for your ["Products" on page 526](#) or ["Digital Journeys" on page 189](#).

To use Dex, click the assistant button in the top right corner of FintechOS Studio and interact with the assistant through the chat-based interface using natural language. Each session maintains dialogue context across multiple exchanges, allowing for follow-up questions and complex logical flows.

To clear the conversation history and begin a new session, click the **New Chat** link at the bottom-right corner of the chat window.

## Dex Context Management

Each query is associated with a predefined context, which is automatically inferred by Dex in order to retrieve the most relevant data. If you want to restrict the conversation to a specific domain, select the desired context and click the **Lock Context** button at the bottom of the chat window. The available contexts are:

- Product - Find and retrieve suitable ["Product Templates" on page 526](#) based on your requirements.
- Journey - Find and retrieve suitable digital journey templates based on your requirements.
- Coding - Help with coding tasks, such as writing ["Server Automation Scripts" on page 1162](#), API calls, or [custom services](#).

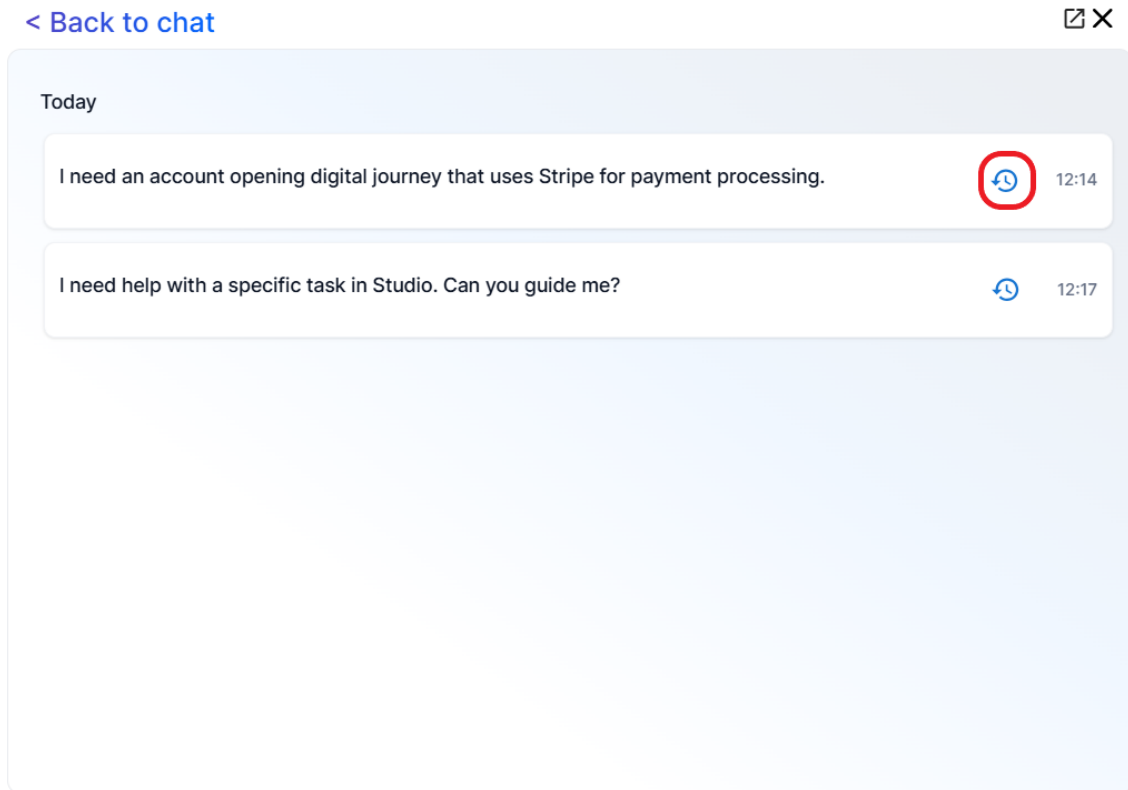
- **Navigation** - Locate and understand features in FintechOS Studio and FintechOS Portal by following step-by-step instructions (including menu paths and verified Studio UI deep-links) or referencing the official FintechOS documentation.
- **Banking** - Guidance on FintechOS Platform Core Banking capabilities.
- **Insurance** - Guidance on FintechOS Core Insurance capabilities.

**NOTE**

For complex queries, Dex may combine relevant information from different contexts into a single response (e.g., provide both navigation guidance and coding solutions in the same answer).

## Chat History

You can access your previous conversations by clicking the **History** link located at the bottom right corner of the Dex window. This opens a list of all past conversations. To view a specific conversation, click the clock icon next to it.



Once a conversation is open, you can browse its contents or continue the discussion from where you left off.

# Evolutionary Data Core

All business logic and presentation components of a software solution require access to a persistent data storage backbone in order to function. The Evolutionary Data Core provides a data persistence layer where the information processed by FintechOS digital solutions is stored and managed.

## Data Models

Defining the data model is typically the first step of developing FintechOS Platform digital solutions, as the business logic and the presentation layer of a digital solution need data in order to function. Data models identify what data is needed and how it should be organized. They also ensure the quality of data via naming conventions and default values. FintechOS Studio data modeling is based on the Entity - Relationship paradigm, which uses the concepts of **entity**, **attribute**, and **relationship** to represent data.

## Entities

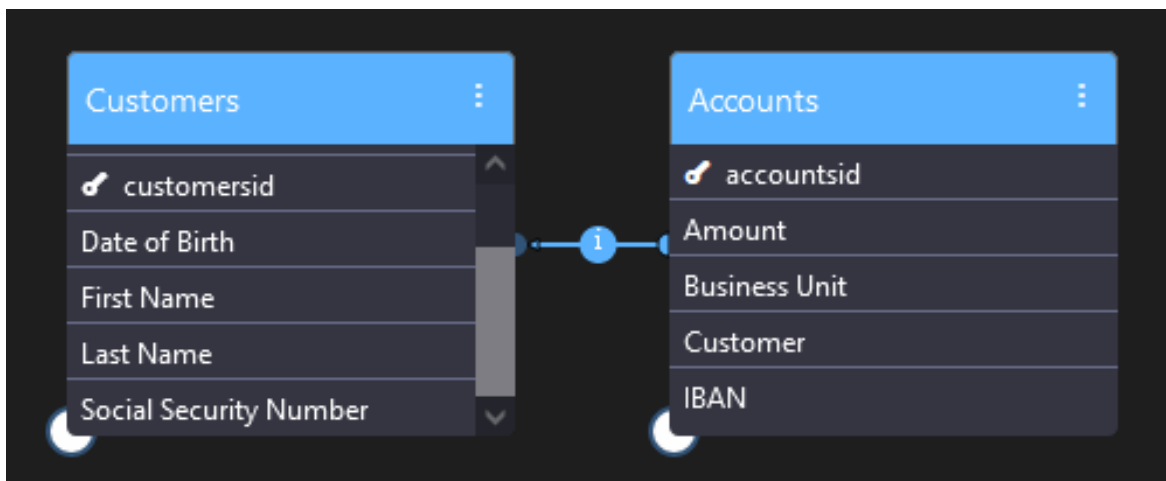
Entities are representations of the main physical or logical components that require record keeping (data storage). Typical examples include customers, invoices, employees, or products. For more info, see ["Business Entities" on page 58](#).

## Attributes

Attributes are used to model the specific properties of an entity that you want to record. For instance, for an entity representing customers, you might want to define attributes such as first name, last name, social security number, or date of birth. For more information, see ["Attributes" on page 71](#).

## Relationships

Relationships define links between entities. For instance, in an online banking digital solution, you may need to record data about your customers and their banking accounts. For this purpose, you will create two entities (one for customers and one for accounts) and a relationship between them representing which customer owns a specific account.



In FintechOS Studio, relationship modeling is implemented by a special type of attribute called **lookup**. In the example above, the Customer attribute in the Accounts entity is a lookup attribute which helps to define the relationship between the Accounts and the Customers entities. For more information, see "[Lookup](#)" on page 77.

## Extensibility

FintechOS Studio allows you to modify your data model as your digital solutions evolve. You can change the underlying data model of a digital solution in order to customize or integrate it, even while that digital solution is live. FintechOS Studio performs checks to ensure that you don't corrupt data or break any dependencies in the process. For instance, you will not be able to delete an entity that is referenced in a relationship or to add a required attribute to an entity that is populated with records (and therefore has no matching entries for that attribute in the legacy data).

# Building with Data at the Core

"Form Driven Flows" on page 221, which are the main building blocks of FintechOS Platform digital solutions, are each built on top of a dedicated data model. This allows a lot of the default business logic and UI rendering to be inferred from the data model itself and generated automatically, greatly reducing development time and improving the quality and reliability of your solution.

Based on the flow's data model, the form fields used to display entity attributes, the session storage, the user authentication and authorization, the logic for loading and saving records, etc. are generated automatically. You can, of course, further customize them, but most of the heavy lifting is done by the platform.

**NOTE**

You also have the option to use form driven mock-up flows which allow you to design a form driven flow without an underlying data model. This lets consultants and developers to quickly define the general layout of the user interface. Developers can then attach a data model to the mock-up, map entity attributes to the corresponding form fields, and work on any additional back-end configurations. For more information, see "Form Driven Mock-up Flows" on page 280.

## Data Integration

The Evolutive Data Core not only ensures the modeling of the data structure, but can also interact with data from legacy systems, processes, and data repositories, greatly extending the FintechOS Platform data persistence capabilities.

FintechOS Studio provides various options to import and interconnect data, such as: "Data Import Templates" on page 165, REST APIs, and "External APIs" on page 972s.

Feel free to explore our [FintechOS Academy](#) training courses.

This section covers the following topics:

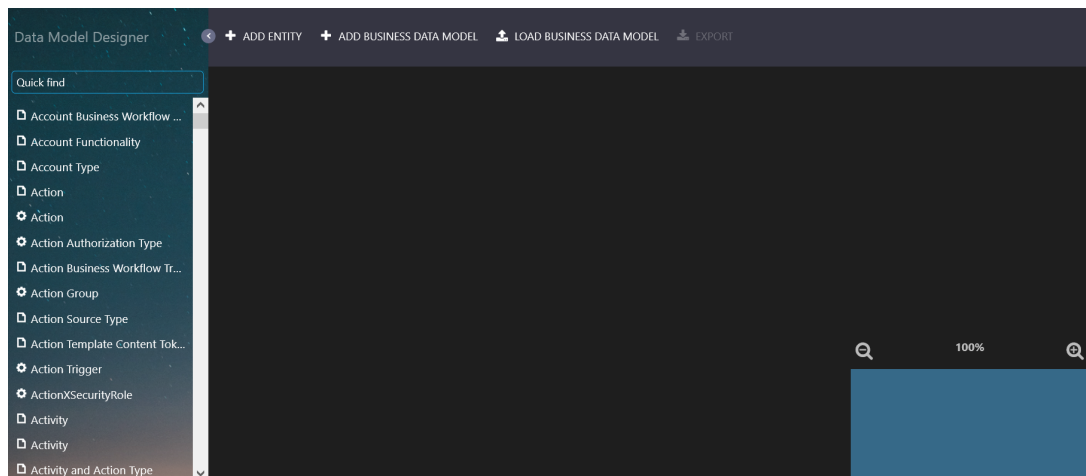
---

# Data Model Designer

Data Model Designer is a graphical tool that simplifies data modelling and increases user productivity when performing data modelling.

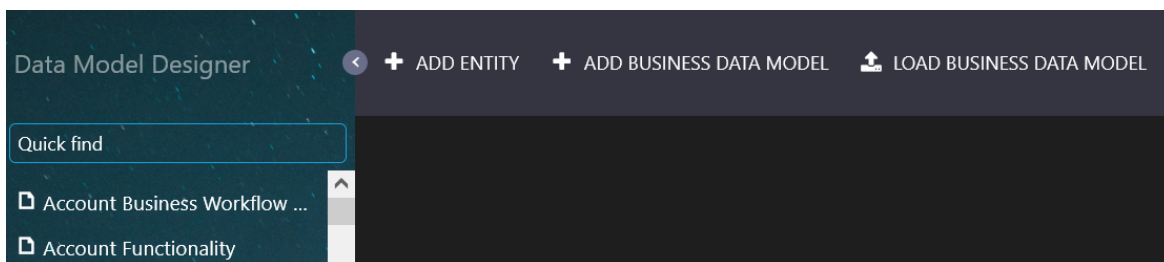
## Access the Data Model Designer

1. Go to **Evolutive Data Core > Data Model Designer**. The **Data Model Designer** appears.  
By default, the Data Model Designer displays the most recently used data model. When first opened during a session, it displays an empty canvas.



## Entities Panel

The entities panel lists all system entities and custom entities. You can hide/show this panel by clicking the Minimize / Expand arrow:



**HINT**

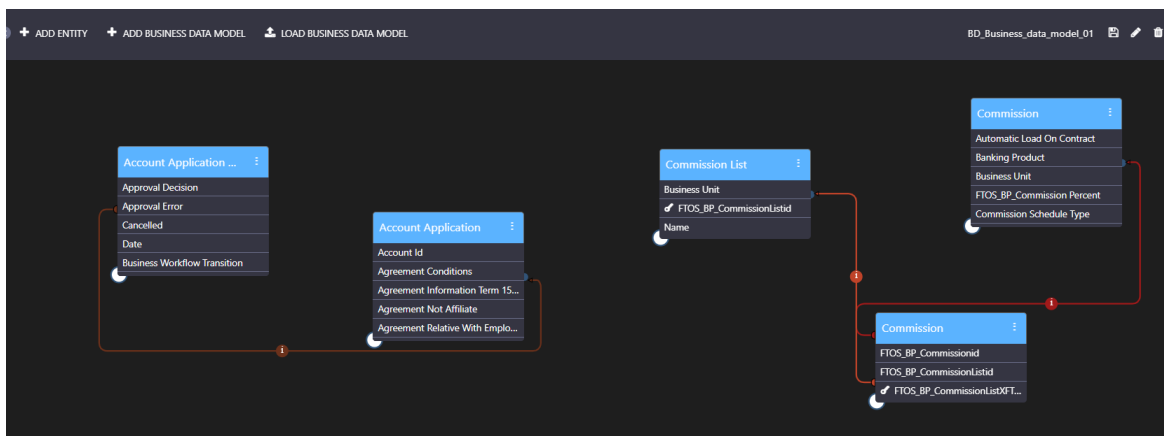
Use the Quick Find box at the top of the entities panel to filter entities. Also, hovering an entity name in the entity panel displays both entity name and entity display name. This is useful to identify entities with identical display names but different entity names.

## Toolbar

The Toolbar provides you with the buttons needed to perform common functions: Add Entity, Add Business Data Model, and Load Business Data Model. When a data model is open, the right-side of the toolbar contains additional icons which give you the means to save (💾), edit (✎), or delete (🗑) the current data model.

## Work Area Panel

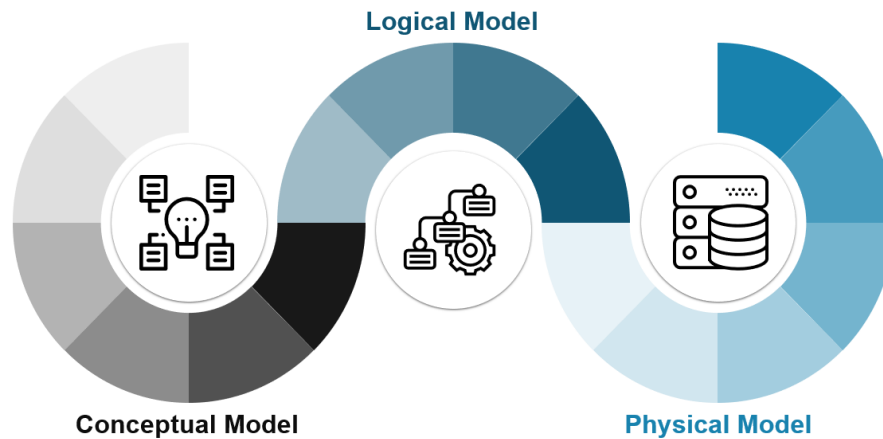
This is where you can design your data model in a visual interface.



- Drag and drop entities from the entities panel to add them to the data model.
- Drag the bottom left corner of an entity to a related entity to define a relationship.
- The overview panel at the bottom right corner of the work area allows you to zoom in and out and displays the current view's outline within the canvas.

## Create a Data Model

To create a well-defined data structure in a database, you need to identify the business concepts (entities, attributes, relationships), define data characteristics, and also define how the data is related.

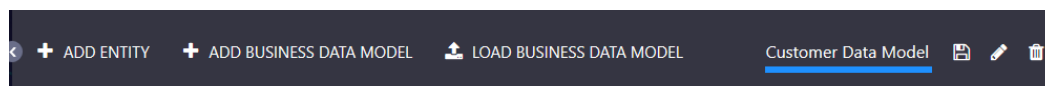


The first step in creating a data model after you open the **Data Model Designer** is to add the data model.

## 1 Add Data Model

1. Go to **Evolutive Data Core > Data Model Designer > Add Business Data Model**. The Add Business Data Model page opens.
2. Type a name for the data model in the **Name** field, a brief description (optional) in the **Description** field, and click **OK**.

The page closes and the Data Model Designer opens the data model you've previously added, its name is displayed on the toolbar.



**NOTE**

The next time you open the Data Model Designer, it will open on the data model you last worked on.

Now that you've added the data model, you can start adding entities.

## 2 Add Entities to Data Model

The next step in creating a data model is to identify all of the entities you will need. This could be a customer, an invoice, etc.

Once identified, make sure that you add the entities in FintechOS Studio. You can add entities in two ways:

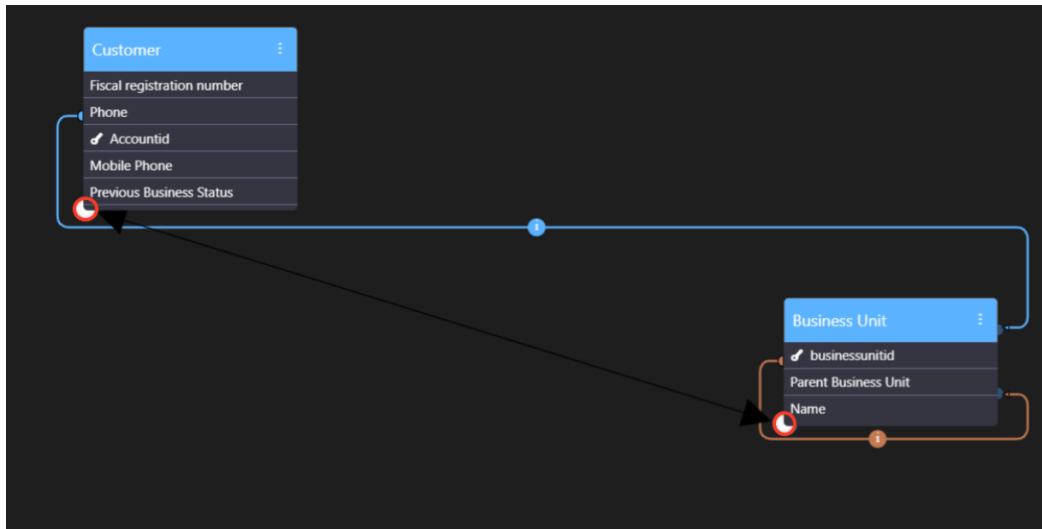
- From the **Data Model Designer** - On the toolbar, click **Add Entity**, provide the mandatory entity information and save it. The entity has by default a set of system attributes. You can extend the data model by adding new attributes as needed.
- From the **Data Model Explorer** - In the **Business Entity List**, by clicking **Insert**. Fill out the required properties, and click **Save and Close**.

In Data Model Designer, locate the newly created entity in the Entities panel, and then drag and drop it on the Work Area.

To easily spot the entity you want to add to your data model, you can use the search feature available at the top of the entities panel. If you need new entities, add them through the Data Model Designer or the Data Model Explorer, go back to the Data Model Designer, drag and drop an entity from the panel into the data model, and that's it!

## 3 Define Relationships

Identify relationships between the entities of your data model and describe the relationship. To define relationship between two entities of your data model, in **Data Model Designer**, select the circle displayed in the left bottom corner of one of the entities and drag it towards the circle of the other entity:



The Add Relationship page appears which allows you to define the relationship. For more information on relationships, their types and how to define them, see [Editing Entities](#).

After you save the relationship between two entities, it is displayed as object in Data Model Designer, and connectors are shown between it and the entities which it links (that is, the entities between which you created the relationship).

## Working with Data Models

### Load Data Models

When you open the Data Model Designer, it opens up the data model you last worked on.

If you want to work on a different data model that you created in FintechOS Studio, from the toolbar, click Load Business Data Model. The Load Business Data Model page appears which lists all data models that you created in FintechOS Studio:

The screenshot shows the 'Edit Business Data Model' dialog box. It has a title bar with a close button (X). The dialog contains the following fields:
 

- Name:** Customer Data Model
- Category:** Select...
- Line:** Select...
- Description:** This is a data model for demo purposes

 At the bottom, there are 'Ok' and 'Cancel' buttons.

In the list of existing data models, select the data model you want to load. It opens in the Data Model Designer.

## Edit Data Model Details

You can edit the details of a data model (name and description) at any time:

1. Load the data model whose details you want to modify (if it's not the current one).
2. Click Edit (✎). The Edit Business Data Model page appears.
3. Update the details as preferred and click OK.

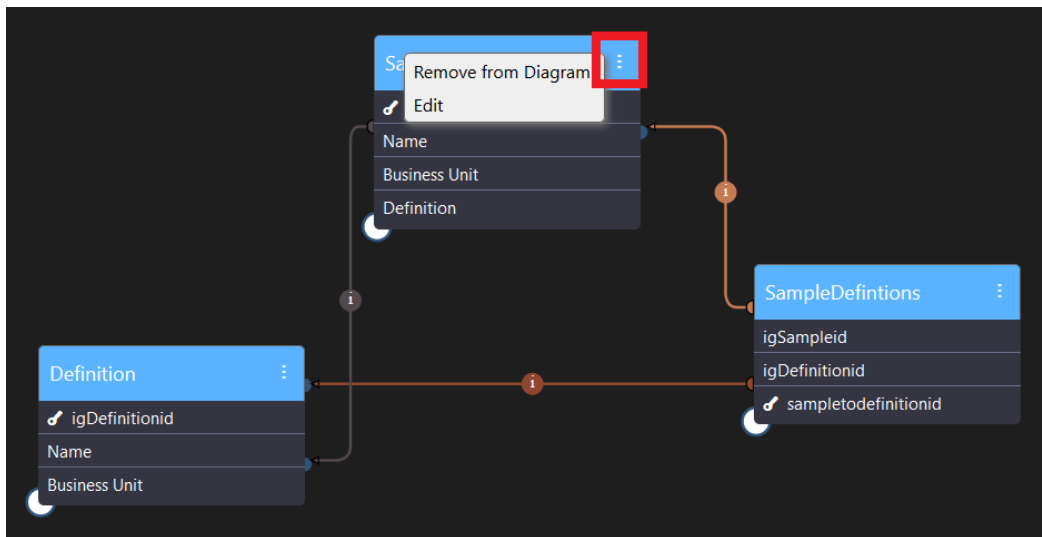
### IMPORTANT!

Make sure that you save the data model every time you edit it; otherwise, you lose the data when adding an entity or navigating to a different page.

## Edit Data Models

In FintechOS Studio, you can edit a data model at any time by first loading it in the Data Model Designer, then add, "[Edit Entities](#)" on [page 69](#) or delete entities, attributes and relationships between entities to best suit your business model.

Use the ellipse button to edit or delete it.




Removing an entity from the diagram also erases the entity's relationship lines, but the lookup attributes used to define those relationships are still preserved in the entity.

Hovering over the letter displayed on the entity lines displays a short description of the relationship, for example "SampleDefinitions linked to Definition through <idDefinitionId>".

Relationship lines have a different color to enable easier tracking in complex data models. There is no color coding employed.

### Delete Data Models

If you want to remove the current data model, on the toolbar, click Delete (  ) and in the confirmation pop-up, click Yes.

**NOTE**

The data model deletion *cannot* be undone, so we recommend you think twice before removing it.

## Data Model Explorer

The Data Model Explorer allows you to create business entities, relationships and attributes in an entity-relationship framework, handling at the same time data persistence and automatic database provisioning. It also facilitates data exchange through integration or bulk import of data.

This section covers the following topics:

<b>Business Entities</b>	<b>58</b>
<b>Attributes</b>	<b>71</b>
<b>Entity Unique Constraints</b>	<b>94</b>
<b>Entity Relationships</b>	<b>97</b>
<b>Advanced Entity Find</b>	<b>105</b>
<b>Data Views</b>	<b>112</b>

---

<b>Data Forms</b> .....	<b>142</b>
<b>Transient Data Entities</b> .....	<b>150</b>
<b>Sample API Calls</b> .....	<b>163</b>

## Business Entities

An entity is an object in the system that you want to model and store information about. Entities are recognizable concepts which have relevance to the database. Some specific examples of entities are: Customer, Product, Offer or Contract. An entity is similar to a table in the relational model.

### Types of entities

In FintechOS Studio, there are three types of entities:

- **Platform Data** - Native FintechOS Platform data which is created and stored within FintechOS Platform.
- **Transient Data** - Entities that temporarily store data that has been loaded from or is going to be saved to an external data source. For more information, see "[Transient Data Entities](#)" on page 150.

Platform Data entities encompasses:

- **System entities** - are used by FintechOS to run as an integrated operating system (OS). You can add Attributes on system entities, define specific forms and views, but you cannot delete them.

System entities are found in the database under the following schemas:

- **ebsMetadata** - stores entities' metadata. For example, information about the entities which are level 1S options on the FintechOS Studio main menu.
- **ebsLocalization** - stores records for entities Language and Currency Code.

- **ebsAudit** - stores logging information The ebsAudit schema is comprised of the EbsLogs.UniversalLog and EbsLogs.ApiLog. For more information, see the [FintechOS Platform Administration Guide](#).

System entities can be used within business processes, if needed. For example, the 'systemuser' entity, which stores information of users authorized to log in the platform can be equally used in 'task management' flows for assigning users to tasks.

- **Custom entities** - are the entities you define for your application in order to accommodate various business flows. For example: 'Contract', 'Application', 'Legal Agreement'.

Custom entities can be found in the database under the ebs schema or under the schema inserted in the organization table.

## View existing business entities

1. Select **Evolutive Data Core > Data Model Explorer**. The **Business Entities List** page appears.

Name	DisplayName	Is System Entity	Entity Type	Digital Asset
AC_Ent1	AC_Ent1	<input type="checkbox"/>	Platform Data	
AC_Ent11	AC_Ent11	<input type="checkbox"/>	Platform Data	AC_DA2
AC_Ent12	AC_Ent12	<input type="checkbox"/>	Platform Data	
AC_Ent13	AC_Ent13	<input type="checkbox"/>	Platform Data	
AC_Ent14	AC_Ent14	<input type="checkbox"/>	Platform Data	
AC_Ent98	AC_Ent98	<input type="checkbox"/>	Platform Data	
AC_Ent11	AC_Ent11	<input type="checkbox"/>	Platform Data	AC_DA1
AC_Ent11_BW	AC_Ent11 Business Workflow Transition	<input type="checkbox"/>	Platform Data	
AC_Ent12	AC_Ent12	<input type="checkbox"/>	Platform Data	AC_DA11
AC_Ent12_BW	AC_Ent12 Business Workflow Transition	<input type="checkbox"/>	Platform Data	

The entity details displayed are:

- Entity Type (Platform Data, External Source Data, and Transient Data)
- Name (as stored in the database), DisplayName (displayed on views and forms)
- Is System Entity (if the value is true, it is a system entity)

You can click column headers to order grids ascending or descending.

**HINT**

You can filter entities by all the search fields displayed in grid, including Name and DisplayName. You can apply multiple filtering criteria at the same time.

## Create Entities

1. Go to the **Business Entities List** page.
2. Click **Insert** and enter the required fields (attributes).

**NOTE**

Based on the attribute type you select, you need to fill-in fields mandatory for the selected ["Attributes" on page 71](#) type.

3. The minimum required fields to create an entity are:

Field	Description
Entity Type	<ul style="list-style-type: none"> <li>• Select one from the list: <ul style="list-style-type: none"> <li>• <b>Platform Data</b> - Native FintechOS Platform data which is created and stored within FintechOS Platform.</li> <li>• <b>Transient Data</b> - Entities that temporarily store data that has been loaded from or is going to be saved to an external data source. For more information, see <a href="#">"Transient Data Entities" on page 150</a>.</li> </ul> </li> </ul>

Field	Description
<b>Name (only use for add entity)</b>	<ul style="list-style-type: none"> <li>The unique entity name that will be stored in the database. Provide a self-explanatory name for what the entity stands for, or provide a hint about the business logic it entails.</li> <li>Use PascalCaseNames (upper camel) for it. Example: StatementPayment.</li> <li><i>You cannot create two entities with the sameName.</i></li> <li>The entity name is used to identify the entity in the system when working with it (read/ write entity records, define entity relationships, create data models, etc.) This field is used by the system and will be displayed only in the application URL. It is not visible to the end-user.</li> <li>The field is also used in implementation when calling all the CRUD operations on the specific entity (getByQuery,getById, etc.).</li> </ul> <div> <p><b>NOTE</b></p> <p>On entity creation, the entity's primary key attribute, that is an</p> </div>

Field	Description
	<p>unique identifier for each entity instance, is automatically generated by the system following this naming convention: 'entityname + Id'. The primary key is displayed in the entity's list of attributes and has the attribute type <b>PK</b>.</p>
<b>Display Name</b>	<p>The entity name that will be displayed on views and forms. It is also the label to be localized in different languages. The Display Name should appear as a noun in singular format (e.g. "Customer", or "Physical Address").</p>
<b>DisplayCollectionName</b>	<ul style="list-style-type: none"> <li>• Provide a Collection Name if you want to display the entity on the left-side menu or pin it on the application homepage.</li> <li>• The DisplayCollectionName attribute stores all the entity records (instances) within the database; therefore, the naming convention for this attribute is the plural data form of noun used for the entity Name.</li> </ul>

Field	Description
<b>TableName (only use for add entity)</b>	<ul style="list-style-type: none"><li>• The name of the table to be generated in the database, associated with the entity, automatically prefilled by the system based on the entity Name.</li><li>• To avoid affecting data integrity and consistency, do not change the value prefilled by the system.</li><li>• After you save the entity, you cannot edit this field.</li></ul>

Field	Description
<b>PrimaryAttributeName (only use for add entity)</b>	<ul style="list-style-type: none"> <li>The name of the main attribute that identifies the entity records from a business perspective. It is a text attribute with a maximum length of 100 characters.</li> <li>Start the name with a lowercase letter and all additional words (on the right) with uppercase.</li> <li>For example, if the Product entity has three attributes: ID, name and price. For each product, the ID will be unique, so it can be the primary attribute for this entity.</li> </ul> <div> <b>IMPORTANT!</b>  Do not confuse the entity's primary key automatically generated by the system to uniquely identify an entity in the database with the Primary Attribute . For example, you can define the Contract Number as a primary attribute but we do not recommend you to define it as a primary key due to the fact that, </div>

Field	Description
	<p>on data import into another system or version of FintechOS Studio, the contract number might already exist.</p>
<b>PrimaryAttributeDisplayName</b> (only use for add entity)	The name of the primary attribute, as displayed in the end-user interface on forms and views.
<b>PrimaryAttributeTableColumn</b> (only use for add entity)	<ul style="list-style-type: none"> <li>The name of the attribute which will be automatically generated in the database. It serves as a primary attribute to identify the PrimaryAttributeName (the main table column stored in the database). Use PascalCaseNames (upper camel) for it: provide a text which starts with an uppercase letter, as do all additional words.</li> <li>After you save the entity, you cannot edit this field.</li> </ul>

Field	Description
<b>Default Entity Status</b>	<ul style="list-style-type: none"> <li>Indicates the status corresponding to your work on this entity. To select the entity status, click the drop-down arrow and double-clicking the entity status. The following default entity statuses are available: <ul style="list-style-type: none"> <li>Active – completed work</li> <li>Draft – work in progress</li> </ul> </li> <li>The status selected in this field will appear on entity records when inserting. For instance, if you select by default Active on entity Application, when you insert an Application, the field <b>entityStatusId</b> is set to <b>Active</b>.</li> <li>The entity status doesn't impact the entity behavior in any way. It is useful as a way to classify and filter your entities when designing your data models.</li> </ul>

4. Click **Save and reload**. The **Add Business Entity** page is replaced by the **Edit Business Entity** page, which contains new sections at the bottom that allow you to add entity attributes, forms, views and more. Other fields appear when creating an entity for the first time, but they are not required to be filled in at creation:

- isAudited** - If selected, the checkbox enables you to track all the changes made on the entity record. An audit icon is displayed on the entity which on click will display the type of changes made on the entity, when the

changes have been made, and by whom. You will also see a side-by-side comparison between old and new values. For more information on entities audit, see the [Platform Administration Guide](#).

- **Business Workflow** - The business workflow that was attached to the entity is displayed. At first, it is a blank field, after a workflow is associated, the name is displayed here as read-only. Business workflows allow you to define states and state transitions for your entity records. For more information, see the [Business Workflows Processor documentation](#).
- **Optimization Search Data (Filter starts with)** - By default, the checkbox is not selected (record filtering is set to "contains"). This means that when Portal users enter search terms in the entity views' filters, the search returns records that contain the search term anywhere in the record attribute.

For example, a FintechOS Portal user does a search ("john") on the Name attribute of the Customers entity. The search returns all names that contain the "john" search term.

CUSTOMERS LIST		
<input type="checkbox"/>	Name	View
	 john	
	John Doe	View
	Johnathan Scott	View
	Mr. John Smith	View

If the Optimization Search Data (Filter starts with) checkbox is selected, the search returns only the records that start with the provided search term.

CUSTOMERS LIST	
<input type="checkbox"/>	Name
	abc john
	John Doe
	Johnathan Scott

- **Support Data Ownership Custom Scope** - Allows you to programmatically assign specific users access to particular entity records. By creating custom data ownership tags, you can associate these tags with the relevant user accounts and the entity records they are permitted to access. For more information, see the Server SDK documentation for the following methods:
  - [ftos.identity.authz.createGrantTag](#) - Create a custom data ownership tag.
  - [ftos.identity.authz.setGrantTagOnRecord](#) - Assign a custom data ownership tag to an entity record.
  - [ftos.identity.authz.getGrantTagOnRecord](#) - Retrieve any custom data ownership tag assigned to a record.
  - [ftos.identity.authz.RemoveGrantTagOnRecord](#) - Unassign a custom data ownership tag from a record.
  - [ftos.identity.authz.setGrantTagToUser](#) - Assign a custom data ownership tag to a user account.
  - [ftos.identity.authz.RemoveGrantTagToUser](#) - Unassign a custom data ownership tag from a user account.

## Edit Entities

You can edit an entity by editing existing attributes, adding more attributes (Data Model section), extend the data model with data extensions (see ["Extend the Data Model" on page 1](#)) and defining relationships.

### IMPORTANT!

Attributes can be deleted from the Data Model, however if the attribute has records that have been inserted by an end-user, then the records have to be deleted first then delete the attribute.

**Edit Business Entity**

**Details**

Entity Type: Platform Data

Name (only use for add entity): Account

DisplayName: Customer

DisplayCollectionName: Customers

Description:

PrimaryAttributeName (only use for add entity): Name

PrimaryAttributeDisplayName (only use for add entity): Name

Default Entity Status: Draft

### NOTE

You cannot edit the following entity attributes: EntityType, Name, PrimaryAttributeName, PrimaryAttributeDisplayName, once you configure it. The Business Workflow is read-only as well, but can be changed from the Business Workflows menu.

## Delete Entities

To delete entities, in the **Data Model Explorer**, select the entity or entities that you wish to delete and click **Delete**.

### IMPORTANT!

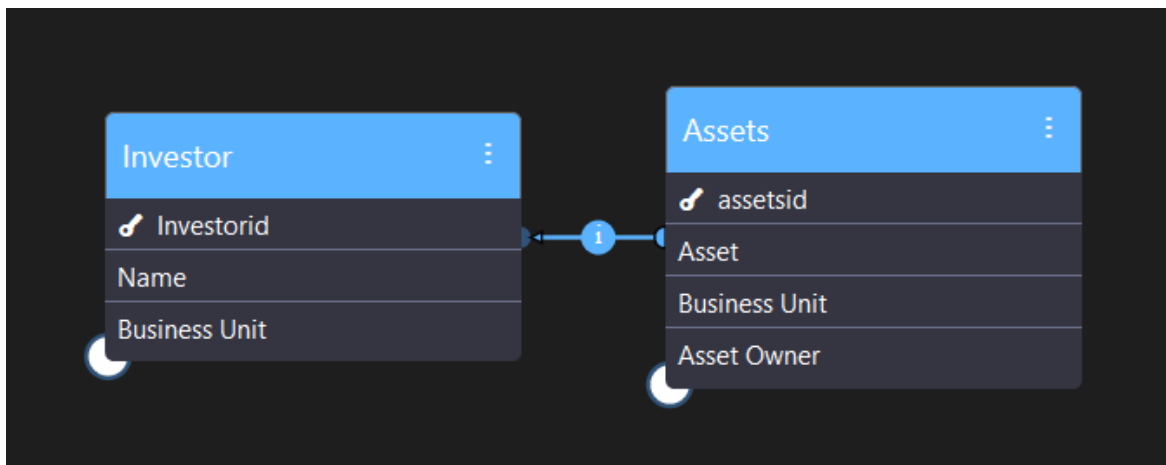
Deleting entities might have severe impact on the structure of your system's database; therefore, we strongly recommend you to make sure that the data model requires the entity deletion to address changes on existing business concepts.

These are a few consequences you need to consider before deleting an entity without prior analysis: custom actions defined on deleted entities will fail. Widgets linked to a deleted entity will fail. Reverse engineering becomes an exhaustive process by dealing with a broken DB structure.


## Referential Integrity Check

When deleting entities that are referenced by another entity, you first need to remove the relationship between the entities (delete the corresponding lookup attribute from the referencing entity).

In the example below, the Investor entity is referenced by the Assets entity.



Attempting to delete the Investor entity without deleting the relationship first, generates the following warning:

 **The following entities are referencing this one. You have to delete attributes: (Assets -> Asset Owner),**

## Attributes

An attribute is equivalent to a column in a table, available for the end-user to input or select data. For example, an entity is the "Account" and the attributes are "name", "age", "product", "no.ofcontract", "policyno", "address", "income". Attributes include primary keys and foreign keys (type lookup) as well.

FintechOS Studio supports a variety of field types, from usual fields like: text, date, boolean or numeric, to advanced fields like lookup (referencing other entities) or optionset (drop-down list).

You can access the entity attributes from the Data Model section of the entity.

▼ Data Model

+ Insert X Delete Export Refresh

Name	Display Name	Attribute Type	Entity	Order Index	Digital Asset
AddressId	AddressId	Pk	Address	1	FTOS Foundation
Name	Address name	Text	Address	2	FTOS Foundation
createdOn	Created On	Date Time	Address	7	FTOS Foundation
modifiedOn	Modified On	Date Time	Address	8	FTOS Foundation
userId	User	Lookup	Address	3	FTOS Foundation
createdByUserId	Created by user	Lookup	Address	4	FTOS Foundation
modifiedByUserId	Modified by user	Lookup	Address	5	FTOS Foundation
businessUnitId	Business Unit	Lookup	Address	6	FTOS Foundation
entityStatusId	Status	Lookup	Address	9	FTOS Foundation
AccountId	Account	Lookup	Address	0	FTOS Foundation
AddressStatusId	Address Status	Option Set	Address	0	FTOS Foundation
AddressTypeId	Address type	Option Set	Address	0	FTOS Foundation
ApartmentNo	Apartment number	Text	Address	0	FTOS Foundation
BuildingNo	Building number	Text	Address	0	FTOS Foundation
CityId	City	Lookup	Address	0	FTOS Foundation
DistrictId	County	Lookup	Address	0	FTOS Foundation
FloorNo	Floor number	Text	Address	0	FTOS Foundation
PostalCode	Postal code	Text	Address	0	FTOS Foundation
StreetName	Street name	Text	Address	0	FTOS Foundation
StreetNo	Street number	Text	Address	0	FTOS Foundation

The buttons at the top left corner of the section have the following functions:

- **Insert** - Adds a new attribute. For details, see ["Add Attributes" on page 85](#).
- **Delete** - Deletes the currently selected attributes.

- Export - Exports the currently selected attributes' metadata in an Excel file.
- Refresh - Refreshes the list of attributes.

## System-generated attributes

When you create an attribute, a new column is added within the table corresponding to your entity in the database.

### IMPORTANT!

All entities have a set of auto-generated attributes that are used for entity auditing purposes. **DO NOT** remove them.

This is the list of system-generated attributes:

Attribute Name	Attribute Display Name	Description
entitynameid	entitynameid	The entity unique identifier, the Name provided when creating the entity name.
createdOn	Created On	The date and time when the entity was created.
modifiedOn	Modified On	The date and time when the entity was updated.
userId	User	The current user or the owner of the entity record.
createdByUserId	Created by User	The user who inserted that record.
modifiedByUserId	Modified by User	The user who made the last updates on the entity record.
businessUnitId	Business Unit	The business unit associated with the attribute userId. It is the business unit of the user.
entityStatusId	Status	The status of the entity record.

### HINT

The PrimaryAttributeName is also generated automatically because it is required to create an entity, but the actual name is chosen by the user.

## Types of Attributes

This section describes the types of attributes (fields) you can add in FintechOS Studio:.

## Text

A basic control that enables the user to type one-line set of characters (text). Use it for short alphanumeric attributes such as names or user IDs.

When adding a text attribute to an entity, you need to provide the following specific properties:

Property	Description
Length	The maximum number of characters users will be allowed to enter in the field.
Is localizable (only for text)	If selected, the text fields will be marked as being localizable. For information on how to localize fields, see <a href="#">"Localization" on page 1269</a>

This is how a text field is configured:

The screenshot shows the 'Edit Attribute' configuration form for a text field. The form has a left sidebar with labels for various properties, and the main area contains input fields for each property. The properties and their values are as follows:

- Name:** ApartmentNo
- Attribute Type:** Text
- Display Name:** Apartment number
- Description:** This is the apartment number for the address.
- Tooltip:** (Empty text area)
- Table Column Name:** ApartmentNo
- Length:** 100
- Is Localizable:** ☐
- Required Level:** None
- Default Value:** (Empty text field)
- Regular Expression:** (Empty text field)
- Is Readonly:** ☐
- Is Securable:** ☐

## Text Area




Defines a multi-line text input control. Use it when users need to provide large amounts of text that exceeds one line such as descriptions, messages, feedback, etc.

When adding a text area attribute to an entity you need to provide the following specific properties:

Property	Description
Length	The maximum number of characters users will be allowed to enter in field.
Is localizable (only for text)	If selected, the text fields will be marked as being localizable. For information on how to localize fields, see <a href="#">Localization</a> .

This is how a text area field is configured:

dit Attribute

Name	FullAddress
Attribute Type	Text Area 
Display Name	Full Address
Description	
Tooltip	
Table Column Name	FullAddress
Length	400
Is Localizable	<input type="checkbox"/>
Required Level	None  
Default Value	
Regular Expression	
Is Readonly	<input type="checkbox"/>
Is Securable	<input type="checkbox"/>

## Whole Number

Use this field to insert any integer (whole number) with a value between - 2,147,483,648 and 2,147,483,647. Use integers for attributes such as number of children, number of insured persons, number of cars, number of monthly payments for a credit, maximum number of co-debitors, etc.

When you use integers, you can't add decimals. If you need to make very accurate calculations - for example, for calculating interest rates, and you need to utilize decimals, use numeric attributes instead.

When adding a whole number attribute to an entity you need to provide the following specific properties:

Property	Description
Is identity (it will be read-only)	If selected, the whole number attribute is automatically incremented for each record and becomes read-only. It is useful for unique record identifiers, such as: the contract number or the policy number.

The value inside the whole number fields is by default formatted to the right.

## Numeric

Defines a field to enter numeric values. Use it when for attributes that require very accurate calculations, or if you typically use queries that look for values that are equal or not equal to another value, for example interest rates.

When adding a numeric attribute to an entity you need to provide the following specific properties:

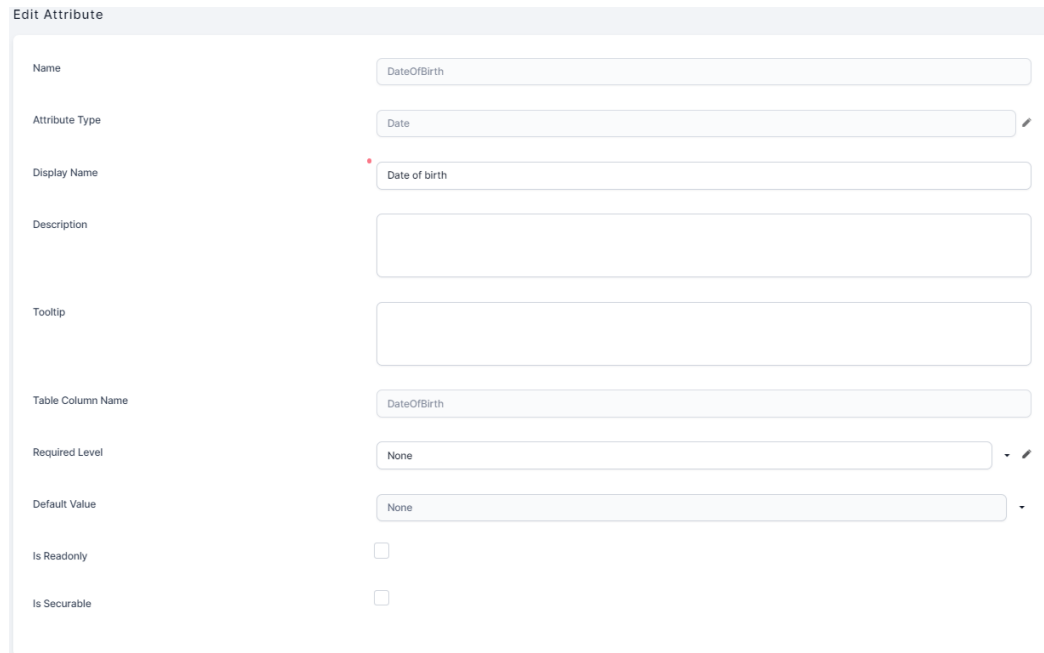
Property	Description
Precision	Specify the number of decimals (up to 9 decimal points of precision) to be displayed in the user interface in case of numeric fields.

## Date

Defines a field which has the format option to display date only. No specific properties need to be provided. Use it for attributes such as birth dates, expiration dates, or issuing dates.

When you click it, the calendar opens, allowing you to select a date.

This is how a date field is configured:



The screenshot shows the 'Edit Attribute' form in FintechOS Studio. The form has a light gray header with the title 'Edit Attribute'. Below the header, there are several input fields and checkboxes arranged in two columns. The left column contains labels for each field, and the right column contains the corresponding input values.

Field Label	Value
Name	DateOfBirth
Attribute Type	Date
Display Name	Date of birth
Description	
Tooltip	
Table Column Name	DateOfBirth
Required Level	None
Default Value	None
Is Readonly	<input type="checkbox"/>
Is Securable	<input type="checkbox"/>

The date format is as defined on the ApplicationLanguage entity, that is, you can have different date formats per language. For details on how you can format the date field throughout FintechOS Platform per language, see [Add a New Language](#).

## Date Time

Defines a field which has the format options to display the date and time. No specific properties need to be provided. Use it for attributes that need to record the precise date and time, such as registration when an issue was raised or when a fraud was attempted.

The format of the dateTime field is as defined on the ApplicationLanguage entity, that is, you can have different date formats per language. For details on how you can format the dateTime field throughout FintechOS Platform per language, see [Add a New Language](#).

This is how a dateTime field is configured:

The screenshot shows the 'Edit Attribute' interface. On the left, a sidebar lists various attribute properties. On the right, each property has a corresponding input field or control. The values currently set are: Name: 'createdOn', Attribute Type: 'Date Time', Display Name: 'Created On', Table Column Name: 'CreatedOn', Required Level: 'None', and Default Value: 'None'. The 'Is Readonly' and 'Is Securable' checkboxes are currently unchecked.

## Bool (Boolean)

Defines an attribute which is a checkbox. It can have one of the following values:

- **NULL** - it is the default state of the bool attribute. Indicates that no action has been performed on the checkbox yet.
- **TRUE** - it indicates that the checkbox has been selected.
- **FALSE** - it indicates that the checkbox has not been selected.

Use it for validations, for instance if the customer is a politically exposed person or if he has a criminal record or not.

## Lookup

Defines a relation between the entity you're working on and another entity (that is, the parent entity), for example an Asset Owner attribute in an Assets entity that refers to the records in the Customers entity.

When you create a new lookup attribute, you are creating a new Many-to-One (N:1) entity relationship between the entity you are working with and the **Lookup to Entity** defined for the lookup.

All lookup attributes display the primary attribute name of the referenced entity; therefore, you should always provide a value for the primary attribute on the parent entity.

When adding a lookup attribute to an entity, you only need to provide the following properties:

Property	Description
Name	The name of the attribute. Make sure that you use the following naming convention: 'the referenced entity' + suffix 'Id', pascal case, no special characters and no blank spaces. For example: a lookup on the 'Contract' entity for the 'ContractType' entity will be named 'ContractTypeId'.
Attribute Type	Select <b>Lookup</b> from the drop-down list. The <b>Lookup to Entity (only for lookups)</b> field becomes mandatory.
Display Name	The name of the attribute that will be displayed on the data form in the user interface if the <b>Auto-generate data form</b> checkbox is selected on the entity level.
Description	Insert a proper description for the attribute.
Tooltip	Insert a proper tooltip to help the user understand what it is about.
Table Column Name	<p>The name of the attribute which will be automatically generated in the database. This field is not visible in the end-user interface.</p> <div> <p><b>NOTE</b> To avoid affecting data integrity and consistency, do not change the value prefilled by the system.</p> </div>
Required level	<p>Select whether the attribute is mandatory to fill in or not:</p> <ul style="list-style-type: none"> <li>• none</li> <li>• recommended</li> <li>• required.</li> </ul>
Lookup to Entity	The parent entity for the entity you are currently working on.

Property	Description
Lookup relationship type	<p>Select one from the list:</p> <ul style="list-style-type: none"> <li>Dictionary (default)</li> <li>IsChildOf (entity A is the child of related entity, it is a 1:N relationship, see <a href="#">"1:N Entity Relationships" on page 97</a>)</li> <li>One to one (this is a one-to-one relationship, see <a href="#">"Entity Relationships" on page 97</a>)</li> </ul>
Lookup Relationship Name	This field is automatically filled-in by the system with the concatenation of the two entity names, following this naming pattern: <b>ChildEntity_PK_ParentEntity</b> .

This is how a lookup attribute is configured:

Edit Attribute

Name	accountCountryId
Attribute Type	Lookup
Display Name	Country
Description	
Tooltip	
Table Column Name	AccountCountryId
Required Level	None
Lookup To Entity	Country
Lookup Relation Type	Dictionary
Lookup Relationship Name	Account_FTOS_CMB_Country
Is Readonly	<input type="checkbox"/>
Is Securable	<input type="checkbox"/>

The following actions are available on lookup fields:

- **Select record:** Opens the view with records existing within the referenced entity.
- **Edit record:** Opens the edit data form of the record selected within the lookup attribute.

## Map

Defines a map to be displayed in the application. No specific properties need to be provided. Use it for attributes that display location data like where a business's headquarters is located on a map.

When displayed on a form, the user can scroll to zoom in and out, drag to pan the map, and click on the map to set/unset location markers.



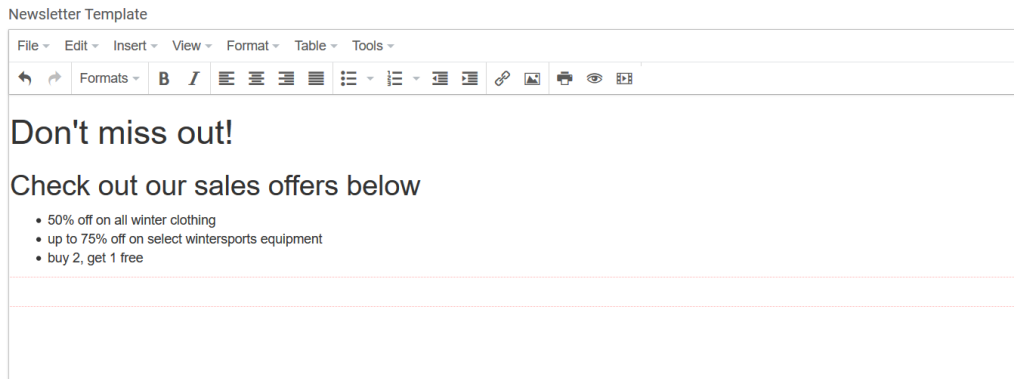
The actual data is saved in the database in the form of a JSON object that stores the geographic coordinates of the location markers and map zoom level.

```
{"markers" : [{"location" : {"lat" : 44.42753573214415,
"lng" : 26.08738040542604}}, {"location" : {"lat" :
44.426929932048424, "lng" : 26.1024179417852}}], "zoom" :
15}
```

## HTML

Defines a HTML block of code. HTML fields allow displaying a rich text editor that can be used to quickly generate the underlying HTML code. It uses tinyMCE which interprets the HTML code.

HTML fields are particularly useful when you want to use customer tailored content, for instance to create newsletter templates to be included in your marketing campaigns.



The underlying HTML code that is stored in the database for the example above is:

```
<h1>Don't miss out!</h1>
<h2>Check out our sales offers below</h2>
<ul>
  <li>50% off on all winter clothing</li>
  <li>up to 75% off on select wintersports equipment</li>
  <li>buy 2, get 1 free</li>
</ul>
```

## Color

Allows selecting and displaying a color. No specific properties need to be provided. Use it for cosmetic customizations, such as user interface themes.

## File

A control that allows end-users to add (upload) an attachment, either by clicking the **Add file** button or by dragging and dropping the file in the corresponding section. Use when the user needs to upload or download a file attachment such as a contract, agreement, or statement.

The uploaded files are stored in the file upload location configured for the environment. For details about file storage, see the [FintechOS Platform Administration Guide](#).

File attributes include a **restrict files number** option that limits the number of files that can be added to an attribute and a field called **maximum number of files** which allows you to set that limit to the desired value. The maximum number of files that can be attached to an attribute is 100.

The screenshot shows the 'Edit Attribute' form with the following fields and values:

- Name:** AccountPicture
- Attribute Type:** File
- Display Name:** Customer picture
- Description:** (empty)
- Tooltip:** Add a file
- Table Column Name:** AccountPicture
- Restrict files number:** ☒
- Maximum number of files:** 1
- Required Level:** None
- Is Readonly:** ☐
- Is Securable:** ☐

### IMPORTANT!

To make sure that the files that you upload are malware-free, you need to enable the malware detection feature. For more information about malware scanning, see the [FintechOS Platform Administration Guide](#).

## JavaScript (JS)

JS fields allow usage of JavaScript code on data form level. No specific properties need to be provided. Use it for advanced customizations that you wish to add through the user interface.

## Order Index

Allows you to drag and drop the rows existing in a grid (view). The order index attributes are not displayed on data form level but are used only in views to order attributes by a particular index number.

When adding a text attribute to an entity, you need to provide the following specific properties:

Property	Description
Order index attribute reference (only if order index)	The attribute based on which the records in views will be ordered by.

In order to display optionset items in a particular order in the drop-down list, drag and drop in the order to display them.

## Option Set

An option set attribute allows you to define a list with several options available for selection. Use it when the attribute can take a single value from a limited set of options, such as country, city, currency, etc.

When you add an **Option Set** field to a data form, you can specify multiple values that will be available for users to select. For more information, see [Add an Option Set Attribute](#).

## Money

It defines a price field which has included a thousand separator. No specific properties are required. Use it for monetary values such as credit values, interests, fees, etc.

## Icon Picker

Allows you to select an icon from the list of available (predefined) icons and display it based on your preference. This type of attribute is currently embedded within FintechOS Platform to allow selection of icons to be displayed on shortcuts that will be pinned on the homepage.

To select an icon, you have to click in the **Icon URL** field, and select the desired icon from the icons selection pane.

## Invariant Date

Defines a field which has the format option to display date only. This type of field takes into consideration different timezones, Daylight saving time, winter time or leap a specific day based on year (e.g. February 28). Use it for attributes such as birth dates, expiration dates, or issuing dates.

No specific properties need to be provided.

You can use it to define dates like: Inception Date, Start Date, End Date or Due Date.

## Unique identifier

This type of attribute is structured with 36 characters from the hexadecimal system (from 0 to F), with four lines.

For example, 43F48DD2-66E7-4ECC-9940-0219F8A5973F.

## CSS

Cascading Stylesheets styles and structured the manner in which the HTML code is displayed.

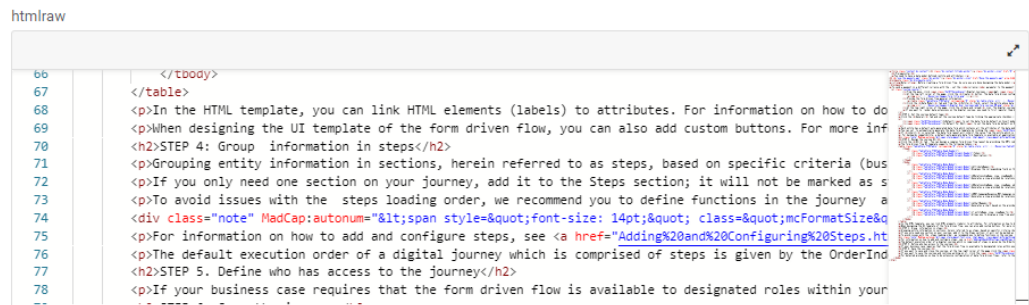
This is how it will look in the FintechOS Portal after you add the code.



## HtmlRaw

Displays the field where you can add HTML code with Monaco editor and HTML IntelliSense support.

This is how it will look in the FintechOS Portal after you add the code.



# RawText

It renders information from the log file. It is not validated XSS.

## Add Attributes

To add new custom attributes to an entity, follow the steps below:

- 1. From the **Edit Business Entity** page, expand the **Data Model** section.
- 2. Click **Insert**. The Add Attribute page appears where you provide the properties.

Data Model					
<div><div>+ Insert</div><div>X Delete</div><div>Export</div><div>Refresh</div></div>					
<input type="checkbox"/>	Name	Display Name	Attribute Type	Entity	Order Index
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	AddressId	AddressId	Pk	Address	1
	Name	Address name	Text	Address	2
	createdOn	Created On	Date Time	Address	7
	modifiedOn	Modified On	Date Time	Address	8
	userId	User	Lookup	Address	3
	createdByUserId	Created by user	Lookup	Address	4
	modifiedByUserId	Modified by user	Lookup	Address	5
	businessUnitId	Business Unit	Lookup	Address	6
	entityStatusId	Status	Lookup	Address	9
	AccountId	Account	Lookup	Address	0
	AddressStatusId	Address Status	Option Set	Address	0
	AddressTypeId	Address type	Option Set	Address	0
	ApartmentNo	Apartment number	Text	Address	0
	BuildingNo	Building number	Text	Address	0
	CityId	City	Lookup	Address	0
	DistrictId	County	Lookup	Address	0
	FloorNo	Floor number	Text	Address	0
	PostalCode	Postal code	Text	Address	0
	StreetName	Street name	Text	Address	0
	StreetNo	Street number	Text	Address	0

### NOTE

Based on the attribute type you select from the **Attribute Type** drop-down list, you

need to provide details corresponding to that specific attribute. For more information, see the Types of Attributes.

This is the generic list of properties you need to provide when adding a new attribute (field).

Property	Description
Name	<p>The name of the attribute. This is used to identify the attribute in the data model when you design the user interface, for instance to specify which attribute is displayed in a specific field.</p> <div> <p><b>NOTE</b></p> <p>A naming convention is an important part in a well-built data model; therefore, we recommend you to use PascalCaseNames (upper camel), except for the first letter. The Name starts with a lowercase letter and all additional words start with an uppercase letter. Example: accountId.</p> </div>
Attribute Type	From the drop-down list, select the type of attribute you want to add.
Display Name	The name of the attribute that will be displayed on the data form in the user interface. You can overwrite the <b>Display Name</b> using other commands directly in the HTML data form.
Description	Insert the proper description.
Tooltip	<p>A message inserted here will show when the user hovers the attribute in the FintechOS Portal.</p> <p>Tooltips can be a powerful UI pattern which help you guide your users to take specific actions within the product; thus, enhancing the user experience.</p> <p>If tooltips are activated on data form driven flows, for all attributes to which you want to show tooltips in the Digital Experience Portal, in the Tooltip field, provide the tooltip text.</p> <p>Optionally, you can add tooltips to specific attributes which can be shown in the Portal UI on data form driven flows.</p>
TabelColumnName	This is the name of the table column.

Property	Description
Required Level	<p>From the <b>Required Level</b> drop-down list you can choose if a specific attribute (field) is going to be mandatory, recommended or optional:</p> <ul style="list-style-type: none"> <li>• <b>None</b> – The field is optional. No error message will be displayed if the field is not completed by the end-user.</li> <li>• <b>Recommended</b> – A blue dot will be displayed on the upper-left corner of the field in the user interface to indicate that it might be useful to fill in the field.</li> <li>• <b>Required</b> - A red dot will be displayed on the upper-left corner of the field in the user interface to indicate that it is a mandatory field. The end-user will not be able to add a new record if the field is not completed.</li> </ul> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• You can only add required attributes to entities which have no records (empty entities). If you try adding a required attribute to an entity for which you already have required attributes stored within the database, you'll receive an error message.</li> <li>• You can add required attributes without creating constraints in the database, from the Forms section by using the <b>After generate events</b> field and the capabilities of field options.</li> </ul> </div>
isReadOnly	The attribute is read-only if true, i.e. the front-end user will not be able to insert any data in this attribute.

Property	Description
isSecurable	<p>Allows you to control access to the attribute using security roles. When this flag is checked, access to the attribute is restricted by default to all user roles except the administrator. To allow access to a secured attribute, you will have to explicitly configure security roles for this purpose (see <a href="#">"Security Roles" on page 1245</a> for details).</p> <div> <p><b>NOTE</b></p> <p>This option is disabled for system-generated attributes.</p> </div>

Always make sure to save your configurations by clicking one of the available save icons.

## Reorder Entity Attributes

If you need to change the order of attributes in an entity, go to the edit configuration page of that entity (by double-clicking on the entity in the **Business Entities List** page), scroll-down to the **Data Model** section and drag and drop attributes from one position to another in the list. The order index will be automatically updated based on the change you made.

If there are many attributes in the list, the "Please wait..." message will be displayed until the DB is updated. Once the update completes, the platform loads the attributes in the new order and a message displays at the bottom of the page indicating for how many records the order index has been updated.

▼ Data Model

+ Insert
X Delete
Export
Refresh

Name	Display Name	Attribute Type	Entity	Order Index	Digital Asset
Name	Name	Text	TestEntity_AT	7	
TestEntity_ATId	TestEntity_ATId	Pk	TestEntity_AT	1	
TestAttribute2	TestAttribute2	Text	TestEntity_AT	2	
Iban	IBAN	Text	TestEntity_AT	3	
userId	User	Lookup	TestEntity_AT	4	
createdOn	Created On	Date Time	TestEntity_AT	5	
modifiedByUserId	Modified by user	Lookup	TestEntity_AT	6	
TestAttribute	TestAttribute	Text	TestEntity_AT	8	
businessUnitId	Business Unit	Lookup	TestEntity_AT	9	
entityStatusId	Status	Lookup	TestEntity_AT	10	
createdByUserId	Created by user	Lookup	TestEntity_AT	11	
modifiedOn	Modified On	Date Time	TestEntity_AT	12	

## Add Option Set Attributes

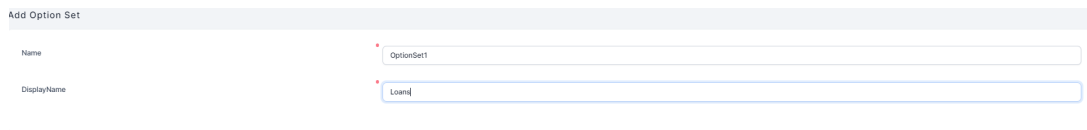
An option set attribute allows you to define a list with several options available for selection. When you add an option set field to a data form, you can specify multiple values that will be available for users to select. When users fill out the data form, they can select one value displayed in a drop-down list.

Users who are familiar with FintechOS Studio (know how to navigate through pages and pop-ups with minimal clicks and actions) can add an option set attribute, then add options sets and items. FintechOS Studio beginners should define the options set, then add the option set attribute.

This section walks you through the steps you need to follow in order to add and define an option set attribute.

### STEP 1. Define the option set (picklist)

1. Select **Admin > Option Sets**. The Option Sets List page will be displayed.
2. Click **Insert**, fill-in the **Name** (must be unique and is not visible in the UI), and **DisplayName** (displayed in the UI).



The screenshot shows a form titled "Add Option Set". It contains two text input fields. The first field is labeled "Name" and contains the text "OptionSet1". The second field is labeled "DisplayName" and contains the text "LoanId". Both fields have a red asterisk to their left, indicating they are required.

3. Click **Save and reload**. The Add Option Set page is replaced by the **Edit Option Set** page. The Items section displayed at the bottom of the page allows you to create, edit, delete and change the order in which options are presented.

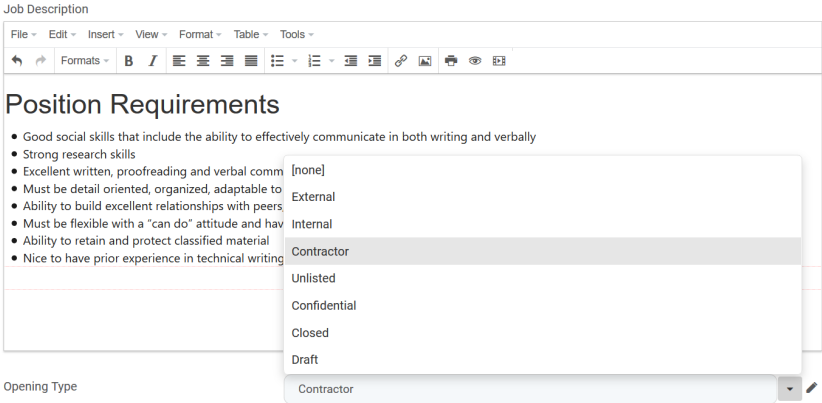
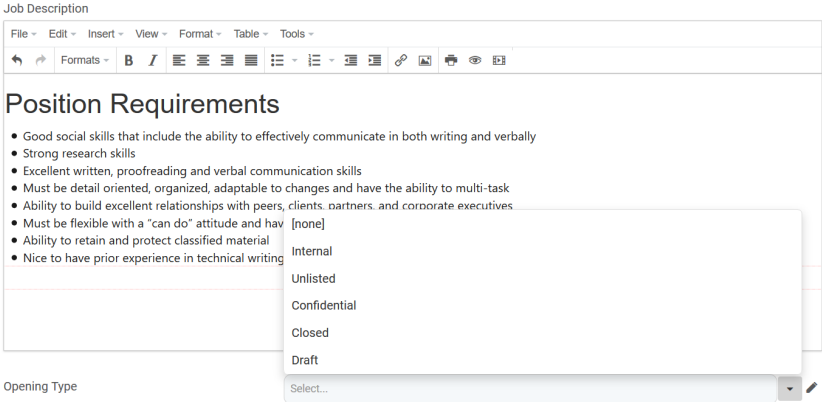
Now that you've created the option set, you can start adding items.

### STEP 2. Add new option set items

Double-click the desired option set and in the **Edit Option Set** page follow these steps:

1. From the Items section, click **Insert** and fill-in the following fields:

Field	Description
Name	The option set item name which will be used by the system.
Display Name	<p>The option set name which will be displayed in the user interface.</p> <div> <p><b>NOTE</b></p> <p>The maximum length of the Display Name for Option Set Items is 200 characters.</p> </div>
Value	It is used for mapping (e.g., if you want to get from a script the optionsetitem). The value must be unique within the options. We recommend that you add incremental values in this field as the value might be useful in workflows, flow automation and business rule engines.
Id	Unique ID automatically assigned by the system for the option set item. No user input is required.

Field	Description
StatusId	<p>The drop-down has two options: Active and Inactive. Select <b>Active</b> to make the business option available.</p> <p>Option set entity attributes will not list inactive option set items in the user interface. For instance, an option set may contain values for internal, external, contractor, unlisted, confidential, closed, and draft job openings:</p>
	
	<p>During a hiring freeze, a decision is made to suspend external and contractor job openings and the corresponding option set items are set as inactive, making them invisible in the user interface for any entity attribute that uses the option set:</p>
	
	<p>Once the hiring freeze is over, the option set items can be activated again.</p>

2. Click **Save and reload** . The page refreshes and the option set option is listed in the **Items** section.

**Edit Option Set**

Name:

DisplayName:

Is System Option Set: ☐

OptionSetItems:

Order	Name	Value
1	Before	1
2	After	2

You can add as many option set items as you want to fulfill your business needs.

#### Change the items order

You can change the order of the items by dragging the items from one position to another in the items list.

#### Edit option set items

You can edit an existing option set item from the **Edit Option Set** page. In the **Items** list table, double-click the desired item. The **Edit Option Set Item** page displays where you can change the **Value**, **Display Name** and **Status Id**.

To save the changes, click **Save and reload**.

#### Delete Option Set Items

Select the item from the **Items** list, click **Delete**, and click **Yes** to confirm deletion.

### IMPORTANT!

If you remove an option that has already been used in entity records, the data in those entity records becomes invalid; therefore, you should not remove the option but mark it as obsolete. To do so, double-click it and from the **Status Id** drop-down list, select **Inactive**.

### STEP 3. Define the option set attribute

To add a new Option Set attribute to an entity, follow these steps:

1. Select **Evolutive Data Core > Data Model Explorer**. The **Business Entities List** displays.
2. Double-click the entity to which you want to add the option set attribute. The **Edit Business Entity** page displays.
3. Expand the **Data Model** section and click **Insert**. The **Add Attribute** page displays.
4. Provide the following properties:

Property	Description
Name	The name of the attribute. Make sure that you use the following naming convention: 'optionset name' + suffix 'Id', <b>Example:</b> an attribute 'AccountTypeId', where 'Account Type' is the optionset name.
Attribute Type	Select <b>Option Set</b> from the drop-down list. The Option set name (only if optionset) field becomes mandatory.
Display Name	The name of the attribute that will be displayed on the data form in the user interface if the <b>Auto-generate data form</b> checkbox is selected on the entity level.
Table Column Name	The name of the attribute which will be automatically generated in the database. This field is not visible in the end user interface.  <div> <b>NOTE</b>            To avoid affecting data integrity and consistency, do not change the value prefilled by the system.         </div>


Property	Description
Option Set	The name of the option set (picklist) you defined. You can either enter the option set name or click the drop-down arrow on the right side of the field and double-click it. If you have many option sets displayed within the list, you can use the search feature available to search the one you want to add to the option set attribute.
Option Set - default value	This property allows you to display a default value in the drop-down list on the data form. For usability purposes, enter the value which is more likely selected in most cases by users from the drop-down list on the entity data form.

## Entity Unique Constraints

Unique constraints allow you to define attributes or combinations of attributes that must have unique values for each entity record. For instance, in an entity that stores personal data, you may want to set up the Social Security Number to be unique for each person. Or, in an entity that stores invoices, you may want to configure the combined invoice number and invoice series to be unique for each invoice.

Once enabled, unique constraints prevent record inserts or updates that do not meet the defined uniqueness criteria. You can define multiple constraints for the same entity.

### Create a Unique Constraint for an Entity

1. Open the Main Menu () and select **Evolutive Data Core > Data Model Explorer**.
2. Double-click the entity for which you wish to add uniqueness constraints, expand the **Entity Unique Constraints** section and click **Insert**.

3. In the **Add Entity Unique Constraint** page:
  - a. Enter a **Name** for the constraint. You cannot have multiple constraints with the same name for the same entity, but you can reuse constraint names on different entities.
  - b. Optionally enter a **Display Name** for the constraint. This is the name that will be displayed in the user interface.
  - c. Optionally enter a **Description** for the constraint.

4. Click **Save and Reload** ()

## Add Unique Constraint Attributes

1. In the **Edit Entity Unique Constraint** page, go to **Entity Unique Constraint Attributes** section and click **Insert**.

### NOTE

You cannot insert attributes in enabled constraints. If your constraint is enabled, you must disable it first (see ["Enable/Disable a Unique Constraint" on the next page](#) for details).


2. Select an **Attribute** that you wish to include in the constraint, from the drop-down list.

The **Name** field will be filled-in automatically.

Add Entity Unique Constraint Attribute

Entity Unique Constraint Attribute

Name  Attribute

3. Click **Save and Close** .
4. Repeat for any additional attributes you wish to include in your constraint. If you add multiple attributes, the constraint will evaluate if all the attributes combined are unique, not if each attribute is unique.

## Enable/Disable a Unique Constraint

When you create a unique constraint, it is disabled by default. To enable the constraint, click **Enable** in the Entity Unique Constraint page. If you need to disable it, you can toggle back to **Disable**.

Entity Unique Constraint

Enabled ☐ **Enable**

Name  Display Name

Description

Entity Unique Constraint Attributes

☐ Name

Edit Entity Unique Constraint

Entity Unique Constraint

Enabled ☒ **Disable**

Name  Display Name

Description

Entity Unique Constraint Attributes

☐ Name

**IMPORTANT!**

If the existing entity records don't meet the constraint requirements, you will not be able to enable the constraint. The following message will be displayed:



**There are duplicate entries in the database that do not respect the constraints.**



## Entity Relationships

Entity relationships define how records can be related to each other in the database, their associations and dependencies.

There are two types of entity relationships in FintechOS Studio:

- **1:N (One-to-Many)** - an entity relationship where one entity record for the primary entity (parent entity) can be linked to many other entities (child entities).

When viewing a parent entity in the user interface, you can also view the list of child entities.

- **N:N (Many-to-Many)** - an entity relationship where many entity records can be linked to many child entities.

---

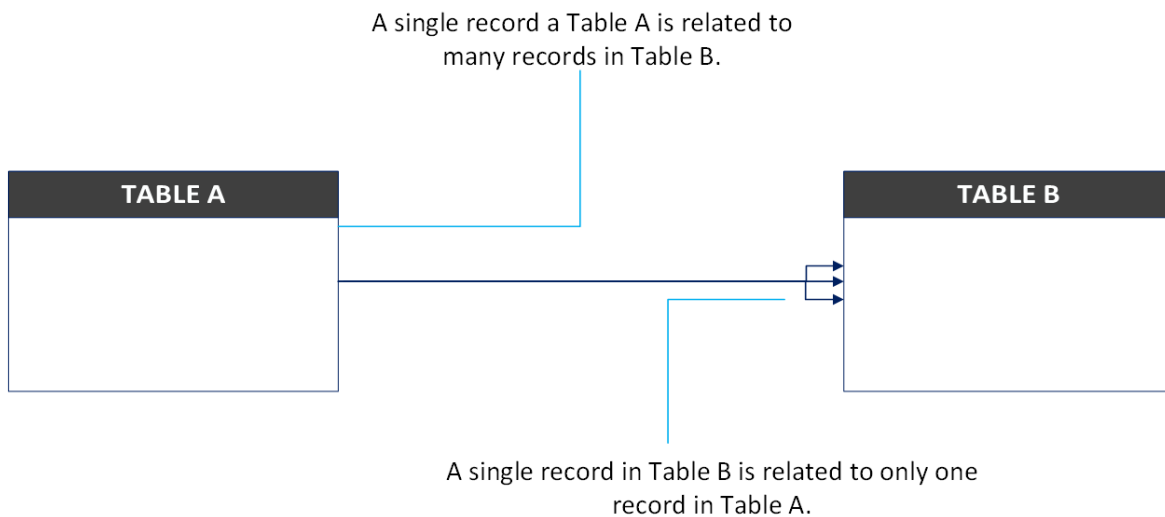
### 1:N Entity Relationships

#### Introduction

A one-to-many relationship is not a property of the data, but a property of the relationship itself.

The one-to-many relationship is only a principle of relational database design. It cannot be explicitly defined in the database structure, but created and enforced by the use of relationships between a parent entity and child entities. One record in a table can be associated with one or more records in another table:

## One-to-many Relationship



In a one-to-many relationship, the parent entity might have zero, one or multiple child records while a child entity can have one and only one parent entity record.

In FintechOS Studio, you can create one-to-many relationships by adding a lookup attribute on the child entity to reference the parent entity. When adding a lookup attribute, the **Lookup Relationship Name** field is automatically filled in by the system with the concatenation of the two entity names, following this naming pattern: **ChildEntityName\_ParentEntityName**. It allows you to display on entity the following:

- the lookup attribute referencing the parent on child data form
- the list with child records on parent entity

### Create a one-to-many relationship

This section provides the use case scenario for recording customer sales transactions over time. The database will contain two tables, as follows:

- The Investor table used to keep the customer details. The table primary key is the **InvestorId** column.

- The Assets table used to keep the track of individual investor assets. It contains the assetOwner foreign key which references the Investorid column in the Investor table to track the customer to whom the asset belongs.

A single asset can only be associated to one investor, while one investor can have many assets. The logic is defined by the one-to-many relationship:

To create a one investor-to-many assets transactions, you need to create a lookup attribute on the Assets entity. To do so, follow these steps:

1. From the menu, click **Evolutive Data Core > Data Model Explorer**. The Business Entities List page appears.
2. Search for the 'Assets' entity and double-click it. The Edit Business Entity page displays.
3. Scroll-down to the **Data Model** section and click the section header. The Data Model section expands.
4. Click **Insert** . The Add Attribute page displays.
5. Enter the attribute properties as follows:
  - **Name** - assetOwner
  - **Attribute Type** - select **Lookup**
  - **Display Name** - Asset Owner
  - **Lookup to Entity** - Click the down arrow, select **Investor** and double-click it.
  - **Lookup Relationship Name** - **assets\_assetOwner\_Investor** is automatically filled in after making the entity selection in the **Lookup to entity** field.
6. Click **Save and close**.

The newly created relationship is displayed on both entities, as follows:

- On the 'assets' entity, the relationship is displayed within the Relationships Referenced section, as this section lists the relationships where the current entity is a child.

▼ Relationships referenced

+ Insert X Delete Export Refresh

Name	Referencing Entity
Asset_assetOwner_Investor	Investor
ebis_Asset_businessunit	businessunit
ebis_Asset_createdsystemuser	systemuser
ebis_Asset_entitystatus	entitystatus
ebis_Asset_modifiedsystemuser	systemuser
ebis_Asset_systemuser	systemuser

- On the 'Investor' entity, it is displayed within the Relationships Referencing section, as this section lists the relationships where the current entity is a parent.

▼ Relationships referencing

+ Insert X Delete Export Refresh

Name	Referenced Entity
Asset_assetOwner_Investor	Asset

## N:N Entity Relationships

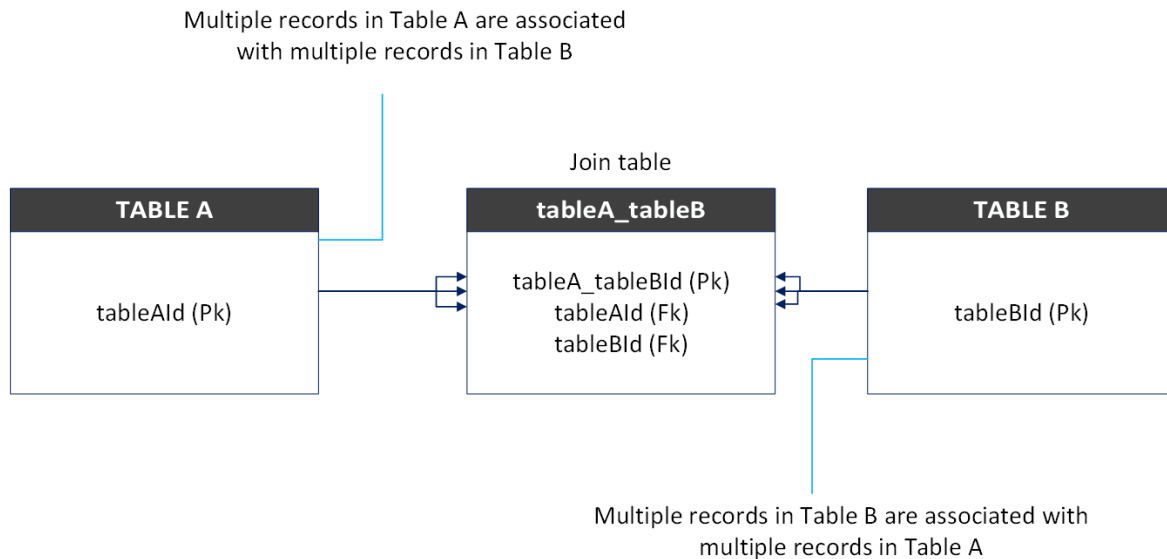
### Introduction

The N:N entity relationship is used to relate many records of one entity to many records of another entity.

Relational databases do not allow implementing a direct many-to-many relationship between two tables, therefore you cannot create a direct N:N relationship between two entities.

To create a N:N entity relationship, you will need to use an intersect/join entity (table) and create two one-to-many relationships between the two entities and the join entity:

## Many-to-many Relationship



Each record in a join table includes a match field (foreign key) that contains the value of the primary keys of the two tables it joins. The foreign key fields are populated with data when records in the join table are created from either table it joins.

Example of a N:N entity relationship is the relationship between employees and departments. Each department can have any number of employees working on a specific task. Also, an employee can work for multiple departments at the same time. Therefore, any number of departments or employees can simultaneously be linked to each other by creating a join table (Employee\_Department) that links them using the Employeeid and the Departmentid.

### Create a many-to-many relationship

In FintechOS Studio you can create a many-to-many relationship between two entities by adding a relationship referencing another entity and entering value **2** in the Relationship Type field.

When adding a Relationship referencing another entity you can choose how the system will behave when deleting records which belong to entities linked through relationship. To do so, you will have to enter one of the following values in the Relationship Constraint (0-none, 1-restrict, 2-cascade) field:

- **0** - There will be no restrictions on deleting entity records that have been linked by other entities. When deleting the entity record, the value null will be displayed in that field for all entities linked to that entity. For example, if deleting a product, then all the customers who had the deleted product will have the value "null" in the field corresponding to that specific product.
- **Restrict** – Restricts the deletion of entity records that have been linked by other entities. For example, you are not able to remove a product that is owned by a customer.
- **Cascade** – Allows the deletion of entity records, including recursive cascades.

### Use case scenario

Consider a database for recording customer products over time. The database will contain two tables, as follows:

- The Account table used to keep the customer details. The table primary key is the Accountid column.
- The Product table used to keep the track of individual products. The table primary key is the Productid column.

A many-to-many relationship exists between customers and products: customers can purchase various products, and products can be purchased by many customers.

To create a many customers-to-many products relationship, follow these steps:

1. From the menu, click **Evolutive Data Core > Data Model Explorer**. The **Business Entities List** page appears.
2. Search for the **Product** entity and double-click it. The **Edit Business Entity** page will be displayed.
3. Scroll-down to the **Relationship Referencing** section and click on the section header. The Relationship Referencing section expands.
4. Click **Insert**. The **Add Relationship** page will be displayed.

## 5. Enter the relationship properties as follows:

Property	Value
Name (only for view)	<b>Product_Name</b> . The value in this field should follow this naming convention: concatenation of the two entity names, <b>ChildEntity_PK_ParentEntity</b> .
Display Name	Customer
Referenced Entity	Click the down arrow and a new page displays. Select <b>Account</b> and double-click it.
Referencing Entity (only for view)	<b>Product</b> is automatically filled in this field (being the current entity you're working on, the child entity).
Relationship Type (only for view) (0-many to one, 1-one to many, 2-many to many)	Type <b>2</b> .
Relationship Constraint (0-none, 1-restrict, 2-cascade)	Type the desired value (0, 1 or 2) based on your business needs.

Add Relationship

Name (only for view)	<input type="text" value="Contract_X_Fees"/>
DisplayName	<input type="text" value="ContractxFees"/>
Referenced Entity (only for view)	<input type="text" value="Asset"/> ⌵ ↗
Referencing Entity (only for view)	<input type="text" value="Address"/> ⌵ ↗
Relationship Type (only for view) (0-many to one, 1-one to many, 2-many to many)	<input type="text" value="2"/>
Relationship Constraint (0-none, 1-restrict, 2-cascade)	<input type="text" value="1"/>

6. Click **Save and close**. The lookup attribute will be saved.

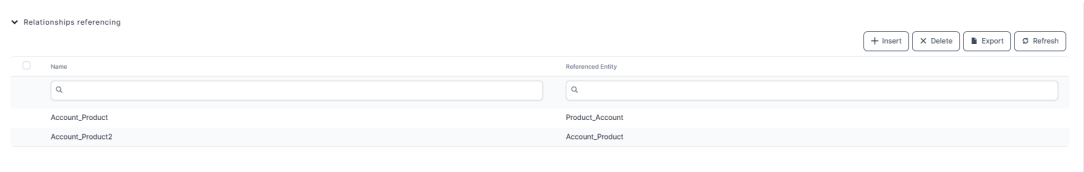
The many-to-many relationship is provisioned in the database and becomes visible in the user interface. Table **Product\_Account** will be the link table between **Account** and **Product**.

To see the details of the relationship entity, go to the Entities list (Evolutive Data > Data Model Explorer), search for the **Product\_Account** entity and double-click it. The Edit entity page will be displayed.

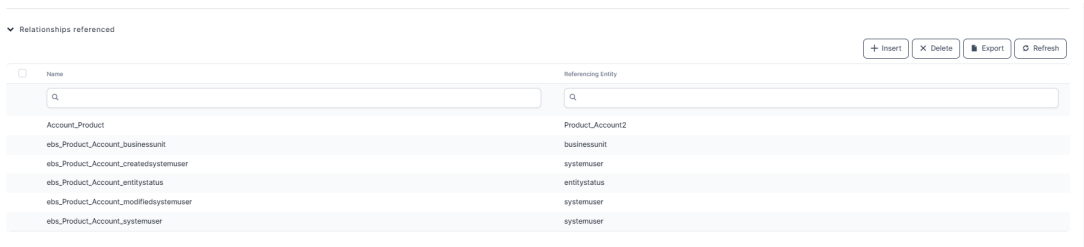
Scroll-down to the Data Model section and click on the section header. The Data Model section expands. It contains the relationship entity primary key attribute (**Product\_Accountid**) and two foreign keys (lookup attributes) to the other two entities (**Productid** and **Accountid**):

The newly created relationship entity is displayed on both entities (Product and Account), as follows:

- On the Product entity the relationship entity (**Product\_Account**) automatically created by the system is displayed in the Relationship referencing section: Both Product and Account entities reference the **Product\_Account** table in the database.



- On the Account entity the relationship entity (**Product\_Account**) is displayed in the Relationship referencing section. The referenced entity the **Product\_Account** join table, corresponding to the many-to-many relationship between the Product and Account entities.




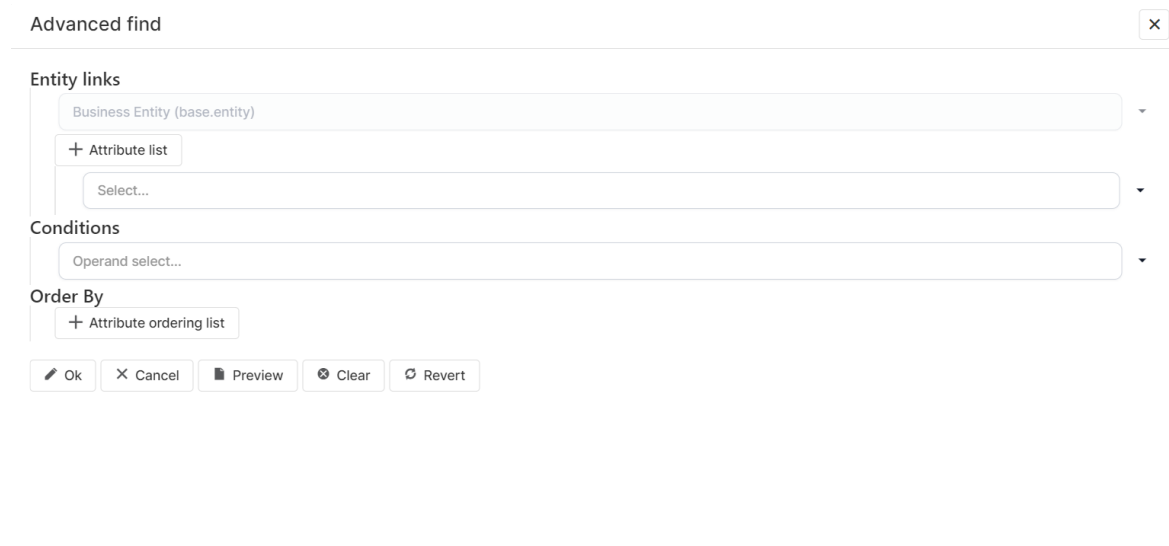
This is how the Product and Account tables will look like in the database when filled with data:

ACCOUNT		Product_Account			Product	
Accountid	Name	Product_Accountid	Accountid	Productid	Productid	Name
1	John Doe	21	1	147	147	Credit card
2	Bruce Wayne	98	1	694	852	Current Account
3	Anne Jacobson	54	3	852	694	Deposit
4	Thomas Hayes	32	2	147	367	Personal Loan

## Advanced Entity Find

Advanced Find is a powerful DB segmentation tool which allows you to filter and customize the entity grid displayed in the Data Model Explorer.

To access Advanced Find, in the "[Data Model Explorer](#)" on [page 57](#), click **Advanced find** (  ). The Advanced find page will be displayed similarly to the figure below. The page allows you to choose the criteria and conditions for displaying your entities.



Advanced find

Entity links

Business Entity (base.entity)

+ Attribute list

Select...

Conditions

Operand select...

Order By

+ Attribute ordering list

Ok Cancel Preview Clear Revert

### 1 Select the attributes you wish to display

Click **Attribute list** to open the Attributes List page. This page lists the attributes that all entities have in common.

entity attributes ✕

Attribute	Attribute type	Select	Alias
Authoring Type	Plain field	<input type="checkbox"/>	
Business Workflow	Plain field	<input type="checkbox"/>	
Customization Set Id	Plain field	<input type="checkbox"/>	
Optimization Search Data (Filter starts with)	Plain field	<input type="checkbox"/>	
Default Entity Status	Plain field	<input checked="" type="checkbox"/>	Default Status
Description	Plain field	<input type="checkbox"/>	
DisplayCollectionName	Plain field	<input type="checkbox"/>	
DisplayName	Plain field	<input type="checkbox"/>	
entityId	Plain field	<input type="checkbox"/>	
EntityMenuSection	Plain field	<input type="checkbox"/>	
Entity Type	Plain field	<input type="checkbox"/>	
IconUrl	Plain field	<input type="checkbox"/>	

Select the attributes you want to include in the fetch by clicking the corresponding checkbox in the **Select** column.

For all selected attributes, in the **Alias** column, type a unique alias name.

### IMPORTANT!

The alias is used by the system to distinguish between various records; therefore, providing a unique alias for all selected attributes is mandatory. Do not use the same alias for two different attributes.

Scroll-down at the bottom of the attributes list and click **OK**.

## 2 Apply filtering conditions

From the Conditions section you can select conditions which must be met in order to include a record into the data fetched from the database.

You can apply multiple conditions which need to be met separately or in combination by selecting either AND or OR as logical operator.

To set up the fetch entity data conditions, follow these steps:

1. From the drop-down field displayed in the Conditions section, select the logical operator:

**AND** – to apply multiple conditions which all need to be met.

**OR** – to apply multiple conditions when at least one condition needs to be met.

You can apply multiple conditions by using as many logical operators as you need. Once you select one, another logical operator field will be displayed beneath the first one, and so on.

2. Select the entity on which the condition is to be applied.

### NOTE

In Advanced Find, you can only select the generic base entity.

After selecting an entity, the **Attribute select** field will be displayed below the selected entity.

3. Click the **Attribute select** field - it will list only the attributes that all entities have in common.
4. Select an attribute of the entity previously selected. Below the selected attribute, the **Select operand** field will be displayed. The table below describes the operands you can choose from.

Operand	Description
Contains	Records will be included in the data fetched from the database if the value of the selected attribute contains the value given in the field displayed below the selected operand field.
NotContains	Records will be included in the data fetched from the database if the value of the selected attribute does not contain the value given in the field displayed below the selected operand field.
Equals	Records will be included in the data fetched from the database if the value of the selected attribute equals the value given in the field displayed below the selected operand field.
NotEquals	Records will be included in the data fetched from the database if the value of the selected attribute is different from the value given in the field displayed below the selected operand field.
Greater Than	Records will be included in the data fetched from the database if the value of the selected attribute is greater than the value given in the field displayed below the selected operand field.
Greater Than or Equal	Records will be included in the data fetched from the database if the value of the selected attribute is greater than or equal to the value given in the field displayed below the selected operand field.
Lower Than	Records will be included in the data fetched from the database if the value of the selected attribute is lower than the value given in the field displayed below the selected operand field.
Lower Than Equal	Records will be included in the data fetched from the database if the value of the selected attribute is lower than or equal to the value given in the field displayed below the selected operand field.

Operand	Description
IsNotNull	Records will be included in the data fetched from the database if the value of the selected attribute does not contain a null value.
IsNull	Records will be included in the data fetched from the database if the value of the selected attribute has a null value.
Last X Days	Records from the last 'x' days will be included in the data fetched from database. Where x is the number of days specified in the field displayed below the operand.
Last X Months	Records from the last 'x' months will be included in the data fetched from database. Where x is the number of months specified in the field displayed below the operand.
Last X Years	Records from the last 'x' years will be included in the data fetched from database. Where x is the number of years specified in the field displayed below the operand.
Last X Weeks	Records from the last 'x' weeks will be included in the data fetched from database. Where x is the number of weeks specified in the field displayed below the operand.
Next X Days	Records which have the creation date greater than today but less than x days from today. Where x is the number of days specified in the field displayed below the operand.
Next X Months	Records which have the creation date greater than today but less than x months from today. Where x is the number of months specified in the field displayed below the operand.
Next X Years	Records which have the creation date greater than today but less than x years from today. Where x is the number of years specified in the field displayed below the operand.

5. Select an operand from the list.
6. If needed, in the text field below the operand field, enter the match value for the selected operand.
7. Repeat for any additional filtering criteria.

For example, the criteria below will filter entities that have either the FTOM\_MKT\_Audience or the FTOS\_MKT\_Campaign business workflows attached.

Advanced find

Select...

Conditions

Or

Operand select...

Business Entity (base.entity)

Business Workflow

Equals

FTOS\_MKT\_Audience

Business Entity (base.entity)

Business Workflow

Equals

FTOS\_MKT\_Campaign

Entity select...

Before saving the fetch, we recommend you to preview the fetch and validate the fetch results.

### 3 Order results

From the **Order By** section, the returned results can be arranged in ascending or descending order.

To order the records:

1. Click **+Attribute ordering list**.
2. Click the **Entity** field and choose one.
3. Click the **Attribute** field and choose one.
4. Click the **Order type** field and choose **Asc** or **Desc** (ascending/descending), then click **+Add ordering** to save the selection.
5. Click **Ok** to return to the **Advanced find** page.

Attribute ordering list

Entity

Business Entity

Attribute

Authoring Type

Order type

Asc

Asc

Desc

+ Add ordering

Ok

Cancel

**NOTE**  
The attribute types from the drop-down list are based on the entity attributes selected.

4 Validate the Fetch Results

To validate the fetch results, preview them by clicking **Preview**. The figure below is a preview example of data fetched from the Account, Contract and Product entities.

Preview

Country	Account	Type	Age	Revenue	Contract	Name	Product
Q	Q	Q	Q	Q	Q	Q	Q
	6e2fa89c-2476-4396...	ba60b23f-7437-44ee...					
Romania	81f1345d-b5be-4187...	ba60b23f-7437-44ee...					
Romania	8cbfb44e-9c0e-4552...	ba60b23f-7437-44ee...					

Ok

Validate the results, then click **Ok** to close the page.

To filter the fetch, in the **Advanced find** page, click **Ok**.

## Data Views

A view is similar to a table from the database. It displays specific data in columns and rows as referenced in the query that defines the view. The columns and rows are dynamically generated when the view is produced.

The view query can fetch entity data coming from different tables within the database. For details on how to fetch entity data, see [Fetch Entity Data](#).

### View basic action handlers

The default view comes with the basic action handlers enabled.

The basic action handlers for views are:


- Insert - allows you to add a new entity record, opens the data form defined as default for insert.
- Delete - allows you to delete the view rows marked to be deleted (applicable only if the user has specific rights). On click, a confirmation pop-up prompts you to confirm deletion.
- Export - allows you to export view data as Excel, PDF (v21.2.2 or later), or CSV (v21.2.2 or later) files .


For CSV files, you can also set the formatting options.

Export current set

Export all data set

Export as


  
Excel


  
CSV

Show Advanced Settings
  
☒

First row as header
  
☐

Column Separator
  
Comma (,)

Text Qualifier
  
Quotes (")

Use local format ⓘ
  
☒

✕ Close

↓ Export

For PDF files, you must also provide a digital document as a template for the exported PDF (see ["Provide View General Information" on page 115](#) for details).

Two export options are available:

- **Export current set** - exports the view data displayed within current page.
- **Export all data set** - exports all data listed in view.

**NOTE**

When exporting view data, the generated .xlsx file is saved into the "UploadEbs\temp" folder.

- Advanced find - allows searching and filtering data based on customized criteria. For details, see ["Advanced Entity Find" on page 105](#).

Action results generate status messages displayed at the bottom of the page.

Create and Design Views

Introduction

An entity can have many views defined, but only one will be used as default without mentioning the name. Other views could be used to display data through relationships or through lookup view option.

Each entity has a default auto-generated view, which displays by default only the primary attribute (defined when the entity was created), but can be customized at any time.

Add Views

The view configuration page allows you to create a new view or edit an existing one.

To add a view to an entity, follow these steps:

1. Go to the entity edit page, scroll-down to the **Data Views** section, and click the section header. The Data Views section expands.

The screenshot shows the 'Data Views' section expanded. At the top, there are buttons for '+ Insert', 'X Delete', 'Export', and 'Refresh'. Below these is a table with columns: 'Entity', 'View Name', 'Is Default', and 'Digital Asset'. The 'Entity' column has a search bar and a dropdown menu. The 'View Name' column has a search bar. The 'Is Default' column has a dropdown menu with '(All)' selected. The 'Digital Asset' column has a search bar. The table contains one row with the following data: 'afe1\_BW' in the Entity column, 'default' in the View Name column, a checked checkbox in the Is Default column, and an empty field in the Digital Asset column.

Entity	View Name	Is Default	Digital Asset
afe1_BW	default	<input checked="" type="checkbox"/>	

2. Click **Insert** . The view configuration page appears. It is comprised of three sections and the General tab will be displayed by default. You can navigate through the view

configuration sections by clicking the tabs.

The screenshot shows the 'General' tab of a configuration interface. It contains the following fields and controls:

- Name:** A text input field with a red asterisk indicating it is required.
- Display Name:** A text input field.
- Show Display Name As Title:** A checkbox that is checked.
- IsDefault:** An unchecked checkbox.
- ShowViewButton:** An unchecked checkbox.
- Digital Document Id:** A text input field with a dropdown arrow on the right.

3. "Provide View General Information" below

4. "Define View Data" below

You can also enable Inline Editing for view and Apply Conditional HTML Formatting.

## Provide View General Information

The **General** tab allows you to provide a name for the new entity view and choose whether it is set as default view and also add a **View** button to all the view entries.

Property	Description
IsDefault	If selected, current view will be displayed when users access the specific entity. If you want to display on a data form a view that is not defined as default, add the view name as a token on the data form <b>{#RelationshipName, view: viewName#.}</b> . By doing so, the view specified by the token provided will be displayed on the entity data form instead of the default entity view.
ShowViewButton	If selected, adds the View column on the entity view which contains a View button. When clicked, the link redirects the users to the details page of the specific record.
Digital Document ID (v21.2.2 or later)	Digital document template to be used as a template if you wish to export the data view in PDF format. For more information, see the <a href="#">Digital Documents Processor documentation</a> .

In order to define the view data, you should first provide the view general information and save the view by clicking **Save and reload**.

## Define View Data

The Data tab allows you to fetch entity data that will be displayed on the view, filter it, sort it, and then decide how it is going to be presented in the data view's grid.

**NOTE**

You have to save the view before fetching entity data.

To define what data is shown in the view and the sorting order of records, follow these steps:

**1 Fetch Entity Data**

Fetching data allows you to filter entity records displayed in view based on specific criteria (e.g business status, security role).

There are multiple ways in which you can fetch entity data:

## Use the Data Field

If the data you need to fetch for your view is not too complex (it is limited only to attributes from your business entity) you can use the Data field to retrieve your data.

In the Data field, write the names of the entity attributes which will define the view columns. Use commas to separate attribute names.

Data

name,attributeld,includeInAudienceSegmentData,description,

## Fetch Entity Data Using the Fetch Designer

In the **Data** section:

1. **Check Out** the Fetch Object Expression
2. Click **Show Fetch Designer**.

The **Advanced Find** page is displayed.

**NOTE**

If you don't check out the fetch object expression, the Show Fetch Designer button is not visible.

### Choose Entity Links

In the Advanced find page, the entity from which you triggered the Advanced search is displayed by default in the Entity Links field and it is non-editable.

### Select the entities

Click the Select field. The list of entities that are linked to the default entity will be displayed.

Advanced find

Entity links

Customer (base.Account)

+ Attribute list

Select...

Account Type - Account Type

Accountid - Address

Accountid - FTOS\_ACC\_AccountFunctionality\_X\_Account

accountid - FTOS\_CMB\_Action

accountid - FTOS\_CMB\_Activity

Accountid - FTOS\_IP\_Insurer

Accountid - FTOS\_IP\_ResellerChannel

Accountid - FTOS\_IP\_ResellerChannel\_Account

accountid - FTOS\_MKT\_Activity

accountid - FTOS\_MKT\_AudienceListMember

accountid - FTOS\_MKT\_TeamMember

Accountid - MKT\_Activity

Accountid - MKT\_MarketingListMember

Select the desired entity from which you want to gather data. Another empty field will be displayed which allows you to select another entity in case you want to fetch data from various entities.

Advanced find

Entity links

Customer (base.Account)

+ Attribute list

Accountid - Address

+ Attribute list

Select...

Select...

Conditions

And

Operand select...

Entity select...

Order By

+ Attribute ordering list

Ok Cancel Preview Refresh

You can add as many entities as you need to extract data from. For example, to create profiles for tailored price offerings and communication, you need to fetch data from the following entities: Account, Product and Contract.

Advanced find ×

Entity links

Customer (base.Account)

+ Attribute list

ContractId - Contract

+ Attribute list

Select...

Product - Product

+ Attribute list

Select...

Select...

Conditions

Or

Operand select...

Select the attributes

After selecting all the entities from where you want to extract information, for each entity, select the attributes to be included in the fetch. To do so, click the **Attribute list** button displayed below each entity. The Attributes list page will be displayed. It lists all the attributes existing on that entity. The figure below shows the attributes of the Account entity:

Account attributes ×

Attribute	Attribute type	Select	Alias
Country	Plain field	<input type="checkbox"/>	
Accountid	Plain field	<input type="checkbox"/>	
Customer picture	Plain field	<input type="checkbox"/>	
Customer Type	Plain field	<input type="checkbox"/>	
Age	Plain field	<input type="checkbox"/>	
Annual revenues	Plain field	<input type="checkbox"/>	
Branch	Plain field	<input type="checkbox"/>	
Business Status	Plain field	<input type="checkbox"/>	
Business Unit	Plain field	<input type="checkbox"/>	
Citizenship	Plain field	<input type="checkbox"/>	
City	Plain field	<input type="checkbox"/>	
Commercial registration number	Plain field	<input type="checkbox"/>	

Select the attributes you want to include in the fetch by ticking the corresponding checkbox in the Select column.

For all selected attributes, in the Alias column, type a unique alias name.

### IMPORTANT!

The alias is used by the system to distinguish between various records; therefore, providing a unique alias for all selected attributes is mandatory. Do not use the same alias name for two attributes of the same or different entities. For example, you should not use the "Mobile" for an attribute existing on the entity "Account" and the "Mobile" alias for an attribute on the entity "Campaign".

Optionally, you can select how the attribute value will be displayed, by choosing one of the values available in the Attribute type drop-down list:

- **Plain field** - displays the actual value of the attribute. It is selected by default for all attributes.
- **Sum field** - displays the sum of the attribute values.
- **Max field** - displays the maximum of the attribute values.
- **Count field** - displays the number of the attribute values.

### Account Attributes

Account attributes				×
Attribute	Attribute type	Select	Alias	
Country	Plain field	<input checked="" type="checkbox"/>	Country	
Accountid	Plain field	<input checked="" type="checkbox"/>	Account	
Customer picture	Plain field	<input type="checkbox"/>		
Customer Type	Plain field	<input checked="" type="checkbox"/>	Type	
Age	Plain field	<input checked="" type="checkbox"/>	Age	
Annual revenues	Plain field	<input checked="" type="checkbox"/>	Revenue	
Branch	Plain field	<input type="checkbox"/>		
Business Status	Plain field	<input type="checkbox"/>		
Business Unit	Plain field	<input type="checkbox"/>		
Citizenship	Plain field	<input type="checkbox"/>		
City	Plain field	<input type="checkbox"/>		
Commercial registration number	Plain field	<input type="checkbox"/>		

### Contract Attributes

Contract attributes ✕

Attribute	Attribute type	Select	Alias
Product	Plain field	<input type="checkbox"/>	
Business Unit	Plain field	<input type="checkbox"/>	
Contractid	Plain field	<input checked="" type="checkbox"/>	Contract
Created by user	Plain field	<input type="checkbox"/>	
Created On	Plain field	<input type="checkbox"/>	
Status	Plain field	<input type="checkbox"/>	
Modified by user	Plain field	<input type="checkbox"/>	
Modified On	Plain field	<input type="checkbox"/>	
Name	Plain field	<input type="checkbox"/>	
Start Date	Plain field	<input type="checkbox"/>	
User	Plain field	<input type="checkbox"/>	

Ok Cancel

## Product Attributes

product attributes ✕

Attribute	Attribute type	Select	Alias
Customer	Plain field	<input type="checkbox"/>	
Business Unit	Plain field	<input type="checkbox"/>	
Created by user	Plain field	<input type="checkbox"/>	
Created On	Plain field	<input type="checkbox"/>	
Status	Plain field	<input type="checkbox"/>	
Modified by user	Plain field	<input type="checkbox"/>	
Modified On	Plain field	<input type="checkbox"/>	
name	Plain field	<input checked="" type="checkbox"/>	Name
productid	Plain field	<input checked="" type="checkbox"/>	Product
User	Plain field	<input type="checkbox"/>	

Ok Cancel

Scroll-down at the bottom of the attributes list and click **OK**. The page listing the entity's attributes will be closed.

Now you have to set up the fetch conditions.

### Apply Filtering Conditions

From the Conditions section you need to select the conditions which must be met in order to include a record into the data fetched from the database.

You can apply multiple conditions which need to be met separately or in combination by selecting either AND or OR as logical operator.

To set up the fetch entity data conditions, follow these steps:

1. From the drop-down field displayed in the Conditions section, select the logical operator:

Advanced find

Entity links

Customer (base.Account)

+ Attribute list

ContractId - Contract

+ Attribute list

Select...

Select...

Conditions

Operand select...

And

Or

**AND** – to apply multiple conditions which need to be met separately.

**OR** – to apply multiple conditions when at least one condition needs to be met.

You can apply multiple conditions by using as many logical operators as you need. Once you select one, another logical operator field will be displayed beneath it.

2. Select the entity on which the condition is to be applied.

Advanced find

+ Attribute list

ContractId - Contract

+ Attribute list

Select...

Select...

Conditions

And

Operand select...

Customer (base.Account)

Customer (base.Account)

ContractId - Contract

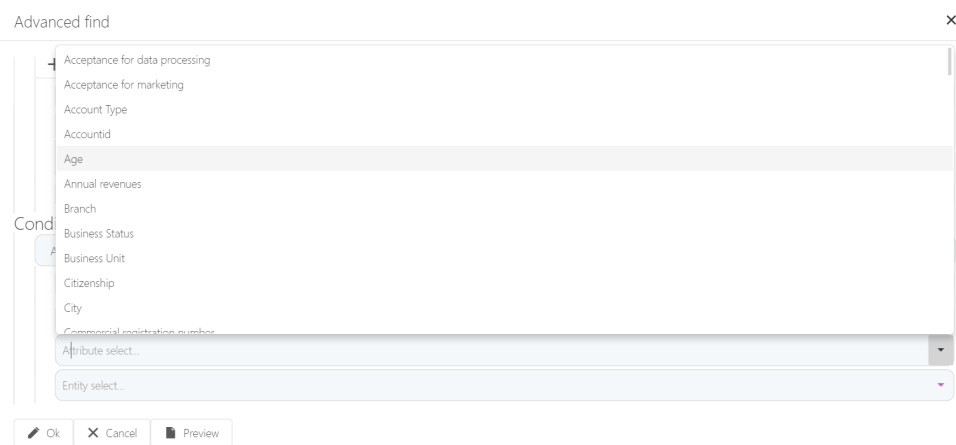
Ok Cancel Preview

**NOTE**

You can only select from the list of chosen entities on which the database segmentation will be done (selected in the Entity Links section). For information on how to choose entities for fetching the data, see ["Advanced Entity Find" on page 105](#).

After selecting an entity, the **Attribute select** field will be displayed below the selected entity.

- 3. Click in the **Attribute select** field. The list shows qonly the entity attributes selected for the fetch.



- 4. Select an attribute existing on the entity previously selected. Below the selected attribute, the Select operand field will be displayed.

Advanced find ×

+ Attribute list

Contractid - Contract

+ Attribute list

Select...

Select...

Conditions

And

Operand select...

Customer (base.Account)

Accountid

Operand select...

Entity select...

The table below describes the operands you can choose from.

Operands	Description
Contains	Records will be included into the data fetched from the database if the value of the selected attribute contains the value given in the field displayed below the selected operand field.
NotContains	Records will be included into the data fetched from the database if the value of the selected attribute does not contain the value given in the field displayed below the selected operand field.
Equals	Records will be included into the data fetched from the database if the value of the selected attribute equals the value given in the field displayed below the selected operand field.
NotEquals	Records will be included into the data fetched from the database if the value of the selected attribute is different than the value given in the field displayed below the selected operand field.

Operands	Description
Greater Than	Records will be included into the data fetched from the database if the value of the selected attribute is greater than the value given in the field displayed below the selected operand field.
Greater Than or Equal	Records will be included into the data fetched from the database if the value of the selected attribute is greater than or equal to the value given in the field displayed below the selected operand field.
Lower Than	Records will be included into the data fetched from the database if the value of the selected attribute is lower than the value given in the field displayed below the selected operand field.
Lower Than Equal	Records will be included into the data fetched from the database if the value of the selected attribute is lower than or equal to the value given in the field displayed below the selected operand field.
IsNotNull	Records will be included into the data fetched from the database if the value of the selected attribute does not contain a null value.
IsNull	Records will be included into the data fetched from the database if the value of the selected attribute has a null value.
Last X Days	Records from the last 'x' days will be included in the data fetched from database. Where x is the number of days specified in the field displayed below the operand.

Operands	Description
Last X Months	Records from the last 'x' months will be included in the data fetched from database. Where x is the number of months specified in the field displayed below the operand.
Last X Years	Records from the last 'x' years will be included in the data fetched from database. Where x is the number of years specified in the field displayed below the operand.
Last X Weeks	Records from the last 'x' weeks will be included in the data fetched from database. Where x is the number of weeks specified in the field displayed below the operand.
Next X Days	Records which have the creation date greater than today but less than x days from today. Where x is the number of days specified in the field displayed below the operand.
Next X Months	Records which have the creation date greater than today but less than x months from today. Where x is the number of months specified in the field displayed below the operand.
Next X Years	Records which have the creation date greater than today but less than x years from today. Where x is the number of years specified in the field displayed below the operand.

- Click in the Select operand field. A drop-down list with all available operands will be displayed.

Advanced find

+ Attribute list

ContractId - Contract

+ Attribute list

Select...

Select...

Contains

NotContains

Equals

NotEquals

GreaterThan

Greater Than Or Equal

Operand select...

Entity select...

6. Select an operand from the list.
7. If needed, in the text field below the operand field, enter the match value for the selected operand.

Advanced find

Conditions

Or

Operand select...

Customer (base.Account)

Accountid

Is Not Null

Customer (base.Account)

Country

Is Not Null

Customer (base.Account)

Customer Type

Before saving the fetch, we recommend you to preview the fetch and validate the fetch results.

#### Validate the Fetch Results

To validate the fetch results, preview them by clicking Preview in the Advanced find page. The figure below is a preview example of data fetched from the Account, Contract and Product entities.

Preview ✕

Country	Account	Type	Age	Revenue	Contract	Name	Product
Q	Q	Q	Q	Q	Q	Q	Q
	6e2fa89c-2476-4396...	ba60b23f-7437-44ee...					
Romania	81f1345d-b5be-4187...	ba60b23f-7437-44ee...					
Romania	8cbfb44e-9c0e-4552...	ba60b23f-7437-44ee...					

✕ Ok

Validate the results, then click **OK** to close the page.

To filter the fetch, in the Advanced find page, click **OK**.

# Fetch Entity Data Using the Fetch Object Expression Field

In the Fetch Object Expression field, provide the custom fetch expression, that is a JavaScript object. It might query and return attributes from various entities.

Fetch Object Expression Show Fetch Designer

Release Check Out

Current version number: 1. CHECKED OUT History

```
7      "where": {
8        "type": "and"
9      },
10     "orderby": [
11       {
12         "attribute": "base.businessUnitId",
13         "type": "asc"
14       },
15       {
16         "attribute": "base.userId",
17         "type": "asc"
18       }
19     ]
20   }
```

**IMPORTANT!**  
Make sure you always define aliases for fetched entities, otherwise the fetch will not work.

When joining several entities together using the Fetch Object Expression field, the first entity in the fetch expression should have the alias 'a'.

Below is an example on how the fetch expression should look like:

```
Joined entities alias
return {
  "entity": {
    "alias": "a",
    "name": "AccountRelEmployer",
    "attributelist": [
      {
        "name": "AccountRelEmployerid"
      },
      {
        "name": "Details",
      }
    ],
    "join": [
      {
        "type": "left",
        "entity": {
          "alias": "entity1",
          "name": "Account",
          "attributelist": [
            {
              "name": "Name",
              "attributeType": 3
            }
          ]
        },
        "fromto": [
          {
            "from": "ReferencedAccountId",
            "to": "Accountid"
          }
        ]
      }
    ],
    "type": "left",
    "entity": {
      "alias": "osi",
      "name": "optionsetitem",
    }
  }
}
```

```

        "attributelist": [
            {
                "name": "name",
                Fintech OS - UI Designer
                "attributeType": 3
            }
        ]
    },
    "fromto": [
        {
            "from": "EmploymentTypeId",
            "to": "optionsetitemid"
        }
    ]
},
]
},
where:{}
}

```

To render the view on the related entity data form, add an empty 'where' condition:

```

Empty 'where' condition
return {
    "entity": {
        "alias": "lookup",
        "name": "AccountRelEmployer",
        "attributelist": [
            {
                "name": "AccountRelEmployerid"
            },
            {
                "name": "Details",
            }
        ],
    },
    "join": [
        {
            "type": "left",
            "entity": {
                "alias": "entity1",
                "name": "Account",
                "attributelist": [
                    {

```

```

        "name": "Name",
        "attributeType": 3
      },
    ],
    },
    "fromto": [
      {
        "from": "ReferencedAccountId",
        "to": "Accountid"
      }
    ],
  },
  {
    "type": "left",
    "entity": {
      "alias": "osi",
      "name": "optionsetitem",
      "attributelist": [
        {
          "name": "name",
          "attributeType": 3
        }
      ]
    },
    "fromto": [
      {
        "from": "EmploymentTypeId",
        "to": "optionsetitemid"
      }
    ]
  },
],
},
where: {}
}

```

**IMPORTANT!**

The Fetch Designer is a visual tool that allows you to populate the Fetch Object Expression field without having to code. When you define a fetch in the Fetch Designer, any existing content in the Fetch Object Expression field will be overwritten.

## 2 Set the default sorting of the view records

By default the view records are sorted ascending by the entity primary attribute.

To change the default view records sorting, in the Sort Expression field, enter the attribute name by which the records will be sorted, followed by an asterisk (\*) symbol and the sorting direction: asc for ascendant and desc for descendant.

Sort Expression (ex.: atr1*asc,atr2*desc)	name*asc
--	----------

### NOTE

When adding a fetch expression with an Order By clause, the records are sorted by clause instead of Sort Expression.

## 3 Define the View Columns

Use the [Generate View Columns](#) button to automatically create the view columns based on your fetch (see " [1 Fetch Entity Data](#)" on page 116).

### NOTE

You can use the **Insert** and **Delete** buttons in the Entity View Columns grid to further refine which columns will be displayed and which not. The view columns added in the Entity View Columns section have a higher priority than the ones mentioned in your data fetch. For instance, if both entity view columns and data columns are defined, the view will display only the columns added in the Entity View Columns section.

To add columns from the Entity View columns, click **Insert**. In the Add Entity View Column page, provide the following view column details:

Property	Description
Entity View	The field is automatically filled in with the name of the selected view.
Width	The column width. Enter a numeric value.

Property	Description
Cell Template	<p>A custom HTML element which defines the cell layout on the entity view column. In this field you can add text, HTML or tokens.</p> <p>A token can be any attribute returned by the fetch expression. Attributes are referenced as tokens by their name prefixed by the alias of the fetch inside curly brackets (e.g., {base.Name}).</p> <p>Cell template example:</p> <pre>&lt;div onclick="window.location.hash = '/entity/MKT_Activity/edit/{a.MKT_Activityid}' " style="cursor: pointer; color: white; display: inline; float: left; border: 2px solid rgba(0, 0, 0, 0); background: #27a8e1; text-align: center; margin: 2px; padding: 2px;"&gt;POP&lt;/div&gt;</pre> <p>For more information on how to use cell templates for views, see <a href="#">Create Views using Cell Template</a>.</p>
Width Is Percentage	If selected, it indicates that the value provided in the <b>Width</b> field is percentage.
Attribute Name	<p>The name of the attribute whose values will be displayed in the column.</p> <p>You can overwrite the attribute when providing a cell template.</p>
Label	The column name that will be displayed on the entity view in the user interface.
Allow Editing	If selected, the column view can be edited.

Click **Save and close**. Add as many view columns as you need. They will be listed in the Entity View Columns section.

Entity View Columns Generate View Columns

+ Insert X Delete Export Refresh

<input type="checkbox"/>	Order <span>①</span>	Attribute Name	Label	Cell Template	Width	Percentage	Allow Editing	Disabled	View
<input type="checkbox"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="(...)"/>	<input type="text" value="(...)"/>	<input type="text" value="(All)"/>	
<input type="checkbox"/>	1	base.business...	{ Account.busi...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">View</a>
<input type="checkbox"/>	2	base.business...	{ Account.busi...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">View</a>
<input type="checkbox"/>	3	base.City	{ Account.City ...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">View</a>
<input type="checkbox"/>	4	base.CMB_Def...	{ Account.CMB...			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">View</a>

The view columns are displayed from left to right in the order of their index (OrderIdx). The first column has the OrderIdx set to 1.

To change the order in which the columns are displayed on the view, in the Entity View Columns list, drag and drop rows.

## Generate View Columns

To auto-populate the columns of a view, in the view configuration page, Data tab, scroll-down to the Entity View Columns section, click **Generate View Columns** and in the confirmation pop-up, click Yes.

Entity View Columns Generate View Columns

+ Insert X Delete Export Refresh

<input type="checkbox"/>	Order <span>①</span>	Attribute Name	Label	Cell Template	Width	Percentage	Allow Editing	Disabled	View
<input type="checkbox"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="Q"/>	<input type="text" value="(All)"/>	<input type="text" value="(All)"/>	<input type="text" value="(All)"/>	
<input type="checkbox"/>	1				0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<a href="#">View</a>

### The view has no fetch data

If the view has no fetch data, the view columns will be populated for all attributes specified in the text field if the field name does not already exist in view column. The column name will be set with the attributes label in the current language.

If you want to replace the display name of an attribute with its localized value, in the **Label** column of the entity view write the following syntax:

**{entityName.attributeName | metadata}.**

Clicking Generate View Columns will add the missing columns.

**NOTE** The system checks if the column already exists by label but it will not delete any column, you will have to manually remove columns.

#### The view has fetch data

If the view has fetch data, the view columns will be populated for all returned fields specified in the fetch (with alias to) field name that does not already exist as view column. The column name will be set with the attributes label in the current language.

**NOTE** If in the view's fetch there is an alias for the attributeName, you should use **aliasAttribute**; otherwise use **entityAlias.attributeName**.

When exporting the view using deployment packages, the label is localized so the label pattern: **{entityName.attributeName | metadata}** will be replaced with the **displayName** of the attribute.

## Inline Editing for View Records

Inline editing for views allows you to quickly make changes to view records directly in the view without opening edit forms.

You can set inline editing for views from the view configuration page, Code tab.

Click the Display Options tab and define the view display options using code similar to the following one:

```
{
  "scrollHorizontal":true,
  "allowEdit":true,
  "editMode":"cell",
  "showColumnHeaders":true,
  "showFilterRow":false,
  "pageSize":10
}
```

The table below describes the available view display options.

Display Option	Description
Display Option	Description
scrollHorizontal	If you have many columns in the view, set it to <b>true</b> .
allowEdit	To allow inline editing on the view, set it to <b>true</b> .

Display Option	Description
editMode	<p>Sets the mode of the inline editing on the view. The following values are possible.</p> <p><b>cell</b> - allows you to edit view records cell by cell.</p> <p><b>row</b> - allows you to edit a view record by editing the cells in a row then saving the view record changes.</p> <p><b>batch</b> - allows you to edit several view records and then save the changes in batch.</p> <p><b>data form</b> - allows you to edit a view record similar to a data form. When clicking on a view record, the selected record information will be displayed similar to a data form.</p>
showColumnHeaders	If you want the view to show the columns header, set it to <b>true</b> .
showFilterRow	If you want to show the filter row which allows you to filter the view records, set it to <b>true</b> .

Click **Save and close**. The view changes are saved.

The following are some examples of how inline editing looks like in the user interface based on the editMode.

#### Cell edit mode

ADDRESSES LIST					
<input type="checkbox"/>	City	Building number	Floor number	Street name	Street number
	Los Angeles	3	1	6th Avenue	3
	New York	3	3	6th Avenue	3
	Los Angeles	<input type="text" value="1"/>	5	3rd Avenue	1

#### Row edit mode

ADDRESSES LIST

☐

City

Building number

Floor number



Street name

Street number

	Los Angeles	3	1	6th Avenue	3	Edit
	New York	3	3	6th Avenue	3	Edit
	Los Ange... ↓	1	5	3rd Avenue	1	Save Cancel

Batch edit mode

ADDRESSES LIST



☐

City

Building number

Floor number

Street name

Street number

	Los Angeles	3	1	6th Avenue	3
	New York	3	3	6th Avenue	3
	Los Angeles	1	5	3rd Avenue	1

Form edit mode

ADDRESSES LIST

☐

City

Building number

Floor number

Street name

Street number

City:

Los Angeles  
↓

Building number:

3

Floor number:

1

Street name:

6th Avenue

Street number:

3

Save

Cancel

	New York	3	3	6th Avenue	3	Edit
	Los Angeles	1	5	3rd Avenue	1	Edit

TABLE OF CONTENTS

136

## Create Views using Cell Template

You can create views using cell templates from the view configuration page (Edit Business Entity > Data Views > click insert to create a new one or double-click on an existing view).

Cell template is a text editor element or a collection of elements. Using cell templates, you can create views with default or custom fetch.

### Creating views default fetch

Reference the attributes by using the fetch alias or by their name.

Cell Template

```
return {
  "entity": {
    "alias": "base",
    "name": "Account",
    "attributelist": [
      {
        "name": "Accountid"
      },
      {
        "name": "typeId"
      },
      {
        "name": "Email"
      },
      {
        "name": "EmploymentStatusId"
      },
      {
        "name": "MAI"
      },
      {
        "name": "MobilePhone"
      },
      {
        "name": "Name"
      },
      {
        "name": "PIN"
      }
    ],
    "join": [
```

```

        {
          "type": "inner",
          "entity": {
            "alias": "BWstatus",
            "name": "BWstatus",
            "attributelist": [{
              "name": "label",
              "alias": "label"
            }],
          },
          "fromto": [
            {
              "from": "businessStatusId",
              "to": "BWstatusid"
            }
          ]
        }
      ]
    }
  }
}

```

Starting with FintechOS Platform you can also fetch view data using the [Code Editor](#).

### Creating views with custom fetch

To create views with custom fetch, reference the attributes by using the custom alias:

```

//Fetch object expression text editor
return {
  "entity": {
    "alias": "base",
    "name": "Account",
    "attributelist": [
      {name: "Accountid"},
      {name: "Name"},
      {name: "UniqueID"},
      {name: "PIN"},
      {name: "FiscalRegistrationNo"},
      {name: "CommercialRegistration"},
      {name: "FirstName"},
      {name: "LastName"},
      {name: "MobilePhone"}
    ],
    "join": [{
      "type": "left",
      "entity": {

```

```

        "alias": "accType",
        "name": "FTOS_CMB_AccountType",
        "attributelist": [{name: "FTOS_CMB_AccountTypeid"}],
    {name: "name", alias: "TypeName"}]
    },
    "fromto": [{
        "from": "typeId",
        "to": "FTOS_CMB_AccountTypeid"
    }]
}
};

```

On the view, in the Code tab, After Generate JS section, provide the code to apply the desired logic.

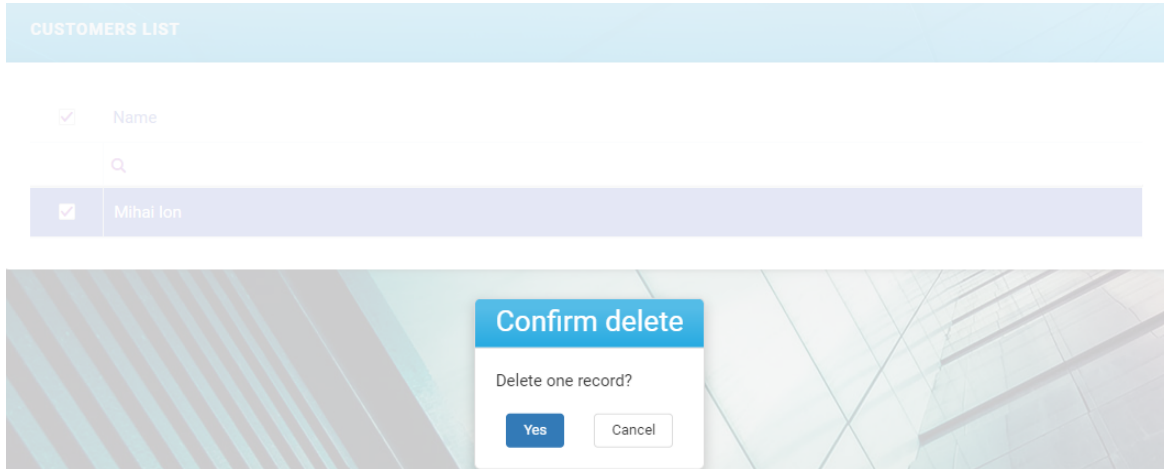
```

var view = {editFormName:"FTOS_CMB_BaseAccount",
insertFormName:"FTOS_CMB_BaseAccount"};
var gridId = 'ebsContainerContent';
var dataGrid = $('#'+ gridId).dxDataGrid('instance');
    dataGrid.option('onRowClick', function (h) {
        console.log("Custom click");
    });
var component = h.component,
    prevClickTime = component.lastClickTime;
    component.lastClickTime = new Date();
if ((prevClickTime && (component.lastClickTime - prevClickTime <
300)) || ebsIsOnMobile) {
    var nameAttr = dataGrid.option("nameAttribute");
    var key = h.key;
    var entityName = dataGrid.option("entityName");
    window.location.href = ebs.getEditNavigationUrl(entityName,
key, null, view.editFormName);
}
});
var formData = ebs.getFormData();
if(currentPageActionHandlers.find(function(x){ return x.text ==
"Insert"; })))
{
    currentPageActionHandlers.find(function(x){ return x.text
== "Insert"; }).handler = function(){
        ebs.generateInsert("ebsContainer", "Account", null, null,
"FTOS_CMB_BaseAccount");
    };
}

```

## Customize the Delete Confirmation

The figure below shows the default delete confirmation dialog for view records:



To customize the delete confirmation dialog displayed when deleting records from a specific entity view:

1. Go to the entity view configuration page, **Code** tab > **Display Options**.
2. Customize the delete confirmation dialog by updating the code provided by default.

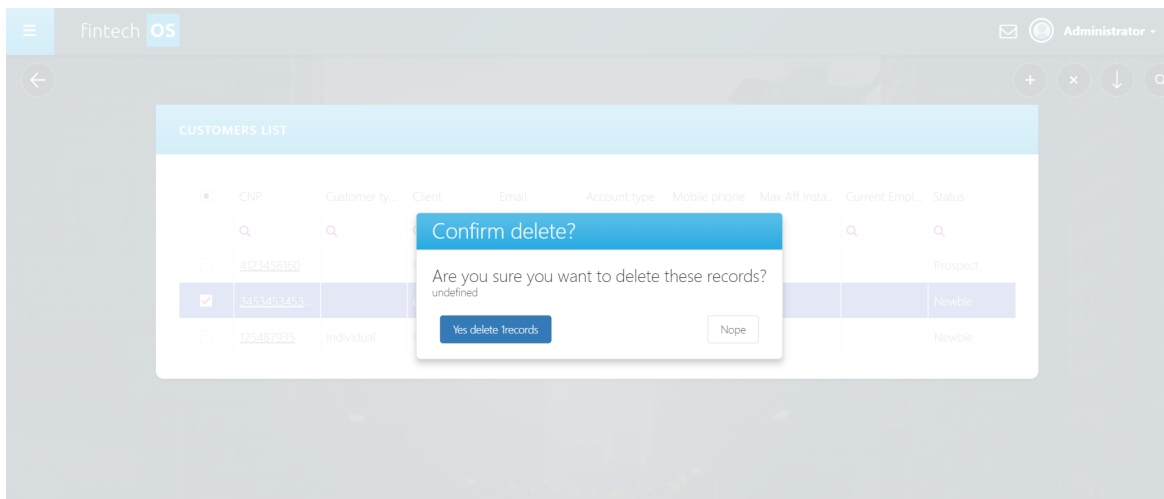
```
ebs.createViewDisplayOptionsObject({
  deleteConfirmation: {
    yesButtonLabel: "Yes, sure",
    noButtonLabel: "Nope",
    getDeleteConfirmationOptions: function (selectedRowsData,
nameAttribute) {
      //console.log(selectedRowsData, nameAttribute);
      var msg = "<h4>Are you sure you want to delete these
records?</h4>";
      var noOfRows = selectedRowsData.length;
      var cancel = false;
      for (var i = 0; i < noOfRows; i++) {
        msg += "<div>" + selectedRowsData[i][nameAttribute]
+ "</div>";
        if(selectedRowsData[i]["a_status"] == "pending")
cancel = true;
      }
      if(cancel){
```

```

        ebs.showMessage("You cannot delete pending
records", "info");
    }
    return {
        yesButtonLabel: "Yes delete " + noOfRows +
"records",
        message: msg,
        cancelDelete: cancel
    }
}
});

```

The figure below shows how the delete confirmation dialog customized above will look like:



The table below describes the properties of DeleteConfirmation.

Property	Default Value
title	\$resources.DeleteRecords_Confirmation_Title
message	\$resources.DeleteRecords_Confirmation_Message   DeleteRecords_Confirmation_Message_Singular
yesButtonLabel	\$resources.DeleteRecords_Confirmation_Yes_Label
noButtonLabel	\$resources.DeleteRecords_Confirmation_No_Label
cancelDelete	false
silentDelete	false

Setting the **cancelDelete** property to **true** will prevent the selected record(s) deletion and no confirmation will be displayed.

Setting the **silentDelete** property to **true** will delete the selected record without confirmation.

If both **cancelDelete** and **silentDelete** are set to **true**, the selected record(s) will not be deleted and no confirmation will be displayed.

## Show Loading Panel

To show the loading panel on views (which have inline editing enabled) when editing view records which trigger the execution of a workflow, go to the entity view configuration page, **Code** tab, click the **Display Options** tab and set the **showLoadingPanel** option to **true**: `"showLoadingPanel":true`

You have an entity FTOS\_CASE\_Case and a child entity, CaseXSelectedRisk2. The child entity has a view, FTOS\_CASE\_Case, which has the cell edit mode enabled. The view has records which when updated trigger the execution of a workflow on the child entity, CaseXSelectedRisk2

To show loading when changing the value of such records, on the child entity, CaseXSelectedRisk2, in the list of views, double-click the FTOS\_CASE\_Case view. In the entity view configuration page, click the **Code** tab, click the **Display Options** tab and set the **showLoadingPanel** option to **true**: `"showLoadingPanel":true`

```
{
  "allowEdit" : true,
  "editMode" : "cell",
  "showLoadingPanel" : true
}
```

To hide the loading panel, either set the **showLoadingPanel** option to **false** or remove the option from the **Display Options** tab.

## Data Forms

Forms are user interface elements that allow you to interact with individual entity records. You can access the entity forms from the Data Forms section of the entity.

Is Audited

☐

Business Workflow

Account Workflow

Optimization Search Data (Filter starts with)

☐

DATA MODEL

DATA FORMS

+ Insert

✕ Delete

📄 Export

🔄 Refresh

<input type="checkbox"/>	Entity	Name	IsDefault	Is Default For Edit	Digital Asset
<input type="checkbox"/>	Account	accountInsertForm	(All)	(All)	

You can create up to 2 forms for the same entity, depending on the different contexts in which users interact with the entity. Some forms may allow users to add records to the entity, some only to edit existing records, some may be read-only, some may have multiple pages and advanced controls and validations, etc.

**HINT**

If you wish to work with more than 2 forms for the same entity, you can use ["App Data Forms" on page 933](#) instead.

When you create an entity, a default form is created automatically which allows you to do basic record inserts and edits. The default form will be used to insert or edit records when an entity is exposed in the Digital Experience Portal or other front-ends without designating a custom form for inserts or edits. The automatically created default form uses the Auto Generate Template option (see below), but can be customized at any time. An entity can have only one default form for record inserts and one default form for record edits at a time.

System entities do not have default forms created automatically.

## Create an Entity Form

1. Expand the **Data Forms** section of an entity and click **Insert**.

Is Audited

☐

Business Workflow

Account Workflow

Optimization Search Data (Filter starts with)

☐

DATA MODEL

DATA FORMS

+ Insert

✕ Delete

📄 Export

🔄 Refresh

<input type="checkbox"/>	Entity	Name	IsDefault	Is Default For Edit	Digital Asset
	🔍	🔍	(All) ▾	(All) ▾	🔍
	Account	accountInsertForm	✔	<input type="checkbox"/>	

2. Fill in the form's general information.

1 General2 UI3 Steps4 Field Options5 Filtered Fields6 Advanced7 Security Roles

Name

accountInsertForm

Display Name

Description

Show Tooltips

User Settings

IsDefault

☒

Is Default For Edit

☐

☒ Auto Generate Template

Auto Generate Template Type

Inherit

Properties

Hide Business Workflow

Default

Read Only

Default

Disable Save Keyboard Shortcut

Default

Flow Title

Default

Style Sheets

Select items to include

Disable prompt for unsaved changes

Default

Save automatically data on leave

No

- **Name** - Name used to identify the entity's form. If the entity has another form defined (each entity can have up to two forms), this name has to be different from the other form's name.

- **Description** - Optional description of the form's function to simplify development.
- **Show Tooltips** - Select if form fields display tooltips on mouse hover. The default is to inherit the user setting.
- **Is Default** - Check to make this the default form for inserting and editing entity records when displaying entity views in the user interface. You can have only one default form for an entity at a time.
- **Is Default for Edit** - Check to make this the default form for editing entity records when displaying entity views in the user interface. You can have only one default edit form for an entity at a time.
- **Auto Generate Template** - Check to automatically generate the form fields based on the entity's attributes using a 1, 2, 3, or 4 column layout. The template will include all entity attributes, except the primary key and system attributes. If **inherit** is selected, the data form layout will inherit the value from the entity which is parent for the current entity.
- **Hide Business Workflow** - Hides the entity record's state and state transition options in the end-user interface. For more information about business workflows, see ["Business Workflows" on page 418](#).
- **Read Only** - Prevents end-users from making any changes to the displayed form fields.
- **Disable Save Keyboard Shortcut** - Prevents end-users from saving and reloading the form by pressing the Ctrl+S keyboard shortcut.
- **Flow Title** - Selects the title displayed on each step (see ["Steps" on page 149](#)):
  - use display name (displays both the flow name and the step name)
  - show only step display name (displays only the step's name)

- **Style Sheets** - Select the style sheet you wish to use. Multiple style sheets can be added. The order in which you select them is also the order in which they are applied. For details, see ["Style Sheets" on page 1209](#).
- **Disable prompt for unsaved changes** - When navigating away from the form or step, for instance by clicking the Back button, if the form contains unsaved data, the user will not be prompted to save the changes.
- **Save automatically data on leave** - When navigating away from the form or step, for instance by clicking the Back button, any unsaved data in the flow will be automatically saved.

3. Click **Save and Reload**.

## Edit an Entity Form

To edit an entity form, expand the Data Forms section of an entity and double click the form. The form's settings are displayed.

### General

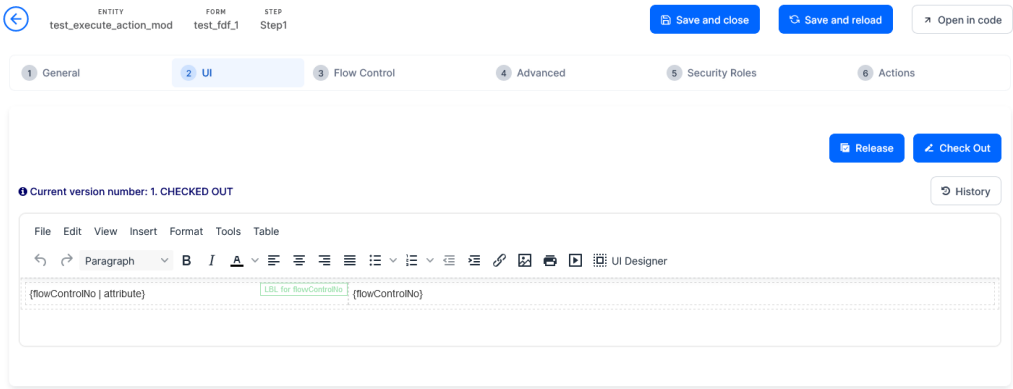
In the General section you can edit any of the general information fields filled in at form creation.

### UI

Use the UI designer to define the form's layout. For details on how to use the UI designer, see ["UI Designer" on page 323](#).

## Legacy Template Designer (deprecated)

For forms created in FintechOS Platform versions prior to v.24, the UI tab displays the legacy template designer which, in addition to the UI designer, provides an HTML editor that allows you to configure the form's HTML elements.



Legacy forms also allow you to use the ["Advanced Code Editor \(deprecated\)" on page 1142](#) or the ["Code Editor" on page 1108](#) respectively to edit their HTML code directly.

The form's UI template supports the following tokens:

Token	Description
{AttributeName}	Displays the corresponding field in the UI. <div><b>NOTE</b> The attribute name must be included between curly brackets; otherwise, a simple text will be displayed on the page instead of the actual field.</div>
{#RelationshipName, view: viewName#}	Generates a view provided by relationship and by view. The viewName is optional and specifies which view to generate. If viewName is not provided, the default view will be displayed on the data form.

Token	Description
{#RelationshipName, view: viewName, editmode:cell#}	<p>Generates a view provided by relationship and by view. This view allows inline editing, meaning that you can edit cells one by one directly in the grid, without opening specific records.</p> <div> <p><b>NOTE</b></p> <p>In order to activate inline editing for a specific cell, you must tick the <b>Allow Editing</b> checkbox displayed on the entity view column (View &gt; View columns).</p> </div>
{#MKT_CampaignResponse_MKT_Campaign,nodelete,noinsert#}	<p>Generates a view provided by relationship and by view, without the <b>Delete</b> and <b>Insert</b> buttons.</p> <p>You can apply the same logic (similar tokens) for the Export (noexport) or Refresh (norefresh) if you no longer need them on the view.</p>
{\${ChartName\$}}	Generates a chart based on the provided chart name.
{? entityName, view: viewName ?}	Allows you to display a view from another entity. For information on how to use it, see <a href="#">Display View from Another Entity</a> .

In the HTML template, you can link HTML elements (labels) to attributes. For information on how to do it, see ["Link Labels to Attributes \(deprecated\)" on page 270](#).

## Steps

Use the Steps section if you want to set up forms with multiple pages. For details on how to work with steps, see ["Add and Configure Steps" on page 231](#).

## Field Options

The Field Options section allows you to customize form fields based on how users have filled out other fields in the form: show field values, show or hide specific fields, etc. For details on how to work with field options, see ["Configure Field Options" on page 248](#).

### Filtered Fields

A filtered field restricts the values available in a form field based on entries in other fields. For details, see ["Defining Filtered Fields" on page 266](#).

### Advanced

The Advanced section allows you to write custom [client-side](#) JavaScript code that is executed before or after the form is rendered. For details on the execution sequence of the various form-level and step-level scripts, see ["Code Execution Sequence" on page 301](#).

### Security Roles

Use the Security Roles section to select the ["Security Roles" on page 1245](#) required for users that can access the form.

To learn more, check our Academy Evolutive Data Core course - [Entity Views module](#).

## Transient Data Entities


Transient data entities allow your ["Form Driven Flows" on page 221](#) to temporarily store and display data that has been loaded from or is going to be saved to an external data source. Thus, legacy systems benefit the most from the use of transient data and the data and metadata are not lost. The transient data entity's attributes are included in the flow's data model through data extensions (see [Extend the Data Model](#) for details) and their values are preserved only while the flow is open. Transient data entities use automation scripts for load/save that are triggered when such a flow is displayed or when it is saved to facilitate data transfers from/to external sources.

This section covers the following topics:

---

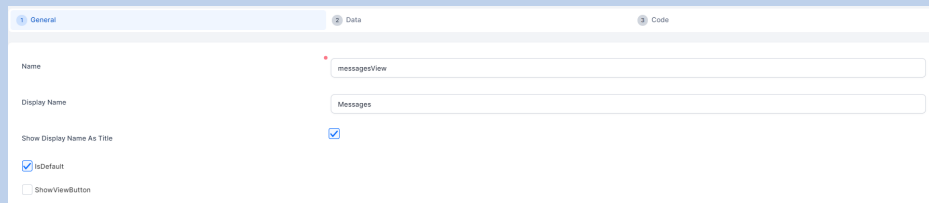
### Create transient data entities

1. Open FintechOS Studio.
2. From the **Main Menu** () , go to **Evolutive Data Core > Data Model Explorer** and click **Insert**.

3. In the **Entity Type** field, select Transient Data.
4. Enter a **Name** for the transient entity. This is a unique name used to identify the entity internally by the system.
5. Enter a **Display Name** for the transient entity. This is the name that will be displayed in the end-user interface.
6. Click **Save and reload** ()
7. In the **Data Model** section, add the entity's data attributes (for details, see ["Attributes" on page 71](#)).
8. If you want the transient data entity to appear in form driven flows as a grid of multiple records (not a single record instance), use the **Data View** section to define a view that includes the attributes you wish to display. For details see ["Data Views" on page 112](#).

## NOTE

Make sure that the view is configured as the default view of your transient data entity, by checking the **IsDefault** checkbox.



The screenshot shows the configuration interface for a transient data entity. The 'Data' tab is selected, and the following fields are visible:

- Name:** messagesView
- Display Name:** Messages
- Show Display Name As Title:** ☒
- IsDefault:** ☒
- ShowViewButton:** ☐

9. Click **Save and reload** ()

## Define the automation script for load

The automation script for load is triggered when a form driven flow with a data extension linked to the transient data entity is displayed. This is useful, for instance, if you use the transient data entity to fetch data from an external source and display it in the form driven flow.

To add an automation script for load:

1. Open the transient data entity in the editor.
2. Use the **Automation Script for Load** field to create the automation script.
  - The automation script must be on-demand (see ["Create On-demand Server Automation Scripts" on page 1168](#) for details).
  - The output structure must be set to your transient data entity (see ["Create On-demand Server Automation Scripts" on page 1168](#) for details). If you will use the transient data entity to show a grid of records (not a single record instance) when displayed in a form driven flow, make sure that the **Output Parameter Type** is set as Collection. Otherwise, choose Single Instance.

#### Examples

## List the user accounts from an external FintechOS Platform instance as a single record

In this automation script for load:

- We use the **ftosOpenAPI** Web API client library that stores the API specifications for another FintechOS Platform instance to make API calls to that instance. For more details about Web API client libraries, see ["Web API Client Libraries" on page 960](#).
- We read the incoming **webserver**, **username**, and **password** input parameters that we have set up in the automation script. For details, see ["Create On-demand Server Automation Scripts" on page 1168](#).
- We authenticate to the above web server with the provided credentials and we run a query on the **systemuser** entity.
- We extract each **userName** from the **Records** object in the result set in the **users** variable, which we then save to the **importedData** attribute we have set up for the transient data entity.

```

context.result = createResult();

let client = importWebApiClient('ftosOpenApi',
context.parameters.webserver);
try {
    let authToken = client.authorize.getToken("client_id",
context.parameters.username, context.parameters.password,
"", "");
    if (authToken && authToken.access_token) {
        let data = client.openApi.query({
            apiInfo: {
                token : authToken.access_token
            },
            request: {
                entity: {
                    name: "systemuser",
                    alias: "t"
                },
                distinct: false
            }
        });
        let users = [];
        data.Records.forEach(myFunction);
        function myFunction(value){
            users.push(value.t_userName)
        };
        context.result.importedData = toJson(users)
    }
    else
        throw new Error('Invalid authentication!');
}
catch(err) {
    context.result.importedData = err
}

```

List the visitors and messages saved in a guest book database on an external FintechOS Platform instance as a collection of records

In this automation script for load:

- We use the **ftosOpenAPI** Web API client library that stores the API specifications for the **<https://anotherFTOS/Pulsarb71i02Portal>** FintechOS Platform instance to make API calls to that instance. For more details about Web API client libraries, see "[Web API Client Libraries](#)" on page 960.
- We authenticate to the above web server with the credentials and we run a query on the **guestBook** entity.
- We store each **visitor** and **message** we find in the guest book in an **entry** object, and we push each entry object in the **result** collection.
- We set the result collection as the script's output, (we have configured the script to have a collection output parameter type).

```
let result = [];
let entry = {};

let client = importWebApiClient('ftosOpenApi',
'https://anotherFTOS/Pulsarb71i02Portal');
try {
    let authToken = client.authorize.getToken("client_id",
'username', 'password', "", "");
    if (authToken && authToken.access_token) {
        let data = client.openApi.query({
            apiInfo: {
                token : authToken.access_token
            },
            request: {
                entity: {
                    name: "guestBook",
                    alias: "t"
                },
                distinct: false
            }
        });
        data.Records.forEach(myFunction);
        function myFunction(value){
            entry = {};
            entry.visitor = value.t_visitor;
            entry.message = value.t_message;
            result.push(entry)
        }
    }
}
```

```

        };

        setResult(result)
    }
    else
        throw new Error('Invalid authentication!');
    }
    catch(err) {
        context.result.importedData = err
    }
}

```

### Define the automation script for save

The automation script for save is triggered when a form driven flow with a data extension linked to the transient data entity is saved. This is useful, for instance, if you use the transient data entity to save data displayed it in the form driven flow to an external destination.

To add an automation script for save:

1. Open the transient data entity in the editor.
2. Use the **Automation Script for Save** field to create the automation script.

## Example: Update an entry in the Swagger sample pet store

In this automation script for save:


- We use the **petstore** Web API client library that stores the API specifications for the <https://petstore.swagger.io/v2> web service. For more details about Web API client libraries, see ["Web API Client Libraries" on page 960](#).
- We call the **updatePet** endpoint to update the **id**, **name**, and **tags** attributes of an entry with the values stored in the respective **petID**, **petName**, and **petTags** input parameters that we have set up in the automation script. For details, see ["Create On-demand Server Automation Scripts" on page 1168](#).
- We leave the **photoUrls** attribute empty.

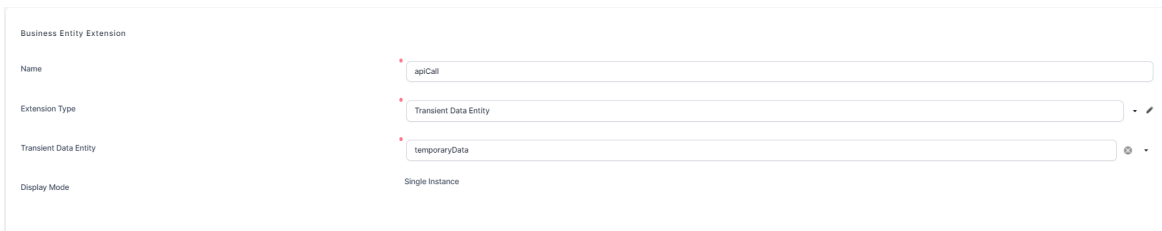
```
var client = importWebApiClient("petstore",
  "https://petstore.swagger.io/v2");
client.updatePet({
  id: context.parameters.petId,
  name: context.parameters.petName,
  tags: context.parameters.petTags,
  photoUrls: []
})
```

## Extend platform data entities with transient data entities

To include a transient data entity in a form driven flow, you must extend the flow's source entity data model with the transient data entity. This will allow you to interact with the transient data entity attributes via fields in your form driven flow. For more information about data model extensions, see [Extend the Data Model](#).

### Step 1. Create a transient data entity extension


1. Open a platform data entity in the editor.
2. Expand the **Extended Model** section and click **Insert**.
3. Enter a **Name** for your entity extension.
4. In the **Extension Type** field, select Transient Data Entity.
5. In the **Transient Data Entity** field, select a transient data entity you created earlier (see ["Create transient data entities" on page 150](#) for details).
6. Click **Save and reload** ()



The screenshot shows a form titled "Business Entity Extension" with the following fields and values:

Field	Value
Name	apiCall
Extension Type	Transient Data Entity
Transient Data Entity	temporaryData
Display Mode	Single Instance

**Step 2. Add virtual attributes (only for transient data entities with single instance outputs)**

1. Open the entity extension for your transient data entity.
2. Select the **Virtual Attributes** tab.
3. Click **Insert** and fill in the virtual attribute's settings.
  - **Related Attribute** - Select the transient entity attribute linked to your virtual attribute.
  - **Name** - Enter a name for your virtual attribute. This is a unique name used to identify the virtual attribute internally by the system.
  - **Display Name** - This is the field name that will be displayed in the end-user interface.
  - **Updatable** - Select to have the data extension value updated automatically.
  - **Attribute Type** - Will be updated automatically to match the Related Attribute.
  - **Length** - Will be updated automatically to match the Related Attribute.
  - **Required Level** - Select if the attribute is optional, required, or recommended to be filled.
  - **Tooltip** - If tooltips are set to be shown on forms and digital journeys and you want to have a tooltip explaining this data extension in the user interface, provide the desired text in the Tooltip text area field.
4. Click **Save and Reload** (.

**Add Virtual Attribute**

Related Attribute: Name

Name: \* name\_name

Display Name: \* extend Data Name

Updatable: ☒

Attribute Type: Text

Length: 20

Required Level: Select...

Use Related Display Name: ☐

Tooltip:

### Step 3. Bind entity attributes to the automation script for load input parameters

If the transient data entity's automation script for load includes input parameters (see ["Define the automation script for load" on page 151](#) for details), you must bind those parameters to attributes in your extended entity. When a form driven flow is displayed, the input parameters will be populated based on the matching entity attributes.

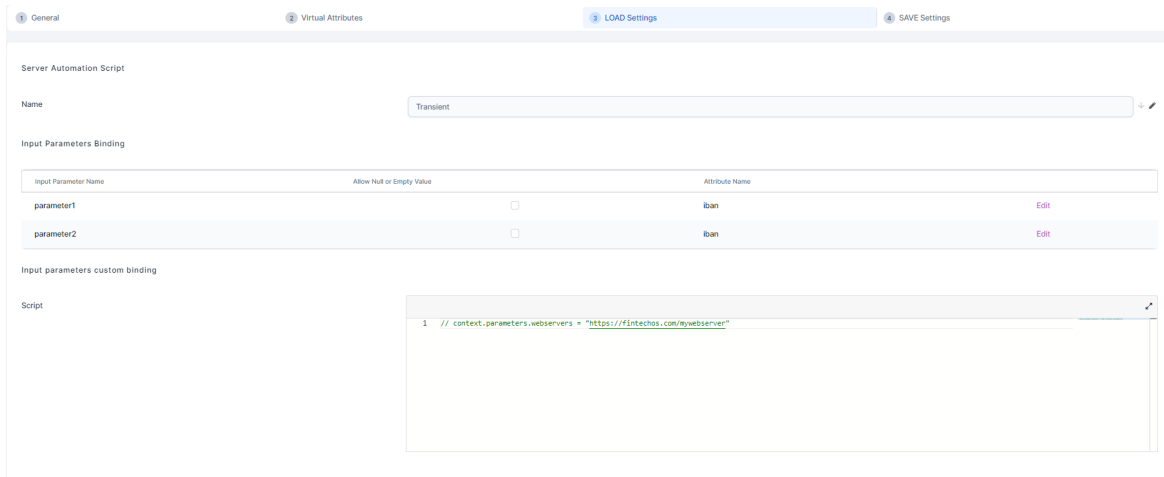
To bind input parameters to entity attributes:

1. Open the entity extension for your transient data entity.
2. Select the **LOAD Settings** tab.
3. The server automation script **Name** will be populated automatically with the transient data entity's automation script for load.
4. In the **Input Parameters Binding** section, match each input parameter to the entity attribute used to populate it. Click the Edit/Save links at the right end of each row to set the attribute names.
5. Optionally, you can manually enter a script in the **Input Parameters Custom Binding** to define custom input values for your input parameters.

**IMPORTANT!**

**Input Parameter Custom Binding** settings override settings in the **Input Parameters Binding** section

6. Click **Save and Reload** ()



1 General 2 Virtual Attributes 3 **LOAD Settings** 4 SAVE Settings

Server Automation Script

Name Transient

Input Parameters Binding

Input Parameter Name	Allow Null or Empty Value	Attribute Name	
parameter1	<input type="checkbox"/>	iban	Edit
parameter2	<input type="checkbox"/>	iban	Edit

Input parameters custom binding

Script

```
1 // context.parameters.webserver = "https://fintechos.com/mywebserver"
```

#### Step 4. Bind entity attributes to the automation script for save input parameters

If the transient data entity's automation script for save includes input parameters (see ["Define the automation script for save" on page 155](#) for details), you must bind those parameters to attributes in your extended entity. When a form driven flow is saved, the input parameters will be populated based on the matching entity attributes.

To bind input parameters to entity attributes:

1. Open the entity extension for your transient data entity.
2. Select the **SAVE Settings** tab.
3. The server automation script **Name** will be populated automatically with the transient data entity's automation script for save.
4. In the **Input Parameters Binding** section, match each input parameter to the entity attribute used to populate it. Click the Edit/Save links at the right end of each row to set the attribute names.

- Optionally, you can manually enter a script in the **Input Parameters Custom Binding** to define custom input values for your input parameters.

The screenshot shows the 'SAVE Settings' tab in the Fintechos Studio configuration interface. It includes sections for 'Server Automation Script', 'Input Parameters Binding', and 'Input parameters custom binding'. The 'Input Parameters Binding' section contains a table with the following data:

Input Parameter Name	Allow Null or Empty Value	Attribute Name
input_bool	<input checked="" type="checkbox"/>	TestAttribute2

### IMPORTANT!

**Input Parameter Custom Binding** settings override settings in the **Input Parameters Binding** section.

- Click **Save and Reload** ()

## Display transient data entity attributes in form driven flows

Once you extend a platform entity data model with a transient data entity (see ["Extend platform data entities with transient data entities" on page 156](#)), you can display the transient data entity's attributes in the extended entity's form driven flows. Transient data entities with single instance output structures allow you to display individual fields for each attribute, while those with collection output structures can be displayed in grids based on the transient data entity's default view (see ["Create transient data entities" on page 150](#) for details).

### Display transient data entity attributes for single instance outputs

- Create a form driven flow based on a platform entity that was extended with a transient data entity with single instance output structure.

- For information on how to create form driven flows, see ["Form Driven Flows"](#) on [page 221](#).
- For information on how to extend a platform data entity with a transient data entity, see ["Extend platform data entities with transient data entities"](#) on [page 156](#).
- For information on how to create automation scripts with single instance output structures, see ["Create On-demand Server Automation Scripts"](#) on [page 1168](#).

2. Make sure to include the transient data entity extension in the form driven flow's data model.

3. In the form driven flow's UI editor, you will be able to insert data templates with the transient data entity's virtual attributes.

4. The corresponding transient data entity's attributes will be displayed in the user interface.

## Display transient data entity attributes for collection outputs

1. Create a form driven flow based on a platform entity that was extended with a transient data entity with collection output structure.
  - For information on how to create form driven flows, see ["Form Driven Flows" on page 221](#).
  - For information on how to extend a platform data entity with a transient data entity, see ["Extend platform data entities with transient data entities" on page 156](#).
  - For information on how to create automation scripts with collection output structures, see ["Create On-demand Server Automation Scripts" on page 1168](#).
2. Make sure to include the transient data entity extension in the form driven flow's data model.

3. In the form driven flow's UI editor, insert a relation container with the value based on the entity extension's name, using the following syntax:

```
{^entity extension name^}
```

4. The corresponding transient data entity's attributes will be displayed in the user interface in a grid based on the transient entity's default view.

**EDIT GUEST BOOK**

Name

me

Refresh

Visitor	Message
<div>Q</div>	<div>Q</div>
John Doe	I liked it
Jane Doe	I didn't like it.

## Sample API Calls

The Data API section allows you to download a [Postman API Client](#) collection with sample API calls pre-populated with the entity's attributes. For detailed information about the API calls supported in FintechOS Platform, see the [API Reference Guide](#).

To download the sample API calls:

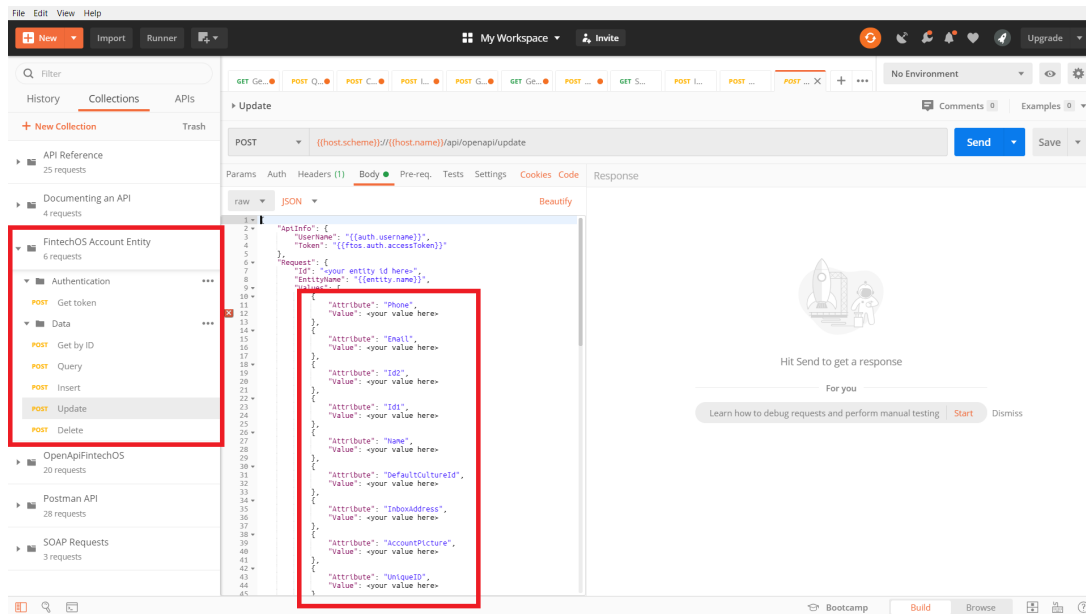
1. Open the entity in the Data Model Explorer.
2. Expand the **Data API** section and click the **Postman REST API Methods** button.

The screenshot shows the 'Edit Business Entity' interface. It features a form with the following fields and sections:

- Details**
  - Entity Type: Platform Data
  - Name (only use for add entity): Account
  - DisplayName: Customer
  - DisplayCollectionName: Customers
  - Description: (empty text area)
  - PrimaryAttributeName (only use for add entity): Name
  - PrimaryAttributeDisplayName (only use for add entity): Name
  - Default Entity Status: Draft
  - Is Audited: ☐
  - Business Workflow: Account Workflow
  - Optimization Search Data (Filter starts with): ☐
- Data Model** (expandable)
- Data Forms** (expandable)
- Data Views** (expandable)
- Data API** (expanded)
  - Postman REST API Methods

3. A JSON file containing the Postman collection will be downloaded on your local machine.
4. In Postman, click **File > Import** and select the downloaded JSON file.
5. Once the import finishes, the API calls collection will be visible in Postman.

6. The collection contains basic API calls for authentication and CRUD operations with the entity's attributes names pre-populated.



## Data Import Templates

To import data (entity records) along with its metadata from another source system follow the instructions below:

### IMPORTANT!

Before importing, go to the source system and export the data in an Excel file.

Do not include the source primary keys in the data exports. Primary keys are generated automatically by the system.

Make sure to include in the source export any attributes that are required in the destination entity, otherwise the import will fail.

1. In FintechOS Studio, go to **Main Menu > Evolutive Data Core > Data Import Templates > Insert**.

2. Fill in the **Name** for the package, the destination **Entity**, and any applicable **Unique Constraint** (see ["Entity Unique Constraints" on page 94](#) for details).
3. Click **Save and Reload**. Two grids will show up: List of Data Import Attributes and List of Data Imports.
4. In the list of data import attributes, click **Insert** to specify the entity attributes you wish to import. For each attribute, fill in the following fields:
  - Name of the **Attribute** you wish to include. Be sure to include all attributes that are part of the unique constraint if applicable.
  - The corresponding **Column Name**.

**IMPORTANT!** The name must be the same as the name of the column in the Excel file.

- **DateTime format and Data Import Attribute Type.**

Add Data Import Attribute

DATA IMPORT ATTRIBUTE

Attribute

Name

Column Name

attribute1

DateTime Format

MM-dd-yyyy

Data Import Attribute Type

Plain Field

5. Click **Save and close**.

6. In the list data imports, click **Insert**.

Edit Data Import

Data Import

Status: New

File: Add file or Drop file here

Data Import: customerimport

Rollback when error: ☒

Data Import Logs

+ Insert X Delete Export Refresh

Message Created On

No data

7. Fill in the following:

- **Status** - This is auto-filled. The options are:
  - **New** (this is the initial status of an import before the "Start import" button has been clicked)
  - **Imported** (this is the confirmation if the import was successful)
  - **Error** (this is the status if the import failed)
- **File** - Add the Excel file here.
- **Data Import** - Name of the template.
- **Rollback when error** - Enables you to revert to the initial state of the data base if the import fails.
- **Allow Partial Import** - Skips the Excel rows that generate errors without stopping the entire import process. Skipped records and the reasons for failure are recorded in the data import log.

8. Click **Start Import**. The process is recorded in the Data Imports Log. If there is an error, the detailed error messages are displayed in the grid.

9. Click **Save and close**. Repeat as many times as needed.

Edit Data Import Template

Data Import Template

Name

Entity

Unique Constraint

List of Data Import Attributes

Attribute	Column Name	Data Import Attribute Type
<input type="text" value="Email"/>	<input type="text" value="Email"/>	<input type="text" value="Plain Field"/>
<input type="text" value="Name"/>	<input type="text" value="Name"/>	<input type="text" value="Plain Field"/>

List of Data Imports

User	Status	Created On
<input type="text" value="New"/>	<input type="text" value="New"/>	<input type="text" value="New"/>

## Data Governance

Data governance allows you to classify sensitive data and anonymize it on request. This is particularly useful to ensure compliance with data protection regulations, such as the GDPR art. 17 Right to erasure (Right to be forgotten).

### IMPORTANT!

When anonymizing data, the relevant record attributes are permanently deleted and replaced with generic information, such as *Sensitive data deleted*.

The data anonymization process consists of multiple steps, involving both back-end and front-end operations, as described below:

1. **Sensitive Data Classification** - In this stage, digital developers or consultants who are familiar with the data model define the "[Sensitive Context And Sensitive Data](#)" on the [next page](#) and the [Sensitive Data](#) that is subject to anonymization in FintechOS Studio.
2. **Issuing Anonymization Requests** - Once the sensitive data has been classified, operators can use FintechOS Portal to make [Data Anonymization Requests](#). The operators search for the relevant data to be anonymized based on generic criteria,

such as the sensitive context and the type of sensitive data (name, telephone number, identification code, etc.), not based on specific entities and attributes from the data model. Then, they identify, validate, and approve the data to be anonymized.

3. **Data Anonymization** - After the anonymization requests are approved, "[Data Anonymization Jobs And Requests](#)" on page 176 can process the requests and anonymize the actual data. For this purpose, you must configure the job server to run an anonymization service for the corresponding sensitive context.

This section covers the following topics:

---

## Sensitive Context And Sensitive Data

The sensitive context groups components that are involved in data anonymization for a specific purpose, such as ensuring GDPR compliance. Specifically, for such a purpose, the sensitive context will help:

- Define the [Sensitive Data](#).
- Provide a reference for the "[Data Anonymization Jobs And Requests](#)" on page 176 to find the relevant data to be anonymized.

### Create a Sensitive Context

1. In FintechOS Studio, go to **Main Menu > Evolutive Data Core > Data Governance > Sensitive Data Settings**.
2. Click the **Insert** button at the top right corner of the screen.
3. In the Add Sensitive Context page, fill in the following fields:
  - **Code** - The sensitive context identifier.
  - **Name** - The sensitive context name.

- **Description** - Optional description for the data anonymization purpose of the sensitive context.

The screenshot shows a web form titled "Edit Sensitive Context". Inside the form, under the heading "Sensitive Context", there are three input fields. The "Code" field contains the text "GDPR" and has a red asterisk to its left. The "Name" field also contains "GDPR" and has a red asterisk to its left. The "Description" field contains the text "Ensure compliance with GDPR art.17 Right to erasure (Right to be forgotten)".

4. Click **Save and Close**.

## Sensitive Data

Sensitive data definitions specify the data anonymization settings applicable for a sensitive context, such as:

- The attributes used by operators to search for the records that contain sensitive data.
- The attribute values that must be anonymized once those records are identified.
- The validations required to allow the data anonymization to proceed.

Each sensitive data definition is built around a sensitive entity (and, optionally, its related entities) on which the anonymization is applied. The sensitive data definition also specifies the validation rules that must be fulfilled in order to allow the anonymization.

### 1 Configure the Sensitive Entity

1. In FintechOS Studio, go to **Main Menu > Evolutive Data Core > Data Governance > Sensitive Data Definitions**.
2. Click **Insert**.

3. In the **Sensitive Entity Configuration** tab, fill in the following information:
  - **Code** - The identifier for the sensitive data definition.
  - **Sensitive Context** - The applicable "[Sensitive Context And Sensitive Data](#)" on page 169.
  - **Entity** - The sensitive entity on which the anonymization is based.
  - **Description** - Optional detailed description of the sensitive data definition.
4. Click **Save and Reload**.
5. In the Sensitive Attributes list, click the **Insert** button to add the sensitive entity's attributes that are relevant for the anonymization (either as search criteria or as data to be anonymized). For each attribute, specify the following values:
  - **Sensitive Entity** - This is automatically populated with the sensitive entity name.
  - **Attribute** - The attribute's name.
  - **Sensitive Type** - Attribute classification used by operators as search criterion when performing [Data Anonymization Requests](#). The following values are available: CNP, Name, Phone, First Name, Last Name.

**NOTE**

Only attributes with sensitive types can be searched in data anonymization requests.

- **To Be Anonymized** - Check to indicate that, if a sensitive entity record is subject to data anonymization, this attribute will be anonymized in accordance with its data type:

Attribute Data Type	Anonymized Attribute Value
Text	Sensitive data deleted.
File	[]
Text Area	Sensitive data deleted.
Numeric	0
Whole Number	0
Date Time	01.01.1900
Date	01.01.1900
Bool	NULL
Option Set	NULL
Lookup	NULL

6. After you finish adding all the sensitive attributes, click **Save and Reload**.

The screenshot shows the 'Sensitive Entity Configuration' window. Under 'Sensitive Entity', the 'Code' field is 'GDPR\_customers', 'Sensitive Context' is 'GDPR', and 'Entity' is 'TestEntity\_AT'. The 'Sensitive Attributes' section displays a table with the following data:

Attribute	Sensitive Type	To Be Anonymized
TestAttribute	CMP	AB

Buttons for '+ Insert', 'X Delete', 'Export', and 'Refresh' are located at the top right of the attributes table.

## 2 Define the Related Sensitive Entities (optional)

The related sensitive entities allow you to propagate data anonymization to a chain of entities that are related to the main sensitive entity. For instance, when anonymizing a customer's data, you may want to also anonymize the

data for all the bank accounts that the customer holds (which is stored in a related entity).

You can also include entities that are related to other related entities, extending the data anonymization scope to a tree of entities originating from the main sensitive entity.

To define a related sensitive entity:

1. In the sensitive data definition editor, select the **Related Sensitive Entities** tab.
2. In the Related Entities list, click **Insert** to add the sensitive entities that are part of the data anonymization scope. For each related entity, specify the following values:
  - **Code** - The identifier for the related entity.
  - **Master Entity** - Automatically populated with the sensitive entity identifier.
  - **Relation** - Select from the list of relationships of the sensitive entity.
  - **Entity** - Related entity name. Automatically populated based on the selected relation.
3. Click **Save and Reload**.
4. In the Sensitive Attributes list, click **Insert** to add any entity attributes that are part of the data anonymization scope. For each attribute, specify the following values:
  - **Sensitive Entity** - This is automatically populated with the related sensitive entity name.
  - **Attribute** - The attribute's name.

- **To Be Anonymized** - Check to indicate that, if a sensitive entity record is subject to data anonymization, this attribute will be anonymized in accordance with its data type.

The 'Add Sensitive Attribute' dialog box contains the following fields:

- Sensitive Attribute:** (Empty text field)
- Sensitive Entity:** Dropdown menu with 'GDPR\_customers' selected.
- Attribute:** Dropdown menu with 'iban' selected.
- To Be Anonymized:** Checked checkbox.

5. Click **Save and Close**.
6. Repeat from Step 4 to add any remaining sensitive attributes.

The 'Sensitive Entity Configuration' tab displays the following information:

- Sensitive Entity:**
  - Code: GDPR\_customers
  - Entity: TestEntity\_AT
  - Description: (Empty text area)
- Sensitive Context:** test
- Sensitive Attributes Table:**

Attribute	Sensitive Type	To Be Anonymized
iban	CNP	<input checked="" type="checkbox"/>
TestAttribute	CNP	<input type="checkbox"/>

7. You can further define downstream related entities by selecting the **Related Sensitive Entities** tab and repeating the process.
8. Click **Save and Close**.

### 3 Add Validation Rules

Validation rules can prevent data anonymization if certain criteria are not met. For instance, you may want to allow data anonymization of a bank account data only if the account balance is 0.

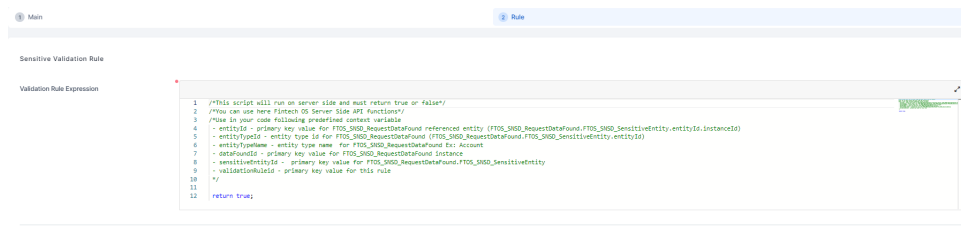
To add a validation rule for a sensitive data definition:

1. In the sensitive data definition editor, select the **Validation Rules** tab.
2. In the Sensitive Validation Rules list, click the **Insert** button to add a rule.
3. In the **Main** tab, specify the following values:
  - **Sensitive Entity** - The identifier for the sensitive data definition.
  - **Code** - The identifier of the validation rule.
  - **Name** - The name of the validation rule.
  - **Description** - Optional detailed description of the validation rule.
  - **Success Message** - Message displayed in the operator's interface when validating the result of a [Data Anonymization Requests](#) if the validation is successful.
  - **Failure Message** - Message displayed in the operator's interface when validating the result of a [Data Anonymization Requests](#) if the validation fails.

The screenshot shows the 'Main' tab of the 'Sensitive Validation Rule' editor. The form is structured as follows:

Sensitive Validation Rule	
Sensitive Entity	GDPR_customers
Code	accVal
Name	Accounts Validation
Description	Validate customer accounts.
Success Message	Validation successful.
Failure Message	Validation failed.

4. In the **Rule** tab specify the expression of the validation rule. The expression must return **true** in case of a successful validation or **false** if the validation fails.



5. Click **Save and Close**.

## Data Anonymization Jobs And Requests

Approved records are anonymized using an SQL procedure. For each "[Sensitive Context And Sensitive Data](#)" on [page 169](#), you must configure the FintechOS Platform job server to run a corresponding anonymization service.

### 1 Set up the Anonymization Service

Open the FintechOS Platform Job Server's **services.config** file in a text editor, and add a service based on the following model:

```

<service>
  <name>FTOS.Anonimization</name>
  <type>mixed</type>
  <steps>
    <step>
      <name>Run Sensitive_Context_Name anonymization</name>
      <type>sql</type>
      <command>exec ebs.uspFTOS_SNSD_DeleteSensitiveData
'Sensitive_Context_Name'</command>
      <sqlConnection>Data Source=SqlServerName ;Initial
Catalog=DataBaseName;User
ID=username;Password=password</sqlConnection>
      <mandatory>true</mandatory>
    </step>
  </steps>
</service>

```

## 2 Set up the Anonymization Service Trigger

Open the FintechOS Platform Job Server's **schedule.config** file in a text editor, and add a trigger for the anonymization service based on the following model:

```
<trigger>
  <name>FTOS.Anonimization</name>
  <!--<calendar>holyday1</calendar>-->
  <startTime>02.01.2019 11:00</startTime>
  <endTime>03.11.2080 11:02</endTime>
  <poolTime>60</poolTime>
  <repeatCount>-1</repeatCount>
  <rescheduleAfterRun>false</rescheduleAfterRun>
  <async>false</async>
  <expression>0/10 * * * * ?</expression>
  <services>
    <service>FTOS.Anonimization</service>
  </services>
</trigger>
```

The following settings apply:

- **<poolTime>** is expressed in seconds.
- **<repeatCount>** when set to -1 will run indefinitely. When set to a positive value (eg. 5) will run for only 5 times at each pool time.
- **<expression>** is a Cron expression. When present, the Cron expression will override the **<poolTime>**.

## 3 Restart the FintechOS Platform Job Server

After editing the **services.config** and **schedule.config** files, you need to restart the FintechOS Platform Job Server for your changes to take effect:

1. On the FintechOS Platform Job Server environment, open the **Command Prompt** application in **Administrator mode**.

2. Enter the following command and wait for the job server service to stop:

```
net stop FTOS.JobServer.<instance name>
```

3. Enter the following command and wait for the job server to restart:

```
net start FTOS.JobServer.<instance name>
```

## Data Anonymization Requests

Data anonymization requests allow FintechOS Portal operators to mark specific records within a ["Sensitive Context And Sensitive Data" on page 169](#) for anonymization. Once the relevant data is identified, validated, and approved for anonymization by the operator, a data anonymization job can process the marked records.

### 1 Create a Sensitive Request

1. In FintechOS Portal, go to **Main Menu > Data Anonymization Requests**.
2. Click the **Insert** button at the top right corner of the screen.
3. Fill in the following sensitive request values:
  - **Request no** - Identifier for the data anonymization request.
  - **Request Date** - Select a reference date for the data anonymization request.
  - **Sensitive Context** - Select the ["Sensitive Context And Sensitive Data" on page 169](#) relevant for the request.
4. Click the **Save and Reload** button at the top right corner of the screen.
5. In the Request Search Attributes grid, you will see the attributes that are available for search according to the sensitive context's [Sensitive Data](#) definition. Edit the search attribute you wish to use as search criteria:

- **Sensitive Request** - This field is populated automatically with the data anonymization request identifier.
- **Search Value** - The key term you wish to use for your search, such as a user's first name or SSN.
- **Sensitive Attribute Type** - The classification for the sensitive attribute type. The following values are available: CNP, Name, Phone, First Name, Last Name.

EDIT REQUEST SEARCH ATTRIBUTE

REQUEST SEARCH ATTRIBUTE

Sensitive Request

005-02/07/2021

Search Value

Doe

Sensitive Attribute Type

Name

## HINT

Although you can define any sensitive attribute type, the search will only work for the values that match the [Sensitive Data](#) definition. This is why the best approach is to edit the desired search value directly in the Request Search Attributes grid.

REQUEST SEARCH ATTRIBUTES

+ Insert

✕ Delete

📄 Export

🔄 Refresh

	Attribute Type	Search Value
	🔍	🔍
	CNP	
<input checked="" type="checkbox"/>	Name	Doe

- Click **Save and Close** at the top right corner of the screen.

6. The data anonymization request is saved with the **Register** status.

**1 Request** **2 Data Found**

**SENSITIVE REQUEST**

Request No: 001-05/07/2021

Request Date: 05/07/2021

Sensitive Context: GDPR

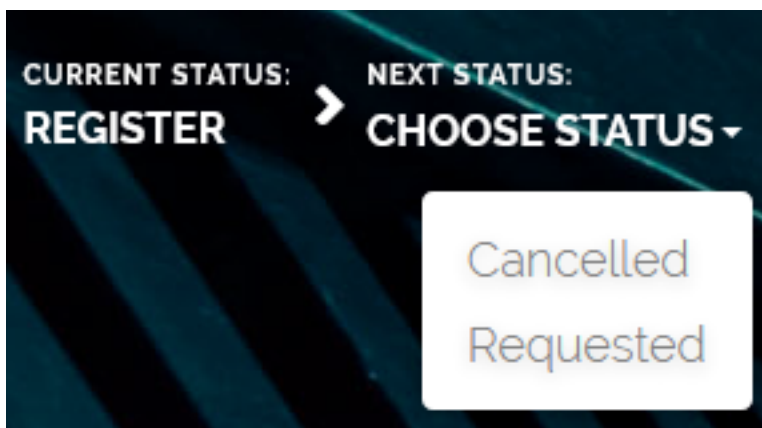
**REQUEST SEARCH ATTRIBUTES**

+ Insert X Delete Export Refresh

Attribute Type	Search Value
CNP	Doe

## 2 Check the Returned Sensitive Data

1. Once all the settings of the Data Anonymization Request are configured, you can choose to either cancel the request or proceed with the search by selecting the desired status from the top left corner of the screen.



2. Select the **Requested** status to lock the search criteria and perform the search. Once the status is updated, the search results are populated in the Data Found tab of the request. Each search result will also be in the Requested business status.

1 Request

2 Data Found

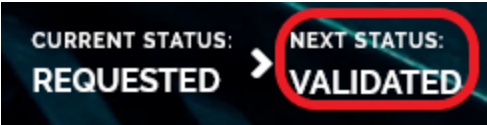
REQUEST ENTITIES

Export

Refresh

<input type="checkbox"/> Sensitive Entity	Identification Name	Business Status
GDPR_customers	77798237498713	Requested
GDPR_customers	123456765432	Requested

3. You can double click on the desired entities to do a preliminary check on the individual results. When ready, select the Validated status at the top right corner of the screen to run the data validations that were configured in the sensitive data (for details, see [3 Add Validation Rules](#)).



4. The request will advance to the Solved status and each search result will go into the Pending Approval status.

← CURRENT STATUS: SOLVED

1 Request

2 Data Found

REQUEST ENTITIES

Export

Refresh

<input type="checkbox"/> Sensitive Entity	Identification Name	Business Status
GDPR_customers	77798237498713	Pending Approval
GDPR_customers	123456765432	Pending Approval

### 3 Approve the Records for Data Anonymization

1. Open each search result in the grid and check it:

EDIT REQUEST ENTITY

REQUEST ENTITY

Sensitive Entity

GDPR\_customers

Identification Name

123456765432

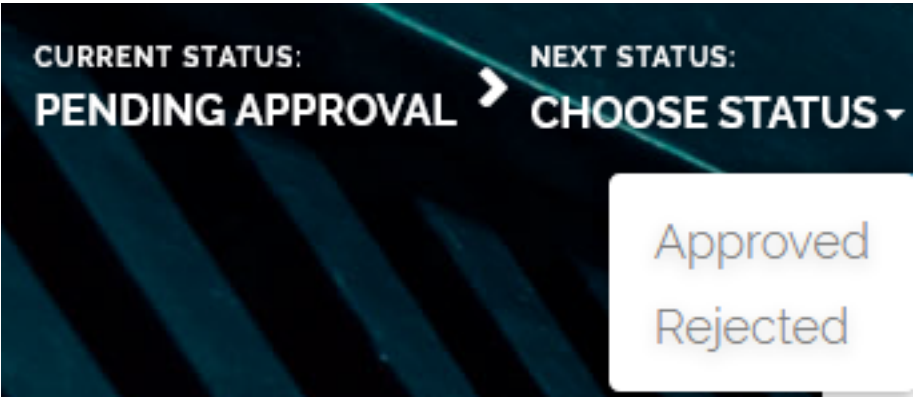
View

Comment

REQUEST DATA RULES

<input type="checkbox"/>	Name	Is Rule Valid	Message
<input type="checkbox"/>	(All)		
<input type="checkbox"/>	Accounts Validation	<input checked="" type="checkbox"/>	Validation successful.

- **Sensitive Entity** - Sensitive entity based on which the search was performed. For details, see [1 Configure the Sensitive Entity](#).
  - **Identification Name** - Primary attribute of the returned sensitive entity record.
  - **View** - Click this button to view the sensitive entity record in its default edit form.
  - **Comment** - Sensitive entity description (if provided).
  - **Request Data Rules** - The results of each validation rule applied to the returned data.
2. Based on the search result's data, from the top left corner of the screen, choose to either approve or reject data anonymization for the record.



**IMPORTANT!**

When you approve a record, the anonymized record attributes are permanently deleted and replaced with generic information, such as *Sensitive data deleted*.

3. Repeat for the remaining search results and either approve or reject them.

1 Request

2 Data Found

REQUEST ENTITIES

Export

Refresh

<input type="checkbox"/>	Sensitive Entity	Identification Name	Business Status
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	GDPR_customers	77798237498713	Approved
	GDPR_customers	123456765432	Rejected

When the FintechOS Platform Job Server runs the corresponding data anonymization jobs, the approved sensitive data is anonymized.

EDIT CUSTOMERS

CUSTOMERS

First Name

Last Name

SSN

ACCOUNTS

+ Insert

✕ Delete

📄 Export

🔄 Refresh

<input type="checkbox"/>	IBAN	Amount	Currency
	54168748521478	456.25	EUR
	74854145126963	232.00	RON
	8459523214874212	680.38	BWP

EDIT CUSTOMERS

CUSTOMERS

First Name

Last Name

SSN

ACCOUNTS

+ Insert

✕ Delete

📄 Export

🔄 Refresh

<input type="checkbox"/>	IBAN	Amount	Currency
	SENSITIVE DATA DELETED	0.00	SEN
	SENSITIVE DATA DELETED	0.00	SEN
	SENSITIVE DATA DELETED	0.00	SEN

The business status of the sensitive entity is set to Anonymized.

1 Request

2 Data Found

REQUEST ENTITIES

Export

Refresh

<input type="checkbox"/>	Sensitive Entity	Identification Name	Business Status
	Q	Q	Q
	GDPR_customers	77798237498713	Anonymized
	GDPR_customers	123456765432	Rejected

# Digital Experience

The Digital Experience encompasses a set of highly customizable components and features designed to build digital solutions for your business, and enrich your audience interaction and experience with your solutions. These components include customer journeys and digital frontends.

## Customer Journeys

Customer journeys are typically built around a progression of steps that an user goes through to achieve a goal. The way the actor gets from one place to another in a steady stream, i.e. process steps is defined in a form driven flow. More advanced user experiences can be built by combining multiple flows in a digital journey.

## Digital Journeys

FintechOS Digital Journeys are based on the Evolutive Data Core, which organizes, manages, and displays the various data collected in a business process. In FintechOS Studio, you can define how users interact with applications based on their needs and expectations to improve the customer journey and create a positive customer experience.

## Form Driven Flows

Form driven flows provide you with the mechanisms to make seamless user experience across the entire digital journey. An ordered collection of components which address an entire need of a digital actor. It is a part of the FintechOS Digital Journey anatomy.

If properly created, they provide you with opportunities to address both your team's and customers' pain points. For more info, see ["Form Driven Flows" on page 221](#).

## Custom Flows

A custom flow is an ordered collection of components which address a singular need in the direction of componentization. It represents a business flow that can be the base for a digital journey, but it is not associated with an entity.

## Differences between the Form Driven Flow, Custom Flow and Digital Journey

Feature	Form Driven Flow	Custom Flow	Digital Journey
Associated to an entity	Yes	No	Yes
Has steps	Yes	No	Yes
Before and after actions	Yes	No	By accessing the Form Driven Flow
Uses source code editor	Yes	Yes	By accessing the Form Driven Flow
Uses buttons with endpoint	Yes	Yes	By accessing the Form Driven Flow
Flow map	Yes	No	By accessing the Form Driven Flow
Digital Journey Map	No	No	Yes
Accesses UI Designer	Yes	Yes	By accessing the Form Driven Flow
Invoked Ad-hoc	No	Yes	No
Uses mock-up steps	Yes	No	No

## UI Designer

The built-in UI Designer enables FintechOS engineers to easily add HTML structures containing attributes, relations, or predefined UI elements without writing code, all from the user interface. For more info, see ["UI Designer" on page 323](#).

## Built-in Analytics

The digital journey analytics are a set of built-in indicators that offer insight on the performance and statistics of your digital journey. The data indicators calculate the success ratio of your digital journey, the number of Not Completed vs. Completed digital journeys, completion time, abandonment rate, average time spent on a specific step, etc. For more info, see ["Built-in Analytics" on page 212](#).

## Digital Frontends

FintechOS Studio enables you to define every interaction that your business has with your internal team as well as with the customers. Broadly defined, digital frontends represent your user experience.

The Digital Experience comprises the following sections:

---

## Customer Journeys

Customer journeys are the different types of processes a user goes through to achieve a goal (e.g.: onboard, apply for a loan, open a bank account, etc.). They vary in complexity from custom flows (HTML forms built from scratch, with no underlying data model), to form driven flows (a succession of steps that run on top of a data model based on a business entity record), to digital journeys (complex user experiences in which you can combine multiple form driven flows and custom flows).

---

## Digital Journeys

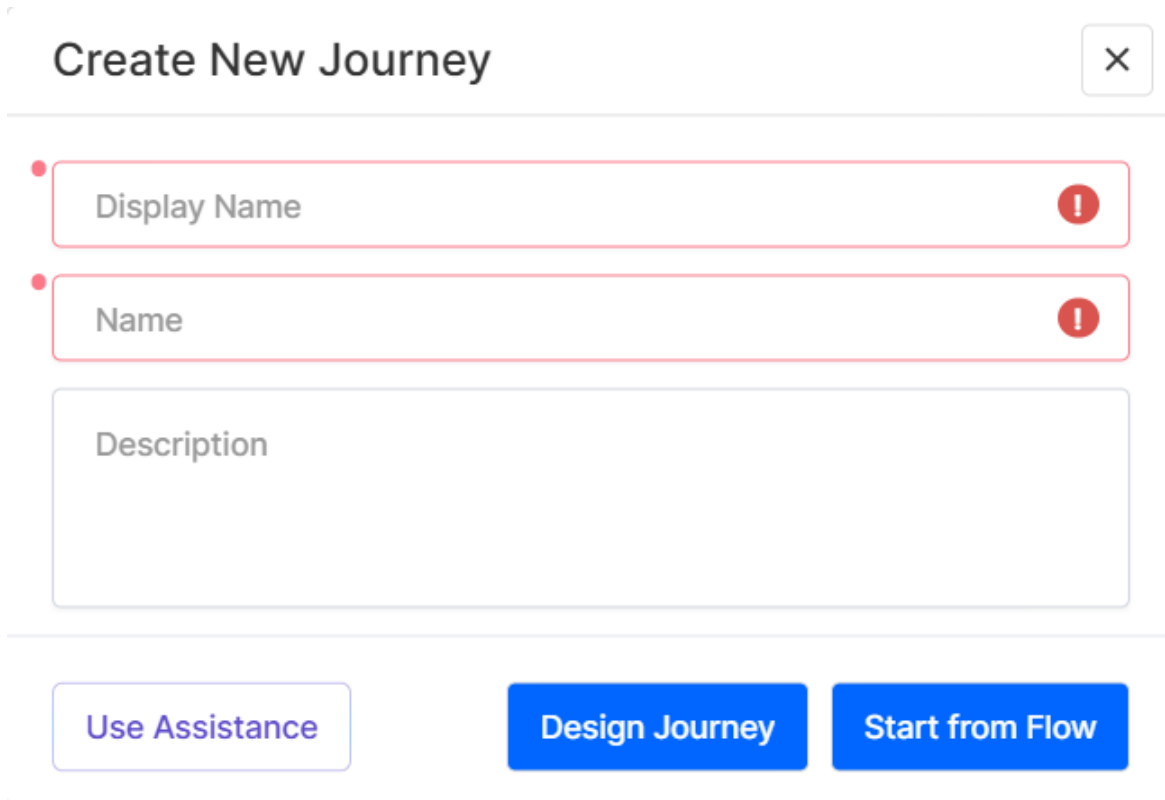
Digital journeys provide a visual representation of the user's path to completing a specific process, such as onboarding or applying for a loan. Digital journeys guide users through a series of steps and decision points, incorporating all necessary customization elements (such as data models, form-driven flows, user interfaces, and automation processors) to ensure the successful completion of the process.

Digital journeys enable you to combine multiple flows into a cohesive experience. You can combine ["Form Driven Mock-up Flows" on page 280](#), ["Form Driven Flows" on page 221](#), and ["Custom Flows" on page 283](#) within the same journey. For example, you might create a co-browsing onboarding journey by first developing a mock-up flow for the user, converting it into a form-driven flow, and then adding a secondary flow for the operator. This setup allows the client to input their information while an operator assists and verifies additional data via video call.

In addition to the graphical user interface, digital journeys also support API configurations, making it possible to load and access them in custom applications, web applications, front-end interfaces, and more. This capability is enabled through the "Exposed API Journey" option. For more information, see the [Digital Journey API documentation](#).

### Create a Digital Journey

To create a digital journey, in FintechOS Studio, go to **Main Menu > Digital Experience > Digital Journeys** and click **Insert**.



**Create New Journey** [X]

Display Name [!]

Name [!]

Description

Use Assistance   Design Journey   Start from Flow

A pop-up window appears, offering three methods to create a digital journey:

- **Use Assistance** - Use ["Dex" on page 45](#) to pre-populate the journey settings and flow based on an existing journey that matches your requirements:
  1. Interact with Dex to provide details about the journey you want to create.
  2. After collecting enough information, Dex generates a list of relevant journey templates. Click the **Open list** button to view the suggestions.
  3. Hover over a proposed template and click either **Use** to pre-populate the journey flow and data configuration based on the template or **Preview** to examine the journey template further before making a decision.
- **Design Journey** - Draft a high-level overview of the digital journey, enabling preview, testing, and stakeholder approval before development begins. This method supports a single underlying mock-up form, which you can later customize for the final product:

1. Use the ["Journey Designer" below](#) to draft the entire user experience.
  2. Once you are happy with the outcome, publish the journey to automatically generate an associated mock-up flow, complete with steps, navigation rules, and actions.
  3. Begin development by attaching a data model to the mock-up form (converting it into a regular form-driven flow) and implementing further customizations.
- **Start from Flow** - Design the journey on top of existing or newly created form driven flows. This method allows you to leverage existing form driven flows and to build complex journeys that integrate multiple flows.

## Journey Designer

If you chose to ["Create a Digital Journey" on page 189](#) using the **Use Assistance** or **Design Journey** option, you can use the Journey Designer to draft a high-level overview of the digital journey. This tool allows you to preview, test, and obtain stakeholder approval before development begins. Once you are happy with the design, you can publish the journey to automatically generate an associated mock-up flow, with the matching structure (sequence of steps, navigation rules, and actions), to kickstart the development process.

Some of the advantages of using the Journey Designer over manually building a mock-up form-driven flow include:

- **Dex AI Assistance:** An integrated chatbot can examine your requirements and help you select a relevant template for the initial draft of your journey.
- **Intuitive Graphical Interface:** Focused on a unified design experience, free from unnecessary distractions.
- **Contextual Linking:** Link products, actors, and channels to the journey to enhance context and receive relevant options during the design process.

- **Simplified Design Components:** Easily define all journey touchpoints, including user interface screens, navigation rules, integrations, back-office tasks, underwriting/eligibility checks, document generation, and omnichannel messaging.
- **Journey Preview:** Visualize the user experience and obtain stakeholder approval before development.
- **Automatic Documentation:** Annotate each design element with detailed specifications such as descriptions, validations, UX definitions, and error handling, then automatically generate a fully-formatted Functional Specifications Document (FSD) aligned with the "Product is the Spec" philosophy.

## Journey Settings

To configure journey settings, click the ellipsis button (...) in the top right corner of the Journey Designer and select **Edit Journey Settings**.

**docMortgage**

**Products**  
This journey will cover one or more products

Select products

**Actor**  
Who will be using this journey

Customer

Agent

Underwriter

Select or type to add an actor

**Type**  
What scope will this journey cover

Loan Origination

**Channel**  
The channels in which this journey will be available

Self service

Use Assistance

Save

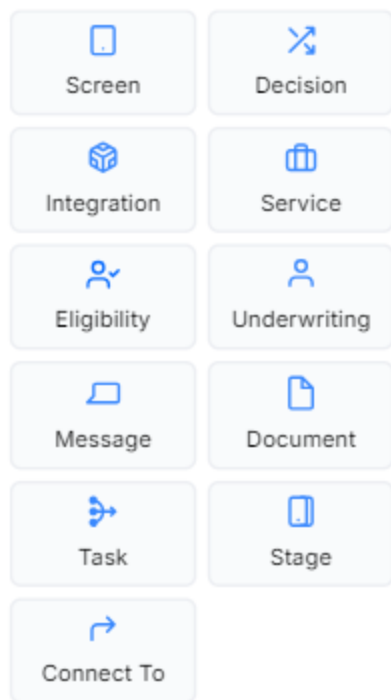
In the settings screen, select the associated products, actors, and channels to enhance the journey context and receive relevant options during the design process.

**IMPORTANT!**

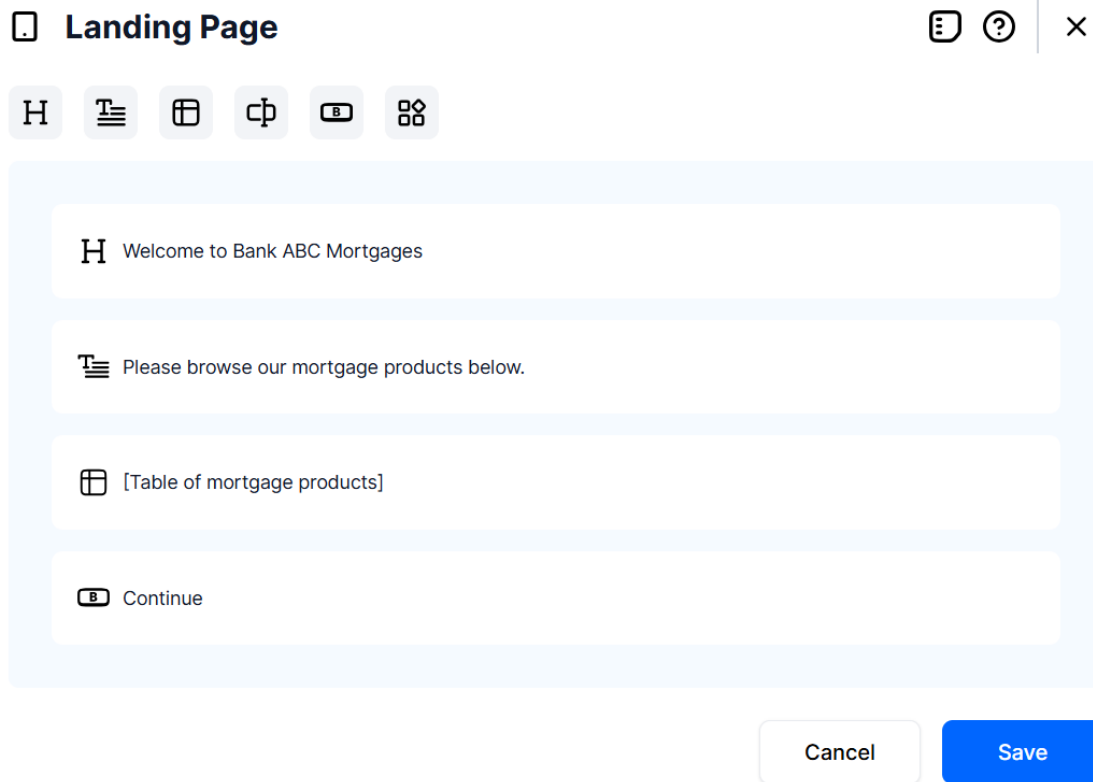
You must link the journey to at least one approved product.

**Add Journey Elements**

Click a + sign on the on the Journey Designer canvas at the desired insertion point to add a new journey element. This opens a pop-up window, allowing you to select the type of element you wish to add.

**Screen**

A screen represents the user interface displayed on screen at a specific step in the digital journey. The UI elements (headings, text, buttons/actions, tables, inputs/data attributes, widgets, etc.) serve as reference for designing the layout of the associated form driven flow step.



The available data attributes are based on product data configurations. Newly added data attributes are included in the journey's data domain. Data attributes serve as reference for defining the data model of the associated form driven flow.

#### Decision

Decisions apply conditional logic to the journey's data to select between alternative navigation paths. You can choose between two or more navigation paths depending on your navigation logic. For example, underwriting assessments often result in three possible navigation paths: "Approved," "Rejected", and "Manual Verification".

**Approve/Reject**

**If** Condition: Not Eligible → connect to → Rejected

**Else If** Condition: Under 25 → connect to → Manual Check

**Else** Condition: Type in your condition... → connect to → Collect ID Data

[+ Add path](#)

Cancel Save

In its simplest form, a decision has an **If** navigation path at the top and an **Else** navigation path at the bottom. If the Else path is linked to another decision element, any additional paths are added as **Else If** paths. Otherwise, additional paths are added at the bottom. This makes them the new Else path, converting the previous Else path into an Else If path. You can reorder the navigation paths using the grab handles on the left of each path to drag it in the desired position (top for If, bottom for Else, or in-between for Else If).

#### NOTE

Only **Else** paths may be linked to another decision element. This prevents them from being moved out of position in this situation.

Use the trash bin icon to the right side of a navigation path to delete it.

#### NOTE

Only **Else If** navigation paths can be deleted.

Navigation paths can either connect to existing elements on the canvas (resulting in a new ["Connect To" on page 205](#) element downstream) or remain unlinked for later connection.

Navigation paths are used to define flow control rules in the associated form driven flow. Use meaningful titles and condition labels to enhance clarity when previewing the journey.

## Condition Editor

You can use the point-and-click editor to define complex conditional logic for your **If** and **Else If** navigation paths without writing any code. To do so:

1. Click the **+ Add Expression** link under the desired If or Else If condition
2. Use the point-and-click interface to define your expression.

In its simplest form, a condition is based on a single form field evaluation (e.g.: Name is not blank). The available operators depend on the data type of the form field: generic operators such as *is blank* or *is not blank*, text specific operators such as *starts with* or *contains*, date specific operators such as *years since* or *days until anniversary*, etc.

You can also create complex boolean logic by grouping conditions hierarchically with logical operators:

- AND - All conditions are true.
- OR - At least one condition is true.
- NOT AND (NAND) - Not all conditions are true (at least one is false).
- NOT OR (NOR) - All conditions are false.

### Integration

Integrations are interactions with core systems or external service providers that perform specialized functions (e.g., fraud detection, identity verification, risk assessment, or credit scoring) at a specific point of the journey.

To select an existing integration, use the list on the left to enable or disable filters for the types of integrations you wish to browse or the search box to look for the desired integration by name.

Core Systems

eSignature

Identity Verification

KYC/KYB

Credit Scoring

Fraud Detection

Payments

Risk Assessment

Data Prefill

Data Quality

Other

Search integrations

Onfido

Onfido Studio eIDV

KYC/KYB • DCI

Electronic Identity Validation using Onfido Studio integration

Available

IDNow

IDNow eIDV

KYC/KYB • DCI

Electronic Identity Validation using IDNow integration

Available

Namirial

Namirial eSign

eSignature • DCI

Electronic Signature using Namirial integration


Available




Create New


Click **+ Create New** to open the integration editor and configure your own custom integration.

TABLE OF CONTENTS

197

 **Identity Verification**

 | 




Integrated system name

Jumio the Great

Select method type


Identity Verification





Integration method description

Check for fraudsters by analyzing the identity document content.


Inputs



 Aa ID card Text

+ Add input

Outputs

 ☒ Fake Boolean

+ Add output

Cancel

Save

TABLE OF CONTENTS





198


To ensure readability on the canvas, define a clear and relevant title for the integration action, such as "Check Account". Specify the inputs and outputs using data attributes to guide the implementation of the journey. For best effect, attach the logo of the integrated system.

### **Service**

Services facilitate interactions with platform back-office systems (either pre-built or custom-designed) that implement business logic to retrieve data, execute specific calculations, update other systems, and others.

The FintechOS platform includes a variety of pre-packaged services for tasks such as claims management, billing and collection, or policy administration. Additionally, you can build your own custom services or even orchestrate multiple services together.


 **Check Claim**   | 

 Upload icon




Description




Check if the claimed loss event falls under the policy scope and validate the claim.

Type

Claims Management 




Inputs

 Aa Claimed loss Text  

 # Claim Amount Numeric  

[+ Add input](#)

Outputs

 ● Valid Claim Boolean  

[+ Add output](#)

Cancel





Save


When specifying a service, define the input and output data attributes that the journey will send to or receive from the service.

## Services Orchestration

In complex scenarios, where you need to coordinate and integrate multiple components (such as back-office services, core systems, external service providers, tasks, etc.) to deliver a specific business outcome, you can use the *Services Orchestration* service type.

These services include a **Go To Service** link, which opens a dedicated editor where you can design a sub-flow for your service orchestration.


 **Advanced Checks**   | 



Description


Service subflow

Type

Services Orchestration 

[Go To Service](#)

Inputs

 # Card ID

Cancel

Save

The sub-flow is encased in a single "Stage" on page 204, and only supports "Decision" on page 194, "Integration" on page 196, "Service" on page 199, "Journey Designer" on page 191, "Journey Designer" on page 191, "Message" on the next page, "Document" on page 203, "Task" on page 203, and "Connect To" on page 205 elements.

Services orchestration flows are displayed on the Journey Designer canvas as a single element. To view or edit the underlying sub-flow, use the **Go To Service** link from the service's editor or access it via the "[Service Orchestration](#)" on [page 207](#) tab in the Journey Designer's Advanced mode.

#### Message

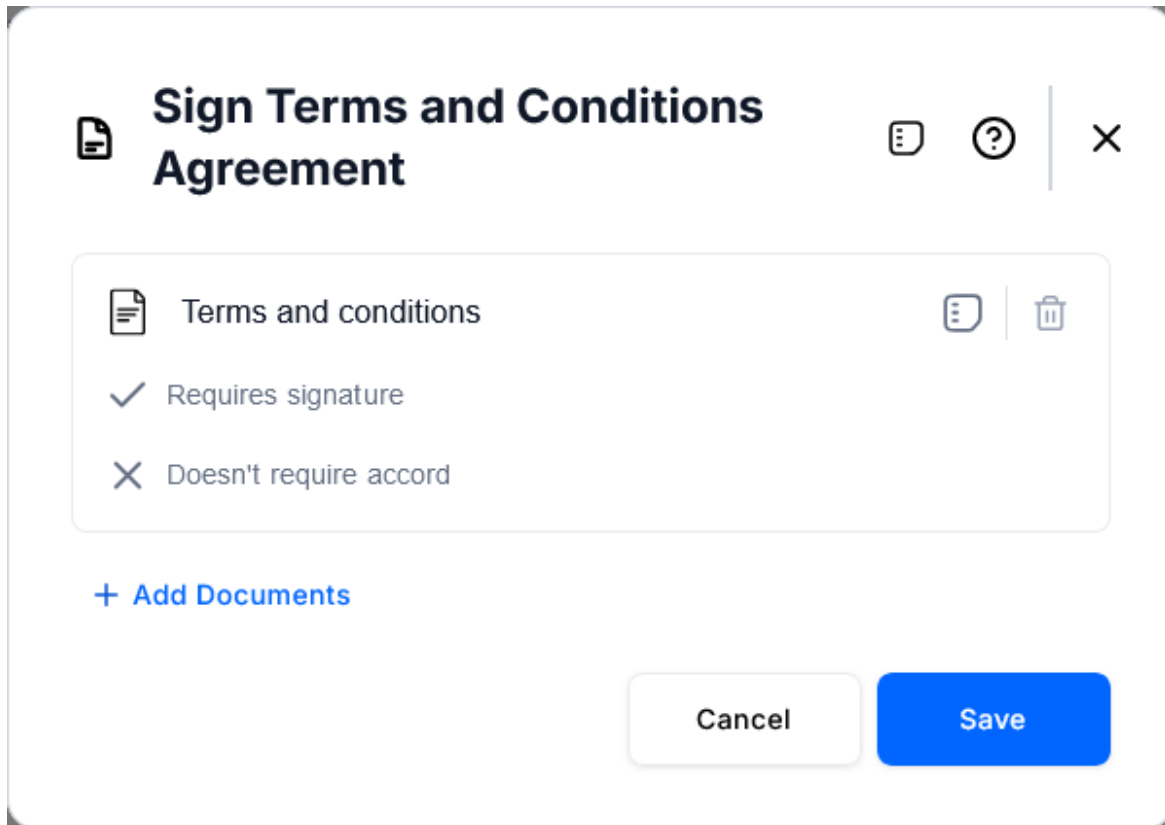
Messages enable communication with users through alternative channels such as push notifications, SMS, and email. Examples include sending one-time passwords via SMS or emailing contract documents. A set of built-in providers for both email and SMS is available, with support for additional providers coming soon.

The screenshot shows a configuration window titled "Loan Approval Confirmation" with a close button (X) and help/question mark icons. It features four tabs: "Email", "Sms" (selected), "Mobile Notification", and "Other". Below the tabs, there is a "Provider" section with a dropdown menu showing "Vonage" and a status indicator "Available". The "To" field is labeled "Applicant". The "Message" field contains the text "Congratulations. Your loan has been approved." At the bottom right, there are "Cancel" and "Save" buttons.

Define a relevant title and a meaningful sample text to ensure good clarity on the canvas.

**Document**

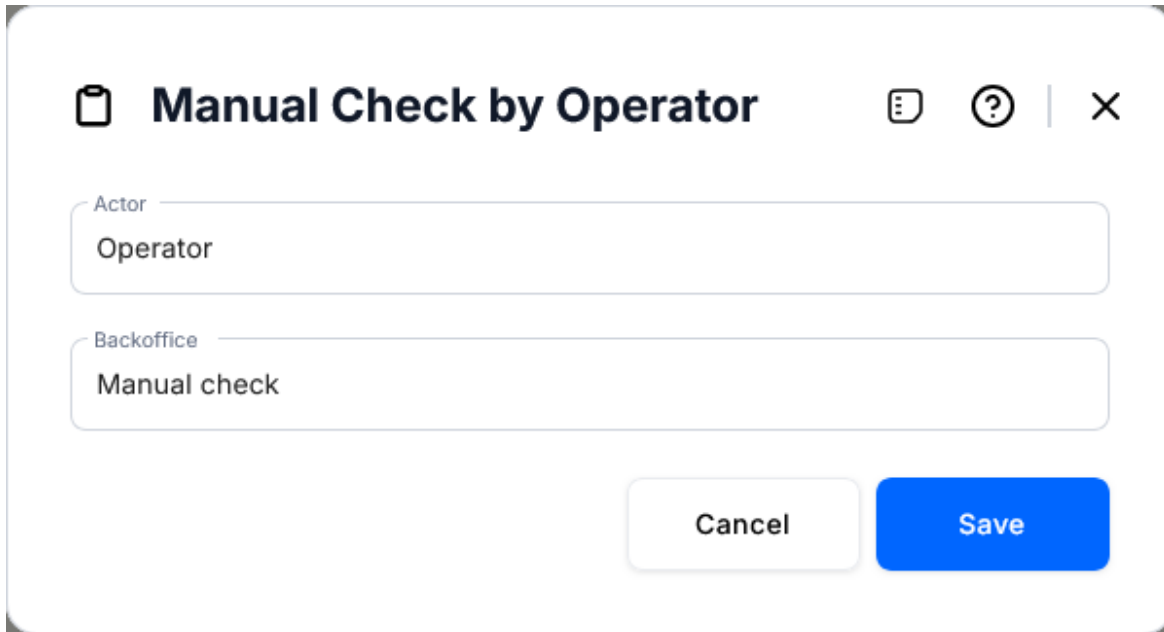
Documents are files that the user must either provide (e.g., proof of identity, income, or ownership) or receive (e.g., terms and conditions or GDPR agreements). You can also specify whether the document requires the user's agreement and/or signature.



Assign a clear title and an appropriate label to ensure good clarity on the canvas.

**Task**

Tasks are back-office operations that must be completed by a different user, typically a company employee, before the journey can proceed. This may involve manual underwriting of an application, initiating a follow-up action within the organization, or other similar activities.



**Manual Check by Operator**

Actor  
Operator

Backoffice  
Manual check

Cancel Save

Specify the responsible actor for the back-office task and what the task entails.

You may need to follow this element with a ["Decision" on page 194](#) to direct the journey along alternate paths based on the task outcome.

## Asynchronous Tasks

Select the **When Completed** checkbox to fork the execution flow into concurrent branches using asynchronous tasks. These tasks trigger a ["Service" on page 199](#) that runs in parallel with the main flow and rejoins at a defined synchronization point, where the main flow awaits its completion before continuing.

### Stage

Stages enhance journey clarity by allowing you to group subsequent elements together and assign them a descriptive name. For example, you might group a sequence of screens, decisions, and integration elements used to collect and validate a user's personal data in a "Data Collection" stage.

When zooming out on the canvas, stages provide a simplified view of the journey by collapsing the enclosed elements, displaying only the stages and the outlines of their contents.



#### Connect To

Navigates to an existing element on the canvas.

#### Move Journey Elements

There are two grab handles above each element that allow you to drag the element to a different position along the journey path. The first grab handle moves the individual element. The second grab handle moves the entire sub-flow from that element to the end of the current "Stage" [on the previous page](#).

When moving "Decision" [on page 194](#) elements, the decision node is detached from its original location and reattached at the new position along the **If** navigation path. The **Else** and **Else If** branches remain connected to the decision node and are moved along with it.

#### Advanced Interface

Use the **</>** button at the top right corner of the journey designer to enable or disable the advanced interface. When enabled, this mode allows you to switch between the "Data Domain" [on the next page](#) and "Service Orchestration" [on page 207](#) views. These views let you model the client-side context data and manage service orchestration flows, respectively.

### Data Domain

The data domain models the context data used by the [Presenter Component](#) to bind the data model and technical logic to the user interface via ["Technical Variables" below](#), ["Data Entities" below](#), and ["Dictionaries" on the next page](#). Whenever you add a new data attribute, you must specify a Data Domain entity, dictionary, or technical variable it belongs to.

## Technical Variables

Technical variables store the visibility states, validation statuses, and other behavioral aspects of the UI components (e.g., toggled sections based on user inputs). These variables control and influence UI behavior, rather than represent business data. Technical variables are scoped to the Presenter context and are not persisted to the backend.

1. Click the **Technical Variables** node to open the editor, then select the **Attributes** tab.  
You can organize your attributes into categories. At least one category is required. A default category named General is provided, which you can either rename or delete once you create additional categories.
2. To add an attribute:
  - From a ["Lexicon Term" on page 897](#): Click **+ Add From Lexicon**, then select the term from the list.
  - From scratch: Click **+ Create New**, then fill in the attribute details.
3. To refine an attribute's definition, open its **Properties** slide-out panel from either the technical variables editor or the Data Domain canvas.

## Data Entities

Data entities represent the structured business data the UI displays or modifies (e.g., customer information or application details). The mechanisms for defining attributes and grouping them into categories are identical to the ones used for ["Technical Variables" above](#). Unlike technical variables, however, you can create multiple data entities and define relationships between them with various cardinalities.

## Dictionaries

Dictionaries represent structured business data displayed in the UI, similar to ["Data Entities" on the previous page](#), but differ in that they contain static, predefined values (e.g., a disclaimer text or a list of allowed currencies selectable by the user).. The mechanisms for defining attributes and grouping them into categories are identical to the ones used for ["Technical Variables" on the previous page](#).

### NOTE

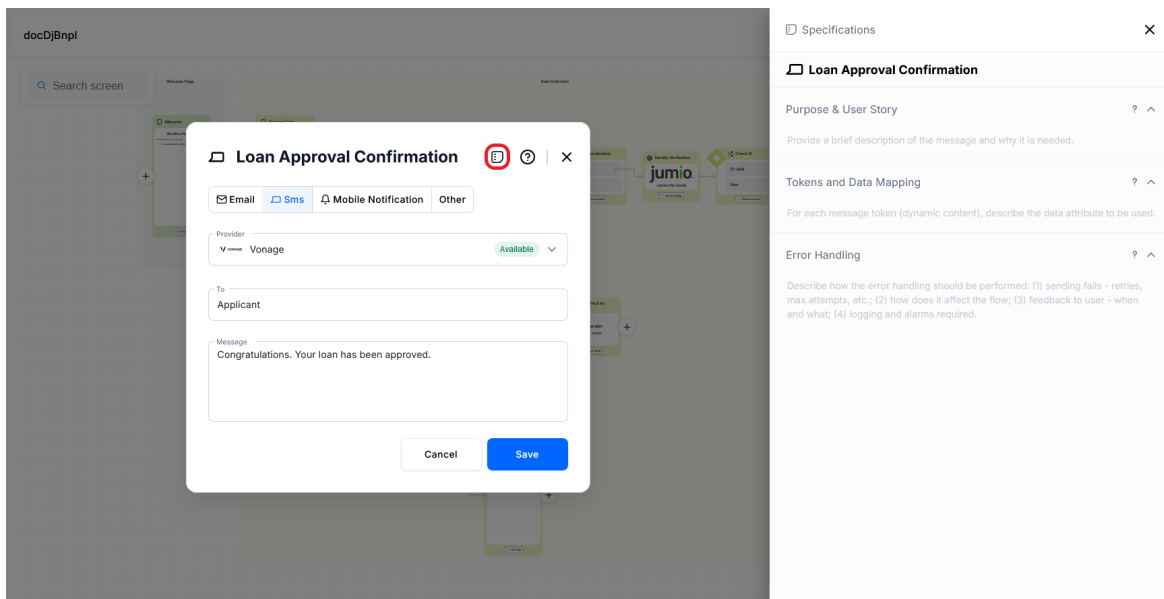
You cannot define relationships between dictionary entities.

### Service Orchestration

This is a repository for all the ["Services Orchestration" on page 201](#) elements defined within the journey, where you can expand and edit each service orchestration flow as needed.

### Edit Specifications and Generate the FSD

Specifications fields for descriptions, validations, UX definitions, error handling, and more are available at the journey, stage, element, and item levels. To add specifications for a component, click the bullet points icon at its top right corner to slide out the specifications pane.



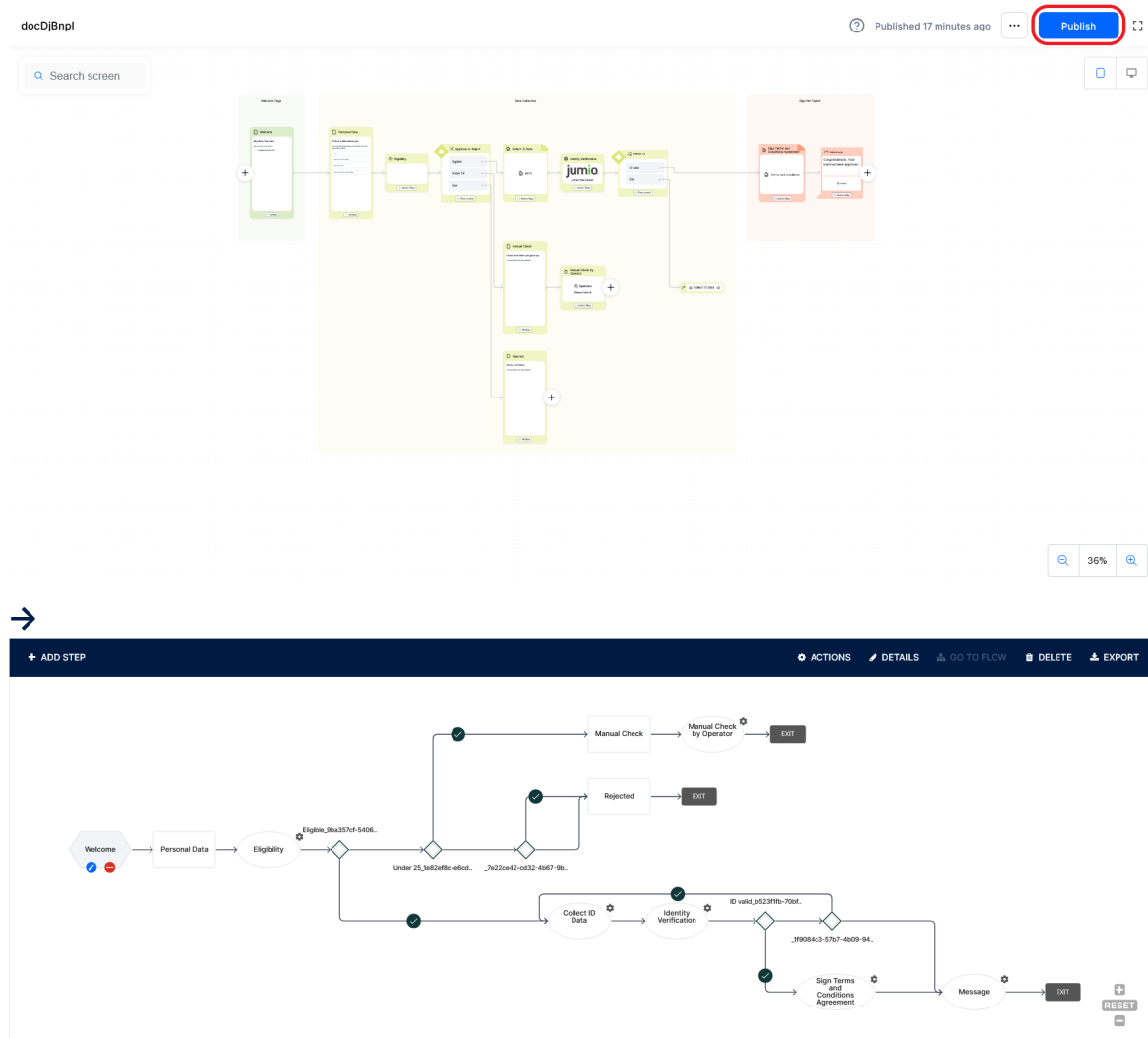
Once you have filled in the specifications for all the components, you can automatically generate a fully-formatted Functional Specifications Document (FSD) for your journey. To do so, click the ellipsis button (...) in the top right corner of the Journey Designer and select **Export document**.

## NOTE

The FSD also includes the specifications for the "Data Domain" on page 206 technical variables and data entities.

## Publish the Journey

Use the Publish button at the top right corner of the screen to propagate the journey design to the platform, automatically generating a mock-up flow with a matching "Flow Map" on page 247.



Journey elements are converted to flow elements as follows:

Journey Element	Corresponding Flow Element
"Screen" on page 193	Default UI step.
"Decision" on page 194	Each exit path generates a flow navigation rule.
"Integration" on page 196	Action step.
"Service" on page 199	Action step.
"Message" on page 202	Action step.
"Document" on page 203	Action step.
"Task" on page 203	Action step.
"Stage" on page 204	N/A.
"Connect To" on page 205	Flow navigation rule.

If you make additional changes to the digital journey, you can click the **Publish** button again to update the associated mock-up flow.

### IMPORTANT!

Once you convert the associated mock-up flow to a regular form drive flow (by attaching a data model), you can no longer publish the journey.

### Preview the Journey

After you "Publish the Journey" on the previous page, the Publish button is replaced with a **Preview** button, which allows you to visualize the journey and navigate its steps to assess the overall user experience.

## General

When creating a new digital journey in FintechOS Studio, you are presented with a few basic options in the **General** tab:

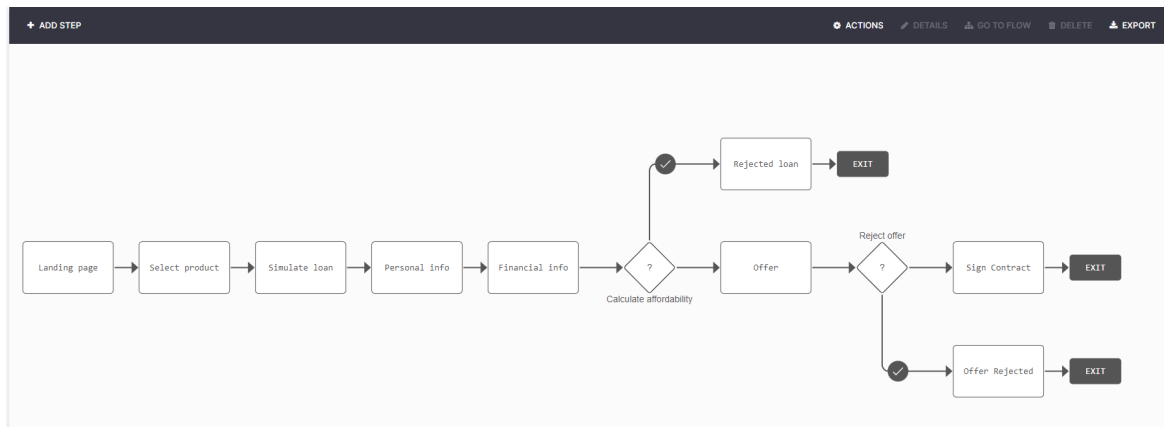
- **Name:** The name of the digital journey used internally by the system. This field is mandatory.
- **Display Name:** This is the name that will be shown in the user interface.
- **Description:** Optional description of the digital journey.

- **Exposed API Journey:** Setting that allows loading and accessing the digital journey in a custom application/ web application/ front-end, etc. Find out more in the [Digital Journey API reference](#).
- **Clone:** Duplicates the digital journey together with its associated flows. The duplicate journey will have the *\_cloned* suffix appended at the end of the name and display name.
- **New Custom Flow, New Form Driven Flow, New Form Driven Flow Mockup:** Allows you to create new "Custom Flows" on page 283, "Form Driven Flows" on page 221, or "Form Driven Mock-up Flows" on page 280 respectively and add them to the journey.
- **Digital Journey Flows:** Allows you add existing form driven flows to the journey. Use the *Is First Flow* column to select the form driven flow that initiates the journey.

## Digital Journey Map

When building a digital journey, it is possible to have many form driven flows dedicated to the same digital journey, but each has a different set of steps that may be related to the other flows. In order to build visually aesthetic journeys with customer flows and operator flows, FintechOS Studio makes it possible to see all the flows on one journey in a map.

This helps the users see all the steps, rules and the flow of the journey.



To access the map:

1. In the main menu, go to **Digital Experience > Digital Journeys**.
2. From the list, select the digital journey you are interested in.
3. Select the **Digital Journey Map** tab. From here, it is possible to perform an array of actions.

#### Add a flow to the map

1. Click **Add flow**. In the grid that appears, fill in the following fields:
  - **Name** and **Description** of the flow.
  - **Type** - can be Form Driven Flow or Custom Flow.
  - **Form Driven Flow** - select the desired flow from the list.
2. Click **Save and reload**. Add as many flows as needed.

#### Edit a step

Click a step to activate the **Details** and **Delete** buttons.

When you click Details or double-click a step, the configurations for the step are shown and it is possible to edit any needed element such as the UI, the flow and the security roles.

The Delete button erases the step from the digital journey map.

Double-clicking a rule on the map, opens the **Edit Flow Control** rule page, which allows you to edit the rule as needed. For more information, see "[Flow Control](#)" on [page 241](#).

Based on the Digital Journey Map settings, the fields in the **Digital Journey Navigation Rules** tab will be populated with the step transitions.

1 General			
2 Digital Journey Map			
3 Digital Journey Navigation Rules			
4 Analytics			
<div> <div>Export</div> <div>Refresh</div> </div>			
EntityForm	Entity Form Step	Flow Control Rule	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
BO	07_VerifyDocuments	MissingDocuments	
BO	08_ApproveLoan	RejectLoan	
Cust	02_CustomerType	IsCompany	
Cust	10_Offer	RejectOffer	

## Built-in Analytics

The digital journey analytics are a set of built-in indicators that offer insight on the performance and statistics of your digital journey. The data indicators calculate the success ratio of your digital journey, the number of Not Completed vs. Completed digital journeys, completion time, abandonment rate, average time spent on a specific step, etc.

### IMPORTANT!

The built-in analytics require the ["Digital Journey Context"](#) on page 320 to be configured on the form driven flow that initiates the journey.

### IMPORTANT!

In order to view the analytics charts, the user must have the **DjAnalytics** or **Developer** security role assigned to them in the platform. For more information, see ["Security"](#) on page 1241. All entities from the journey must have business transitions workflows defined. For more info, see ["Business Workflows"](#) on page 418.

### IMPORTANT!

If you customized the journey's form driven flows navigation by replacing the default Next buttons with custom controls, you must include the `ebs.logNext` command in the event handler that triggers the transition to the next step in the flow. Otherwise, the journey analytics will not function properly.

In the **General** tab of your digital journey, under **Digital Journey Flows**, there is the **Digital Journey States** grid. This displays all distinct entities that have business states and are part of the form driven flows included in the digital journey. You must match the entity business states to the corresponding Journey States that encompass the key phases of the user journey:

- **N/A (default)**
- **In progress** – journeys started but not yet completed.
- **Expired/Abandoned** – journeys where the user quit the process or did not take any action within a certain amount of time.
- **Completed successfully** – journeys that resulted in a contracted loan, or a loan disbursement.
- **Completed unsuccessfully** - journeys that didn't result in a completed loan application, even though the user went through all the steps of the journey.

Digital Journey States

[Refresh States](#)

[Export](#) [Refresh](#)

<input type="checkbox"/> Entity	Business Workflow	Business Workflow Status	Journey State
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Loan	Loan	Expired	Expired
Loan	Loan	InProgress	In Progress
Loan	Loan	RejectedLoan	Completed Unsuccessfully
Loan	Loan	RejectedOffer	Completed Unsuccessfully
Loan	Loan	Signed	Completed Successfully

A more in-depth representation is available as charts, in the **Analytics** tab. Here, you can filter the results by day, week, month, year, or customize your selection to show analytics from a specific time frame.

1 General 2 Digital Journey Map 3 Digital Journey Navigation Rules 4 Analytics

Today Yesterday Week **Month** Quarter Year [View Analytics](#)

Start Date End Date Business Unit

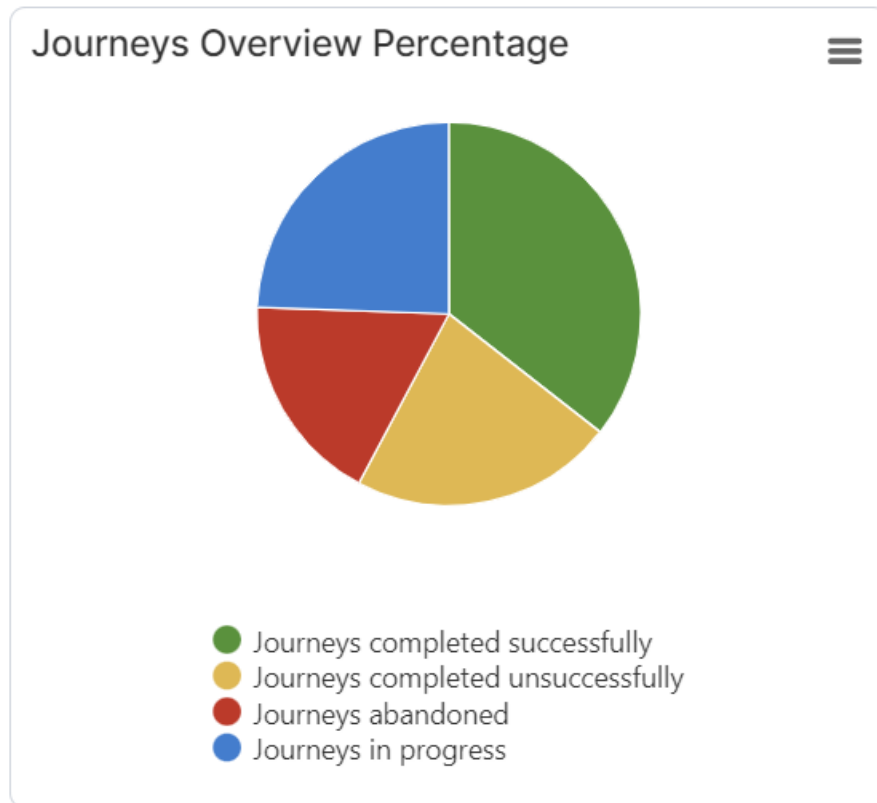
☐ Hierarchy

The charts can be downloaded individually in PNG, JPEG, PDF, and SVG format. The following statistics are included in the performance indicators charts:

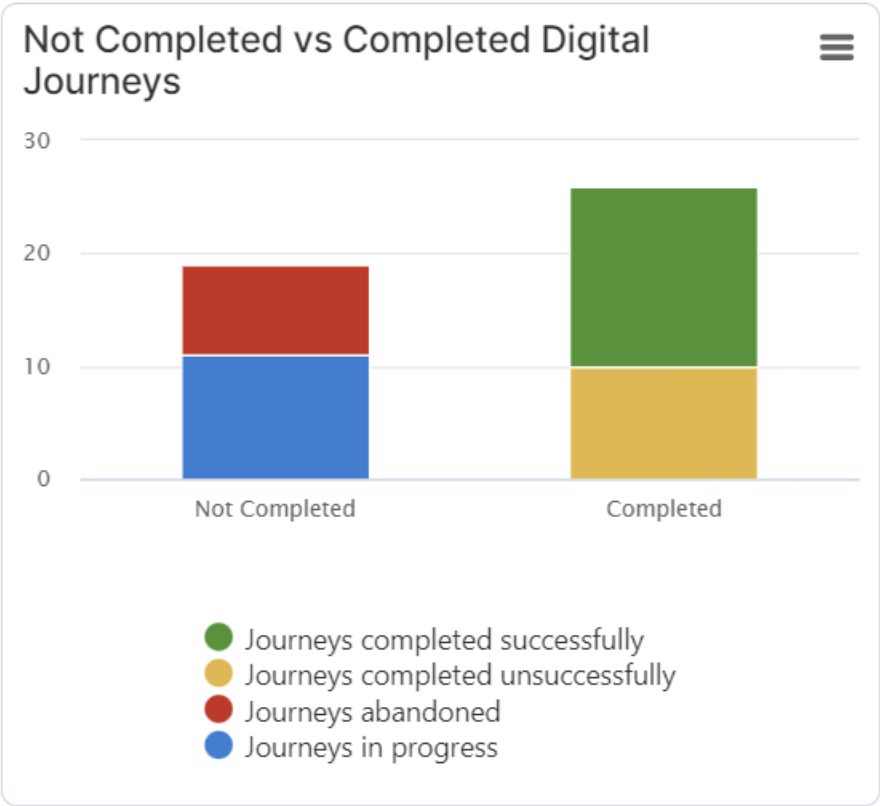
- **Journeys Overview** – shows the total number of the digital journeys and their state and outcome: in progress, abandoned, successfully completed, and unsuccessfully completed.



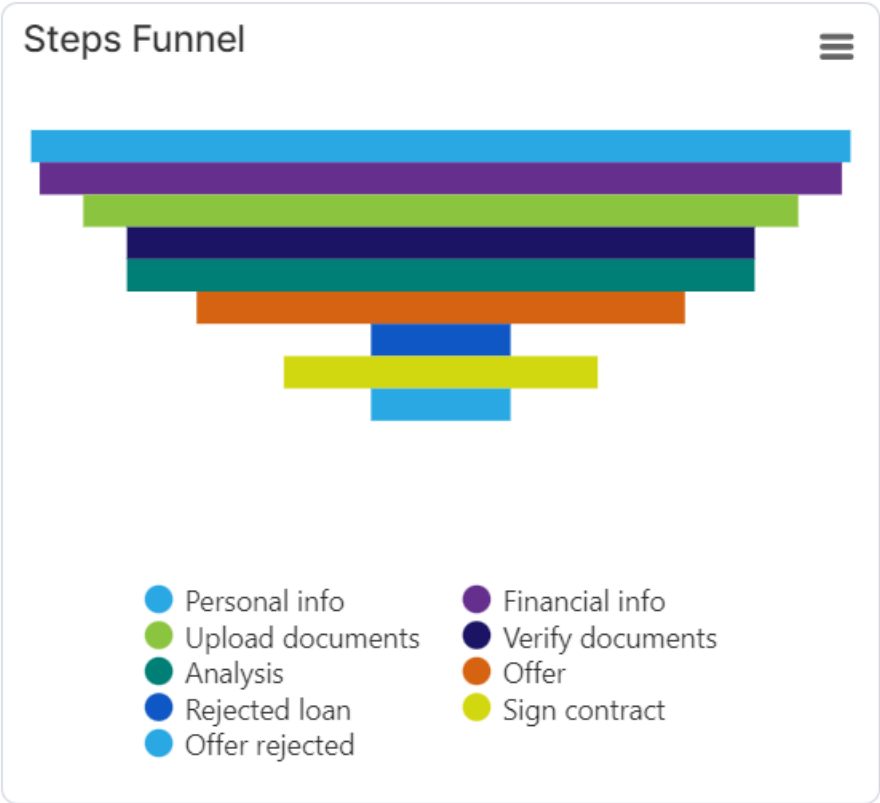
- **Journeys Overview Percentage** – shows the percentages of the different states of digital journey instances: in progress, abandoned, completed successfully, and completed unsuccessfully.



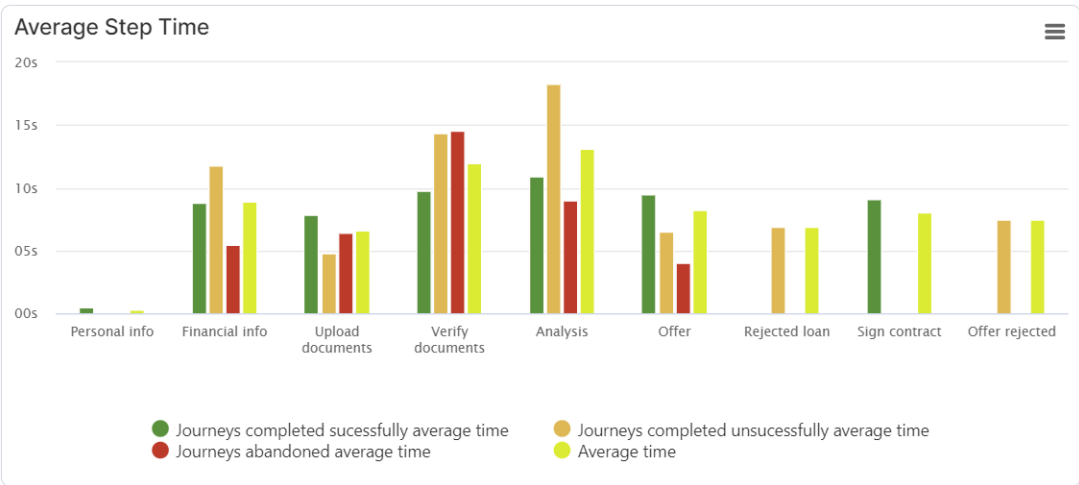
- **Not Completed vs Completed Digital Journeys** – shows the number of not completed (in progress / abandoned) and completed (successfully / unsuccessfully) digital journeys.



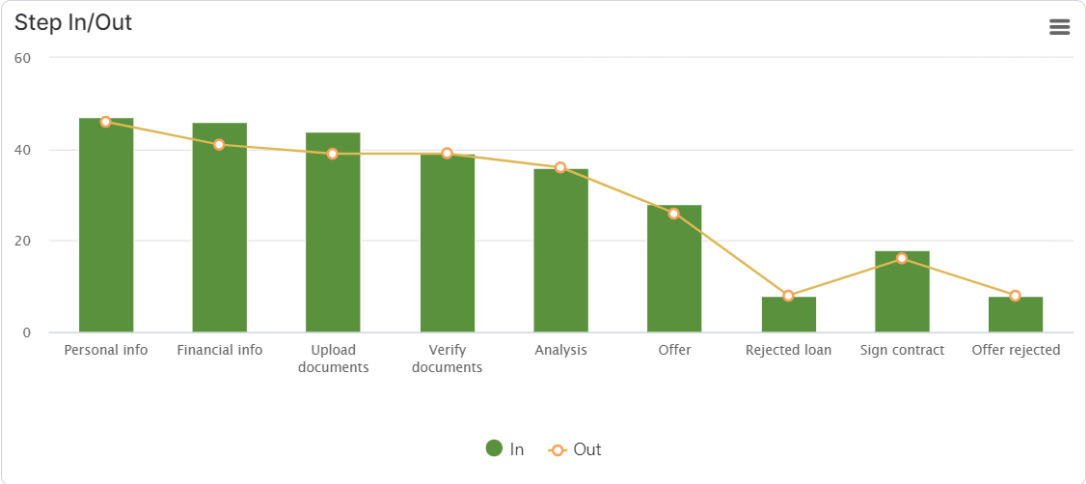
- **Steps Funnel** – shows all the steps of the journey and the number of remaining users when transferring to subsequent steps, as they begin the journey and navigate from one step to another. Each step shows the number of users who have gone through it, counting in progress, completed successfully, unsuccessfully, and abandoned digital journeys.



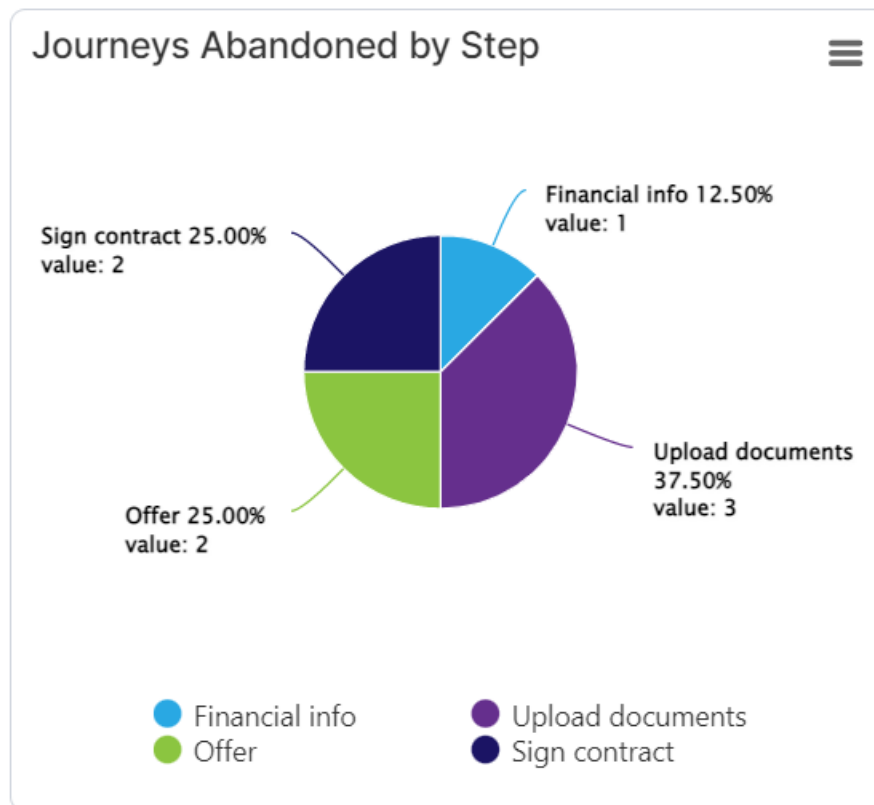
- **Average Step Time** – shows the average amount of time spent per step and whether the specific step is associated with a successful/unsuccessful/abandoned digital journey.



- **Step In/Out** – similar to the Funnel chart, this shows the number of users who went through a step of the digital journey (Step In) and the number of users who moved to a different step from there (Step Out), counting the number of entries per instance. When the user goes through the same step, if the flow control/navigation rules require them to do so, it is considered an entry. It applies to in progress, completed successfully, unsuccessfully, and abandoned digital journeys.

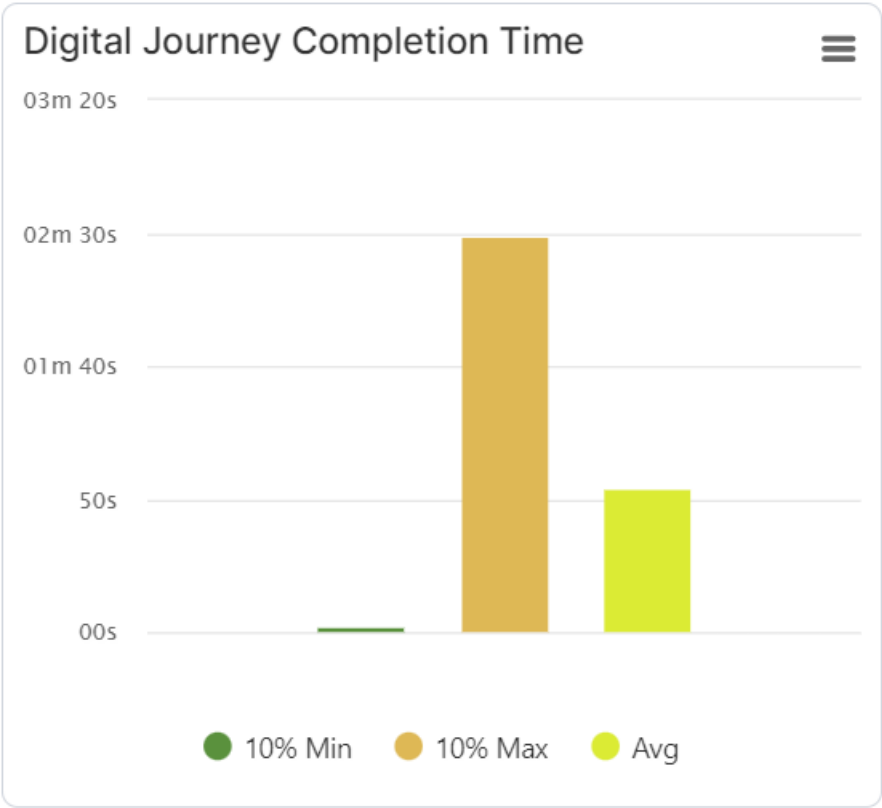


- **Journeys Abandoned by Step** – shows the specific steps (in percentages) where the digital journey instances have been abandoned.

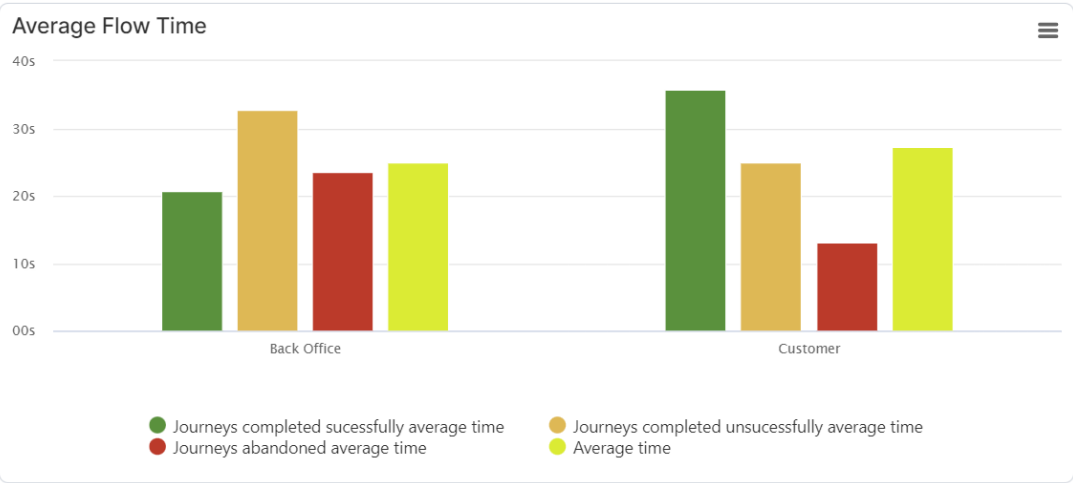


**IMPORTANT!** The charts for Steps Funnel, Average Step Time, Step In/Out, and Journey Abandoned by Step show the steps in alphabetical order, by their display names. To see the actual order of the steps, as consumed by the user, you should number the steps when configuring the flow(s) for the digital journey.

- **Digital Journey Completion Time** – displays the average of the 10% digital journeys that took the least/longest amount of time to complete (in minutes). This could help determine proportionally whether your digital journey is easy to go through or not. The “Avg” metric displays an average between all digital journey instances. This chart counts the digital journeys that are completed successfully, unsuccessfully, or abandoned.



- **Average Flow Time** – shows the average amount of time spent (in minutes) on specific form driven flows that are part of the digital journey.



## Form Driven Flows

Form driven flows are a series of steps that guide users through a process to accomplish a specific objective, such as filling out a loan application or onboarding digitally. These flows are designed to simplify complex tasks by breaking them down into manageable steps. For intricate scenarios, multiple form driven flows can be combined to create ["Digital Journeys" on page 189](#), enabling a seamless and efficient user experience.

### Prerequisite

Before creating a form driven flow, you must have a clearly defined data model for the flow (an entity and its attributes). A form driven flow is specifically linked to a primary business entity (the **main entity**). If you intend to incorporate extra attribute fields within your flow, you must ["Extend the Data Model" on page 1](#) of the main entity by adding the relevant data extensions.

## Create a form driven flow

1. From the menu, go to **Digital Experience > Customer Journeys > Form Driven Flows**.
2. Click **Create**.

<

1 General

2 Data Model

3 Flow Map

4 Steps

5 Field Options

6 Filtered Fields

7 Header Items

8 Actions

9 Advanced

Name

ab\_GammaInsurance

Display Name

GammaInsurance

Description

This is an application for a home insurance.

Show Tooltips

User Settings

☐ IsDefault

☐ Is Default For Edit

Clone Form

▼ Properties

Hide Business Workflow

Default

Read Only

Default

Disable Save Keyboard Shortcut

Default

Show Bullets Progress Bar

Default

Flow Title

Default

Style Sheets

Select items to include

Hide Business Transaction

Default

Hide Action Buttons

Default

Disable prompt for unsaved changes

Default

Save automatically data on leave

No

3. In the **General** section, provide the form driven flow's general information:

Field	Description
Name	<p>The name of the form driven flow used internally by the system. This field is mandatory.</p> <div> <p><b>NOTE</b></p> <p>A naming convention is an important part in a well-built data model. We recommend you to use PascalCaseNames (upper camel). The Name starts with an uppercase letter, as do all additional words. For example: FTOS_CMB_BaseAccount.</p> </div>
Display Name	This is the name displayed in the user interface. Choose a suggestive name.
Description	Optional description of the form driven flow.
Show Tooltips	<p>Select how you want the tooltips to be shown for specific attributes on journeys in the user interface. The following values are available:</p> <ul style="list-style-type: none"> <li>• User Settings (default value). Give users the possibility to show tooltips by toggling on / off the Tooltips button displayed at the top-right corner of the screen.</li> <li>• YES. Always show tooltips in the FintechOS Portal UI when users hover their mouse over attribute fields which have tooltips. The users cannot toggle the tooltips off.</li> <li>• No. Never show tooltips in the FintechOS Portal UI.</li> </ul>

Field	Description
Is Default	Sets the form driven flow as the default method for adding records to the flow's main business entity (see <a href="#">"Prerequisite" on page 221</a> ).
Is Default For Edit	Sets the form driven flow as the default method for editing records of the flow's main business entity (see <a href="#">"Prerequisite" on page 221</a> ).
Hide Business Workflow	Hides the entity record's state and state transition options in the end-user interface. For more information about business workflows, see <a href="#">"Business Workflows" on page 418</a> .
Read Only	Prevents end-users from making changes to the displayed form fields.
Disable Save Keyboard Shortcut	Prevents users from saving and reloading the form by pressing the Ctrl+S keyboard shortcut.
Flow Title	<p>Selects the title displayed on each step (see <a href="#">"Add and Configure Steps" on page 231</a>):</p> <ul style="list-style-type: none"> <li>• use display name - displays both the flow name and the step name</li> <li>• show only step display name - displays only the step's name</li> </ul>
Style sheets	Select the style sheet(s) you wish to use. If you add multiple style sheets, the order in which you select them matches the order in which they are applied. For details, see <a href="#">"Style Sheets" on page 1209</a> .
Hide Business Transaction	Hides the Business Transactions button at the top right corner of the screen, which displays the business workflow transitions of the record. For more information, see <a href="#">"Business Workflows" on page 418</a> .

Field	Description
Hide Action Buttons	Hides the Actions button at the top right corner of the screen, which allows you to open action groups or reports associated with the form driven flow (for details, see <a href="#">"Defining Action Groups" on page 263</a> and <a href="#">"Add Power BI Reports to Digital Journeys" on page 1035</a> ).
Disable prompt for unsaved changes	When navigating away from the form driven flow, for instance by clicking the Back button, if the form contains unsaved data, the user will not be prompted to save the changes.
Save automatically data on leave	When navigating away from the form driven flow, for instance by clicking the Back button, any unsaved data in the flow will be automatically saved.

4. Click the **Data Model** tab and from the **Entity** field, select the entity whose data model you wish to use (main entity).

The screenshot shows the 'Data Model' tab in the Fintechos Studio interface. At the top, a breadcrumb navigation bar includes tabs from 1 to 10, with '2 Data Model' selected. Below this, the 'Entity' is set to 'FTOS\_PWC\_Loan'. The 'Business Entity Extensions' section contains a table with columns 'Name', 'Extension Type', and 'Relation Attribute'. It lists an extension named 'Applicant' with a 'One to One' relationship and 'FTOS\_PWC\_Applicantid' as the relation attribute. Below this is the 'ALLOWED Attributes' section, which includes a table with columns 'IsReadOnly', 'Name', and 'Display Name'. A dropdown menu is set to '(All)'. The table lists ten attributes: Affordable, Email, FirstName, FTOS\_PWC\_Productid, Installment, LastName, LoanValue, MobileNumber, MonthlyExpenditure, and MonthlyIncome, each with an unchecked 'IsReadOnly' checkbox. At the bottom, there are pagination controls showing '5', '10' (selected), and '20' on the left, and '1' and '2' on the right.

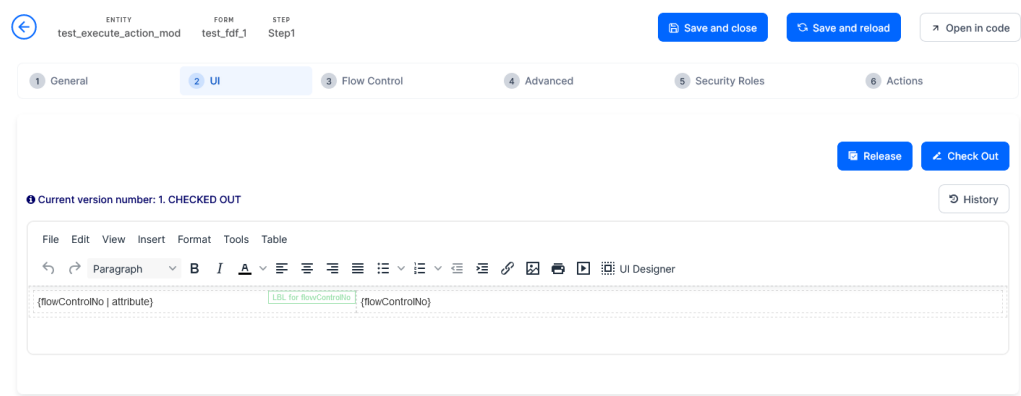
- 5. Click **Save and reload** .
- 6. Add any required **Business Entity Extensions** to the data model (see "[Prerequisite](#)" on [page 221](#)).
- 7. If you wish to restrict the flow's access to specific attributes in the data model, insert the corresponding values in the **Allowed Attributes** and **Allowed Virtual Attributes** sections. By default, all attributes in the data model are available in the form's context. Once you start populating the allowed attributes sections, the form will be restricted from accessing any attributes that were not explicitly defined in these sections.

3 Design the flow UI

Click the **UI** tab and use the "UI Designer" on page 323 to define the form's layout.

Legacy Template Designer (deprecated)

For form driven flows created in FintechOS Platform versions prior to v.24, the UI tab displays the legacy template designer which, in addition to the UI designer, provides an HTML editor that allows you to configure the form's HTML elements.



Legacy form driven flows also allow you to use the "Advanced Code Editor (deprecated)" on page 1142 or the "Code Editor" on page 1108 respectively to edit their HTML code directly.

The form's UI template supports the following tokens:

Token	Description
{AttributeName}	Displays the corresponding field in the UI. <div><b>NOTE</b> The attribute name must be included between curly brackets; otherwise, a simple text will be displayed on the page instead of the actual field.</div>

Token	Description
{#RelationshipName, view: viewName#}	Generates a view provided by relationship and by view. The viewName is optional and specifies which view to generate. If viewName is not provided, the default view will be displayed on the data form.
{#RelationshipName, view: viewName, editmode:cell#}	Generates a view provided by relationship and by view. This view allows inline editing, meaning that you can edit cells one by one directly in the grid, without opening specific records.  <b>NOTE</b> In order to activate inline editing for a specific cell, you must tick the <b>Allow Editing</b> checkbox displayed on the entity view column (View > View columns).
{#MKT_CampaignResponse_MKT_Campaign,nodelete,noinsert#}	Generates a view provided by relationship and by view, without the <b>Delete</b> and <b>Insert</b> buttons. You can apply the same logic (similar tokens) for the Export (noexport) or Refresh (norefresh) if you no longer need them on the view.
{\$ChartName\$}	Generates a chart based on the provided chart name.
{? entityName, view: viewName ?}	Allows you to display a view from another entity. For information on how to use it, see <a href="#">Display View from Another Entity</a> .

In the HTML template, you can link HTML elements (labels) to attributes. For information on how to do it, see "[Link Labels to Attributes \(deprecated\)](#)" on [page 270](#).

## 4 Divide the flow in steps

Breaking down a complex flow into manageable steps can improve user experience. You can group flow information into steps based on specific criteria, such as business or operational considerations. To add a step, navigate to the **Steps** tab and add as

many steps as needed based on your criteria.

If you only have one entry in the Steps section, it will not be marked as a step when displayed in the FintechOS Portal.

For information on how to add and configure steps, see ["Add and Configure Steps" on page 231](#).

The default execution order of a digital journey which is comprised of steps is given by the OrderIndex of the steps, as set in the Steps tab. You can change the default flow of a digital journey by using the ["Flow Control" on page 241](#) feature.

## Define access to the flow

To restrict access to the form driven flow to designated ["Security Roles" on page 1245](#), click the **Security Roles** tab and add the security roles who should have access. If no security roles are added here, all FintechOS Portal users will be able to access the flow.

## Add Code to the Flow

For advanced scenarios that require elaborate customizations, the **Advanced** tab allows you to attach client-side or server-side JavaScript code to your flow. For details, see ["Code Execution Sequence" on page 301](#).

## Clone a form driven flow

To duplicate a form driven flow with all of its configurations, in the **General** tab, click **Clone Form**.

The screenshot shows the 'General' tab of the Fintechos Studio interface. The form is named 'ab\_GammaInsurance' with a display name of 'GammaInsurance'. The description is 'This is an application for a home insurance.' The 'Show Tooltips' section has a dropdown menu set to 'User Settings'. There are checkboxes for 'Is Default' and 'Is Default For Edit', both of which are currently unchecked. A blue 'Clone Form' button is visible. Below the 'Clone Form' button is a 'Properties' section with several settings: 'Hide Business Workflow' (Default), 'Read Only' (Default), 'Disable Save Keyboard Shortcut' (Default), 'Show Bullets Progress Bar' (Default), and 'Flow Title' (Use Display Name).

Insert a name for it and click **Clone**. The new form driven flow opens immediately for editing.

The screenshot shows the 'General' tab of the Fintechos Studio interface. A 'Clone Form' dialog box is open in the center. The dialog has a title bar with a close button (X). Inside the dialog, there is a 'New Form Name' label and a text input field containing 'Cloned form'. Below the input field is a 'Clone' button. The background shows the same form settings as the previous screenshot, but the 'Clone Form' button is now disabled.

**NOTE**  
The data set behind it is not cloned, just the flow and its settings.  
The **Is Default** and **Is Default For Edit** settings are not replicated in the cloned form driven flow, since you can have only one default form driven flow per entity.

Additional flow settings are covered in the following topics:

## Add and Configure Steps

Dividing forms in steps based on specific criteria (business, operational, or other relevant to you) is useful in complex scenarios where you have to display a lot of information on the entity forms and user journeys.

This sections walks you through the configurations that you need to follow to add steps and configure them.

### 1 Add a step

Entity Form Steps

Buttons: + Insert, X Delete, Export, Refresh

Name	DisplayName	Order	Digital Asset
07_VerifyDocuments	Verify Documents	1	
08_ApproveLoan	Approve Loan	2	

1. On the configuration page of the form driven flow, click the **Steps** tab.
2. In the **Entity Form Steps** section, click **Insert**. The step configuration page appears, displaying the general tab.
3. Provide the following step information:
  - **Name** - The name of the step that will be used by the system. We recommend you use PascalCaseNames (upper camel). The Name starts with an uppercase letter, as do all additional words.
  - **Display Name** - The step name to be displayed in the user interface.
  - **Step Type** - Select either Default, Processor (see " [Custom Processor Steps](#)" on page 238), or Action (see " [Action Steps](#)" on page 240).
4. Fill in any desired step properties:
  - **Show "Next" button** - Display or hide the button to navigate to the next step.
  - **Show "Previous" button** - Display or hide the button to navigate to the previous step.

- **"Next" button label** - Override the default "Next" button label with a custom text.
  - **"Previous" button label** - Override the default "Previous" button label with a custom text.
  - **Disable prompt for unsaved changes** - When navigating away from the step, for instance by clicking the Back button, if the step contains unsaved data, the user will not be prompted to save the changes. Overrides the equivalent setting configured at the flow level.
  - **Save automatically data on leave** - When navigating away from the step, for instance by clicking the Back button, any unsaved data in the step will be automatically saved. Overrides the equivalent setting configured at the flow level.
5. Click **Save and reload**. The step configuration page appears, containing more tabs, displayed by default on the **General** tab. You can configure the step, by clicking the tabs and making the desired settings.

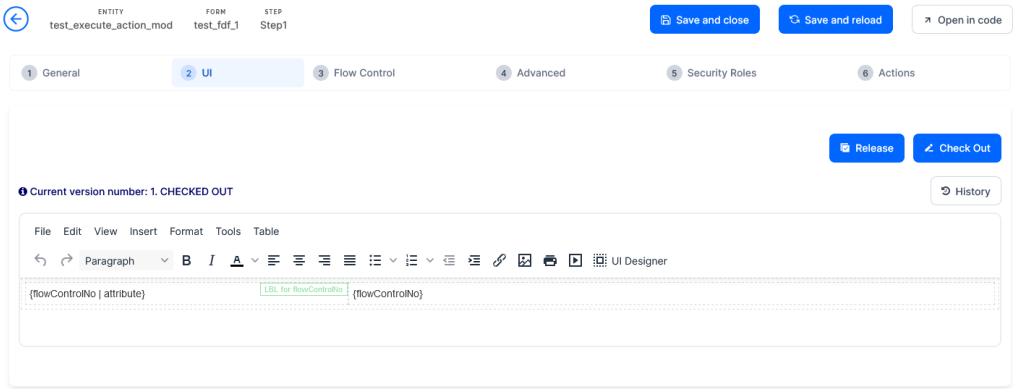
The screenshot shows the 'UI' tab of the FintechOS Studio configuration interface for 'Step2'. The interface includes a top navigation bar with tabs for ENTITY (Documentation Entity), FORM (docFlow), and STEP (Step2). On the right, there are buttons for 'Save and close', 'Save and reload', and 'Open in code'. Below the navigation bar, a horizontal menu shows six tabs: 1 General, 2 UI (selected), 3 Flow Control, 4 Advanced, 5 Security Roles, and 6 Actions. The main configuration area is divided into two columns. The left column lists properties: Name, DisplayName, Step Type, Is Additional, a collapsed 'Properties' section, Show 'Next' button, Show 'Previous' button, 'Next' button label, 'Previous' button label, Disable prompt for unsaved changes, and Save automatically data on leave. The right column contains the corresponding input fields: 'Step2' for Name, 'Enter your data' for DisplayName, 'Default' for Step Type, a checkbox for Is Additional, dropdown menus for Show 'Next' button and Show 'Previous' button (both set to 'Default'), text input fields for 'Next' button label ('Payment >') and 'Previous' button label ('< Shipping'), a dropdown for Disable prompt for unsaved changes (set to 'Default'), and a dropdown for Save automatically data on leave (set to 'No').

## 2 Design the step layout

Click the **UI** tab . Use the "[UI Designer](#)" on [page 323](#) to configure the step's user interface.

## Legacy Template Designer (deprecated)

For steps created in FintechOS Platform versions prior to v.24, the UI tab displays the legacy template designer which, in addition to the UI designer, provides an HTML editor that allows you to configure the step's HTML elements.



Legacy steps also allow you to use the ["Advanced Code Editor \(deprecated\)" on page 1142](#) or the ["Code Editor" on page 1108](#) respectively to edit their HTML code directly.

The step's UI template supports the following tokens:

Token	Description
{AttributeName}	Displays the corresponding field in the UI. <div><b>NOTE</b> The attribute name must be included between curly brackets; otherwise, a simple text will be displayed on the page instead of the actual field.</div>
{#RelationshipName, view: viewName#}	Generates a view provided by relationship and by view. The viewName is optional and specifies which view to generate. If viewName is not provided, the default view will be displayed on the data form.

Token	Description
{#RelationshipName, view: viewName, editmode:cell#}	<p>Generates a view provided by relationship and by view. This view allows inline editing, meaning that you can edit cells one by one directly in the grid, without opening specific records.</p> <div> <p><b>NOTE</b></p> <p>In order to activate inline editing for a specific cell, you must tick the <b>Allow Editing</b> checkbox displayed on the entity view column (View &gt; View columns).</p> </div>
{#MKT_CampaignResponse_MKT_Campaign,nodelete,noinsert#}	<p>Generates a view provided by relationship and by view, but the <b>Delete</b> and <b>Insert</b> buttons are not displayed on the view. You can apply the same logic (similar tokens) for the Export (noexport) or Refresh (norefresh) if you no longer need them on the view.</p>
{\${ChartName\$}}	Generates a chart based on the provided chart name.
{? entityName, view: viewName ?}	Allows you to display a view from another entity. For information on how to use it, see <a href="#">Display View from Another Entity</a> .

In the HTML template, you can link HTML elements (labels) to attributes. For information on how to do it, see ["Link Labels to Attributes \(deprecated\)" on page 270](#).

### 3 Set advanced flow control rules

Use the ["Flow Control" on page 241](#) tab if, on leaving the step, you wish to automatically:

- Override the default navigation sequence
- Apply ["Form Actions" on page 258](#).
- Apply conditional navigation rules.

#### 4 Configure event handlers

You can use logic blocks to configure event handlers through an intuitive no-code interface. You can define complex actions and conditional logic that can be triggered by attribute field updates, custom events, or at different stages of a step's execution sequence. For more information, see ["Logic Blocks" on page 313](#).

#### 5 Provide the code to be executed after the step is generated (optional)

For advanced scenarios that require elaborate customizations, the **Advanced** tab allows you to attach [client-side](#) or [server-side](#) JavaScript code to your step. For details on the execution sequence of the various form-level and step-level scripts, see ["Code Execution Sequence" on page 301](#).

##### NOTE

Variables and functions declared in other steps or in the form driven flow's Advanced section are not visible in the current step. To access such variables and functions, use the [formData.formScope](#) object.

#### 6 Define who has access to the step

If your business case requires that specific steps are available to designated ["Security Roles" on page 1245](#), click the **Security Roles** tab and add the security roles who should have access to them. If no security roles are added here, all flow users will be able to view the step.

#### 7 Actions

You can set specific ["Form Actions" on page 258](#) to trigger automatically when entering or saving a step, if specific conditions are met.

There are two grids displayed: **On Step Enter** and **On Step Save**.

The screenshot shows the 'Actions' tab in the Fintechos Studio interface. It contains two identical grids for configuring actions. The top grid is for 'On Step Enter' and the bottom grid is for 'On Step Save'. Each grid has a header with tabs: 'General', 'UI', 'Flow Control', 'Advanced', 'Security Roles', and 'Actions' (selected). Below the tabs are four search bars labeled 'Name', 'Order Index', 'Form Actions', and 'Execute Always'. The 'Execute Always' dropdown is currently set to '(All)'. Below the search bars, the text 'No data' is displayed. To the right of each grid are four buttons: '+ Insert', 'X Delete', 'Export', and 'Refresh'.

Depending on when you wish to trigger the actions (either on entering or saving the step), in the corresponding grid:

1. Click **Insert** to add a new action.
2. Fill in the following:

Field	Description
Name	Insert a name for the action.
Action type	This is automatically filled based on the grid you are populating (On Step Enter or On Step Save).
Condition expression	Insert the condition(s) that must be met to trigger the action. For details, see <a href="#">"Apply Conditional Navigation Rules" on page 243</a> for how to define a rule expression.
Form Action	Select the form actions(s) you wish to trigger from the configured <a href="#">"Form Actions" on page 258</a> .
Execute Always	By default, the action is executed only when the condition expression is met.  Check the <b>Execute Always</b> option to ignore the condition expression and always trigger the action.

3. Click **Save and Close**.

## Custom Processor Steps

This feature allows you to add an automation processor such as [Computer Vision](#), [eSign](#), [Video Streaming Processor](#), or [Face Recognition](#) to a step without writing code. This way you can rapidly build a fully automated digital journey with multiple steps, easily call processors from the flow, and simplify step-by-step navigation. To do so:

1. From the main menu, select **Digital Experience > Customer Journeys > Form Driven Flows**.
2. Select the Form Driven Flow you wish to edit by adding a processor to one of its steps.
3. After opening the form driven flow, click **4. Steps** to add a new step. For more information, see ["Add and Configure Steps" on page 231](#).
4. Click **Insert** and fill in the name of the step, display name and for Type from the Option set select the **Processor**.

5. Click **Save and reload**. The general page opens.

## 6. Fill in the fields:

The screenshot shows the 'General' tab of a configuration form. The fields and their values are as follows:

- Name:** test
- DisplayName:** test
- Step Type:** Processor
- Is Additional:** (empty)
- Settings:**
  - Processor Name:** ESign Processor
  - Processor Setting:** Esign\_AutomaticProfileExample
  - Existing Compare File:** Yes
  - Success Navigation Step:** 05\_FinancialInfo
  - Fail Navigation Step:** 01\_ConfirmEmail

Fields	Description
Name	It is automatically filled. It is the name of the step.
Display name	It is automatically filled. It is the name of the step.
Step Type	It is automatically filled. It is the name of the step.
Processor name	Select the name of the processor.
Processor settings	Select the settings you configured earlier for the processor.
Existing Compare file	When using the Face recognition or Face recognition with Liveness, this field enables the system to compare the selfie with the photo for an ID.
Success navigation step	Select the step you wish to be next if the processor step is a success.
Fail navigation step	Select the step you wish to be next if the processor step is a failure.

7. Click **Save and reload**.

8. Add the proper security roles for this step by clicking **Insert** and choosing one of the security roles. For more details, see ["Security Roles" on page 1245](#). Click **Save and close**.
9. Repeat for as many steps as needed.

## Action Steps

Action steps allow you to trigger ["Form Actions" on page 258](#) by simply navigating to a step in the form driven flow. They are similar to the default steps, except they don't have a UI.

To create an action step:

1. From the main menu, select **Digital Experience > Customer Journeys > Form Driven Flows**.
2. Select the Form Driven Flow you wish to configure. Make sure the form has the desired form action configured, otherwise create a new one (see ["Form Actions" on page 258](#)).
3. In the form driven flow interface, open the **Steps** tab to add a new step.
4. Follow the instructions in ["Add and Configure Steps" on page 231](#). In the **Step Type** field, select Action.

General	
Name	step3
DisplayName	Agree to terms and conditions
Step Type	Action
Is Additional	<input type="checkbox"/>

5. Click **Save and reload**. The **General** page will open.
6. Fill in the fields to configure the ["Flow Control" on the next page](#), Advanced, ["Security Roles" on page 1245](#) and ["Form Actions" on page 258](#). They behave as for a default step.

7. Click **Save and reload**.
8. Add the proper security roles for this step by clicking **Insert** and choosing one of the security roles. For more details, see ["Security Roles" on page 1245](#).
9. Click **Save and close**.
10. Repeat for as many steps as needed.

## Flow Control

Flow control enables you to apply custom navigation rules and/or trigger ["Form Actions" on page 258](#) when leaving a flow step.

### IMPORTANT!

The flow control is available only for Form Driven Flows.

### Flow Execution Sequence

For a form driven flow comprised of steps, the default execution order is the following:

1. The main form driven flow is executed. First the *beforegenerate.js* script of the flow is executed and then the *aftergenerate.js* script.
2. For each step of the form driven flow, when the step is activated:
  - i. Step elements (HTML, CSS) are loaded and the form driven flow is rendered.
  - ii. The *aftergenerate.js* script of the step is executed.

The execution order of a digital journey which is comprised of steps is determined by the order index of the steps set in the digital journey map (if not otherwise specified on the steps).

There are two ways in which you can change the execution order of a digital journey. On a step you can:

- Override the default next step navigation.
- Set a flow control rule that enforces a certain next step navigation when specific conditions are met.

Thus, when clicking **Next** in a digital journey, the flow is:

1. BeforeSave script on the step is executed.
2. The flow control rules of type Cancel Navigation are evaluated and only the first one is executed.
3. The data is saved.
4. The remaining flow control rules are evaluated and only the first one that is fulfilled is executed. This overrides the default next step given by the order index with the step specified in the condition.
5. The AfterSave script on the step is executed and the digital journey advances to the next step .

### Override the Default Navigation

Use the **Default Navigation Rule** section to override the default next step behavior set in the digital journey's order index.

- **Close Flow** - Marks the step as the final step of the flow (the flow closes once the step is complete). Use the **Navigate to** option to set the page to display after closing the flow:
  - **List** - An entity view.
  - **Dashboard** - A portal dashboard.
  - **Homepage** - The portal homepage.
  - **Previous context** - Back to the page that was open when the flow was initiated.
- **Navigate to another Step** - A step in the current flow.

- **Navigate to another Flow** - A step in a different flow from the same journey.
  - **Select Digital Journey Flow** - Form driven flow name
  - **Select Digital Journey Flow Step** - Step name
  - **Reference Attribute** - Attribute in the current data model that indicates the primary key of the record you wish to open in the destination form (Tick **Use Virtual Attribute** if you want to select a virtual attribute).

### HINT

Typically, you want to open a related entity record by selecting the corresponding lookup attribute. If the destination flow uses the same base entity, you may want to reuse the primary key (or, if it self-references, a lookup attribute that points back to the source entity).

### Apply Form Actions

Use the **Actions to be Performed** text box to select ["Form Actions" on page 258](#) you wish to apply on the transition to the next step.

### Apply Conditional Navigation Rules

Use the **Flow Control Rules** section to apply navigation rules and/or form actions only when specific conditions are met:

1. Click **Insert** to open the Add Flow Control Rule page.

Flow Control Rule

Name

Form Section

Description

Evaluate only for Journey

Define rule expression

And +

× Status Equals Active

Action

☐ Cancel Navigation

☐ Close Flow

☐ Navigate to another Step ☒ Navigate to another Flow

Select Digital Journey Flow

Select Digital Journey Flow Step

Reference Attribute  
the value of this attribute will be used as the primary key value for the target entity

☐ Use Virtual Attribute

Attribute

Actions to be Performed

2. Provide a **Name** for the rule. In the Form Section field, the name of the step you are editing is filled automatically.
3. Insert a **Description** for the rule.
4. If you wish to evaluate the rule only in the context of a specific digital journey, select the corresponding journey from the **Evaluate only for Journey** drop-down box. For information, see "[Digital Journey Context](#)" on page 320.

5. Use the **Define rule expression** section to set which criteria must be met for the rule to apply.
  - You can change the logical operator used to aggregate your conditions:
    - **And** (default) - All conditions/groups must be true.
    - **Or** - At least one condition/group must be true.
    - **Not And** - At least one condition/group must be false.
    - **Not Or** - All conditions/groups must be false.
  - Use the + symbol to add new conditions or groups to the expression. Groups allow you to evaluate the included conditions/groups together using the same logical operators described above (And, Or, Not And, Not Or). You can create multiple levels of subgroups (groups inside other groups) for complex boolean logic.
  - Each condition allows you make an evaluation either using a custom processor (see ["Checking with Custom Processor" on the next page](#)) or based on an attribute in the data model.
    - Some operands are generic and apply to all attributes: Equals, Does not equal, Is blank, Is not blank.
    - Some operands are specific to the attribute's data type. E.g.: Text attributes support operands such as Contains, Does not contain, Starts with, or Ends with. Numeric attributes support operands such as Is less than, Is greater than, Is less than or equal to, Is greater than or equal to, Is in between, etc.

**HINT**

You can configure flow control rules to apply only to targeted

user profiles (based on demographics, location, age, needs, previous interaction with a business, etc.). Such users can be defined as "[Customer Personas](#)" on [page 907](#), and they can be grouped in audiences (see "[Audiences Management](#)" on [page 926](#)). This comes in handy when creating a personalized, data-driven journey, adapted to specific customer needs and context.

- Date/time attributes have a small arrow next to them allowing you to expand and select a derivative value instead of the actual attribute value: Minutes since, Minutes until, Hours since, Hours until, Days since, Days until Day Of, Weekday of, Days until Anniversary, Has anniversary today, etc.
6. In the **Action** section, set up the navigation override and/or the "[Form Actions](#)" on [page 258](#) you wish to perform if the rule expression is true. Navigation setup is based on the same "[Override the Default Navigation](#)" on [page 242](#) instructions provided above, with the addition of an option to **Cancel Navigation** which stops the flow on the current page and allows you to display a custom message.
  7. Click **Save and close** to save your flow control rule.

You can add as many flow control rules as needed. Only the first rule that is fulfilled is applied (however, rules that cancel the navigation are evaluated first).

### Checking with Custom Processor

It is possible to create a rule that checks the attribute mappings of an endpoint to return the values true or false from the boolean attribute.

The screenshot shows the 'Flow Control Rule' configuration page. It has several sections: 'Name' (text input), 'Form Section' (dropdown with 'CustomerReferenceData'), 'Description' (text area), 'Evaluate only for Journey' (dropdown), 'Define rule expression' (logic builder with 'And' condition and 'Checking with Custom Processor' rule), 'Action' (checkboxes for 'Cancel Navigation' and 'Close Flow'), and 'Select Step' (dropdown). A modal window is open for the 'Custom Processor' configuration, showing 'ai\_bool' as the processor, 'Age' as the input parameter, and 'Age' as the numeric parameter. There is an 'Apply' button and a 'Navigate to another Flow' link in the modal.

1. Create the endpoint with the input and output mappings as bool in Server Automation Scripts. In the code section write the following line:
 

```
context.result = true;
```
2. Navigate to the Form Driven Flow, to the Flow Control, and click **Insert Rule**.
3. In the rule add the condition with checking the custom processor.
4. Select from the option set the name of the endpoint and map the input and output to the bool attributes you wish to have the data in.
5. Click **Apply**.
6. Select the next step where to navigate and click **Save and close**.

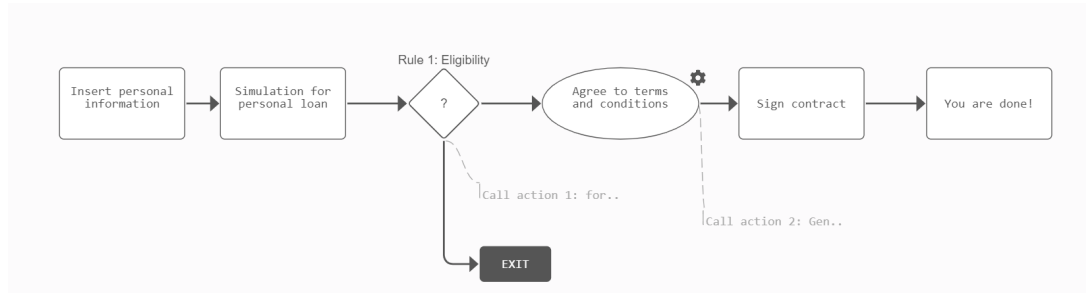
## Flow Map

This map shows the user the steps in a flow, the rules and transitions. The difference from the "[Digital Journey Map](#)" on page 210 is that this map only shows one flow, as opposed to the digital journey map that shows all flows in a journey.

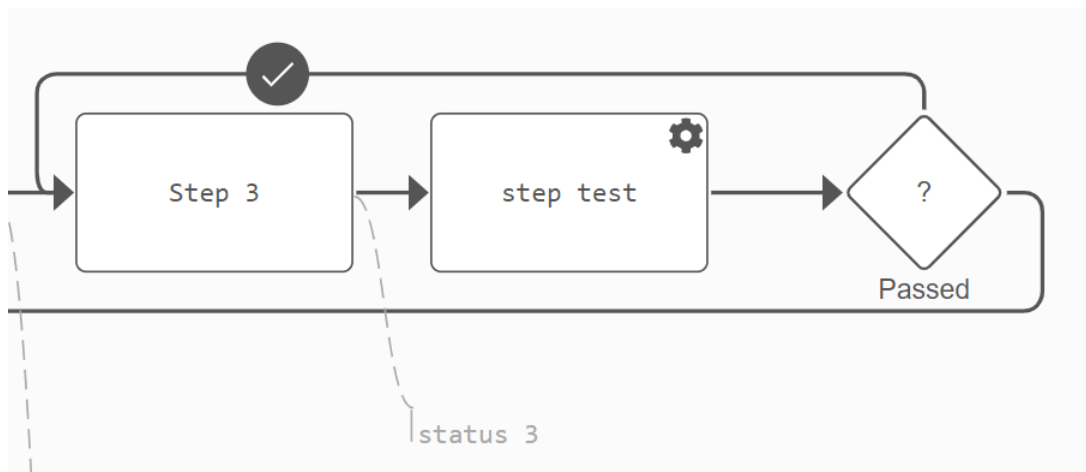
The Flow map exhibits the actions performed on a step configured in "[Flow Control](#)" on page 241. To do so:

1. From the main menu, select **Digital Experience > Customer Journeys > Form Driven Flows**.

2. Select the existing Form driven flow you wish to modify or create a new one.
3. After opening the flow, click on **"Flow map"**.



4. Click on a step to see the UI configurations or the rule for flow control. Configure to add a step or an action or a transition.



For example, step 3 is a default step, step test is a processor step, the passed step is a rule and the status 3 is an action performed after the step 3 data is saved and the action is triggered.

5. Click **Save and reload** to save the changes.

## Configure Field Options

Field options customize form fields behavior in order to build dynamic forms and user journeys. For instance, you can:

- Show tooltips with detailed descriptions for your form fields
- Make fields read-only to prevent modifications
- Mark mandatory fields as required
- Streamline the user experience with custom JavaScript code that executes when a field updates. For instance, you can show only relevant cities based on the selected country, automatically change an account currency based on the selected country, or show/hide CIF/CNP fields based on selected customer account type.

**NOTE**

Form field options are applied only at the user interface level, not in the data model. For instance, marking an attribute field as required in a form driven flow, does not make the attribute mandatory in the entity's data model.

**Customize Form Fields**

To customize form fields for specific attributes, in the configuration page of the form driven flow, select the **Field Options** tab and click **Insert** to customize the field for a specific attribute. Then, fill in the corresponding settings for the field:

Field	Description
Use Virtual Attribute	Tick when rendering fields for data extensions to make them available in the "Attribute" below field.
Attribute	At the right-side of the <b>Attribute</b> field, click the drop-down and select the attribute for which you wish to customize the form field. This field is mandatory.

Field	Description
Show Tooltip	<p><b>Inherit</b> (default value). Inherits the tooltip show options selected on the General tab. If no options have been previously selected, the default will be User Settings, which means that the users will be able to show tooltips by toggling on / off the <b>Tooltips</b> button displayed at the top-right corner of the UI.</p> <div> <p><b>NOTE</b> The <b>Tooltips</b> toggle button is available in the Digital Experience Portal only when tooltips are available on the journey.</p> <ul style="list-style-type: none"> <li>• <b>YES.</b> Always show tooltips in the Portal UI on forms and user journeys when users hover their mouse on the attributes on fields which have tooltips. The users do not have the possibility to toggle the tooltips off.</li> <li>• <b>No.</b> Never show tooltips in the Portal UI on the data form / digital journey. The users do not have the possibility to toggle the tooltips on.</li> </ul> </div>
Custom Tooltip	<p>The text which will be shown in the tooltip in the Portal Ui. The tooltip text is localizable.</p> <div> <p><b>NOTE</b> The maximum length of the tooltip text is 500 characters.</p> </div>
Lookup View Name (Lookup attributes)	Predefined view used to select values for the attribute. If no value is selected, the attribute's default view is used.
Lookup as DropDown (Lookup attributes)	Display the lookup attribute values in a drop down grid instead of a separate grid window.
Lookup as Select Box (Lookup attributes)	Display the lookup attribute values in a simplified list. The list doesn't have a header and contains only the values from the first column of the referenced entity's data view.

Field	Description
Show Edit Button (Lookup attributes)	Display the pencil button allowing you to edit the selected lookup attribute.
Lookup Edit Form (Lookup attributes)	Predefined form used to edit the selected lookup attribute. If no value is selected, the attribute's default edit form is used.
Lookup Show Insert Button (Lookup attributes)	If the lookup attribute is selected from a grid window, not from a drop down list, the grid will not include the Insert button (you will not be able to add new lookup attribute records).
Field is Read Only	The field will be non-editable.
Field Required Level	<p>The <b>Required Level</b> drop-down allows you to choose if a specific attribute (field) is to be mandatory, recommended or optional:</p> <ul style="list-style-type: none"> <li>• <b>None</b> – The field is optional. No error message will be displayed if the field is empty.</li> <li>• <b>Recommended</b> – A blue dot will be displayed on the upper-left corner of the field in the user interface to indicate that it might be useful to fill in the field.</li> <li>• <b>Required</b> – A red dot will be displayed on the upper-left corner of the field in the user interface to indicate that it is a mandatory field. The end user will not be able to add a new record if the field will be left blank.</li> </ul>
UI Template	<p>Allows you to select a placeholder for either email or phone. Select the template available for your attribute. After the selection is done, a HTML code will be displayed.</p> <div> <div> UI Template UI Template Options </div> <div> Phone Placeholder <pre> 1 return { 2   "type": "text", 3   "autocomplete": "on", 4   "placeholder": "", 5   "defaultCountry": "", 6   "onlyCountries": [], 7   "preferredCountries": [], 8   "excludeCountries": [], 9 } </pre> </div> </div>

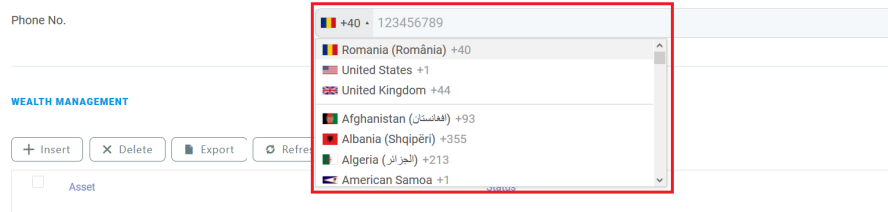
Field	Description
Order Option Set Items (Option Set attributes)	<ul style="list-style-type: none"> <li>• Default - Same as By Order Index.</li> <li>• By Order Index - Sorts option set items based on the order index set up in the option set.</li> <li>• Alphabetical - Sorts option set items alphabetically by display name. Sorting is applied after localization, ensuring correct order in any language.</li> </ul>
Open Calendar on Click (Date, DateTime, and Invariant Date attributes)	<ul style="list-style-type: none"> <li>• Yes - Clicking the input field opens a date picker (a calendar interface) that allows the user to select the date.</li> <li>• No - When clicking the input field, the user has to type the date manually. The date picker is still available via the calendar icon located to the right of the input field.</li> </ul>
Placeholder (Date, DateTime, and Invariant Date attributes)	Displays a temporary grayed out text in the input field when empty. Allows you to provide a hint or example of the expected input format or type of data (e.g. DD/MM/YYYY).
Validation Message Mode (Date, DateTime, and Invariant Date attributes)	<ul style="list-style-type: none"> <li>• Auto - Displays the red exclamation mark if the input is invalid. The corresponding error message expands only if the user clicks the icon.</li> <li>• Always - Automatically expands the error message when the input is invalid (e.g., "Value must be a date or time" or "Value is out of range").</li> </ul>

Field	Description
Use Mask Behavior (Date, DateTime, and Invariant Date attributes)	<p>This option is applicable only if "<a href="#">Open Calendar on Click (Date, DateTime, and Invariant Date attributes)</a>" on the previous page is disabled.</p> <p>Allows the user to enter a date or time using keyboard navigation:</p> <ul style="list-style-type: none"> <li>• <b>Left/Right arrow keys</b> select segments (day/month/year/hour/minute, etc.).</li> <li>• <b>Up/Down arrow keys</b> increment or decrement the selected segment.</li> <li>• <b>Direct typing</b> overrides the currently selected segment.</li> <li>• The mask auto-advances (e.g., entering two digits in the day segment moves the cursor to the month).</li> <li>• Invalid inputs are blocked or corrected automatically (e.g.: typing 19 in a month field replaces it with 09).</li> <li>• <b>Delete</b> or <b>Backspace</b> resets the current segment and moves to the next or previous one.</li> <li>• <b>Home</b> or <b>End</b> selects the first or last segment, respectively.</li> </ul>
Minimum Value (Numeric attributes)	The smallest number that can be entered in the field.

Field	Description
Minimum Value (Date or DateTime attributes)	<p>The earliest date allowed. Prior dates are disabled in the date picker and trigger an error if entered using the keyboard.</p> <ul style="list-style-type: none"> <li>• <b>Reference Date</b> - A static (exact date) or dynamic (now, first date of the current month, or last date of the current month) date that you can use as is or as a starting point for defining the minimum value.</li> <li>• <b>Select Exact Date</b> - If you picked an <i>exact date</i> for the reference date, specify the date.</li> <li>• <b>Plus or Minus</b> - Adds or subtracts a time interval from the reference date.</li> <li>• <b>Interval Unit</b> - Select if the interval you wish to add or subtract from the reference date is measured in days, months, or years.</li> <li>• <b>Interval Value</b> - Enter the number of days, months, or years you wish to add or subtract from the reference date.</li> </ul>
Maximum Value (Numeric attributes)	The largest number that can be entered in the field.

Field	Description
Maximum Value (Date or DateTime attributes)	<p>The latest date allowed. Later dates are disabled in the date picker and trigger an error if entered using the keyboard.</p> <ul style="list-style-type: none"> <li>• <b>Reference Date</b> - A static (exact date) or dynamic (now, first date of the current month, or last date of the current month) date that you can use as is or as a starting point for defining the maximum value.</li> <li>• <b>Select Exact Date</b> - If you picked an <i>exact date</i> for the reference date, specify the date.</li> <li>• <b>Plus or Minus</b> - Adds or subtracts a time interval from the reference date.</li> <li>• <b>Interval Unit</b> - Select if the interval you wish to add or subtract from the reference date is measured in days, months, or years.</li> <li>• <b>Interval Value</b> - Enter the number of days, months, or years you wish to add or subtract from the reference date.</li> </ul>
File Open (File attributes)	<p>Allows you to set the option to either preview or download the files on your computer. If you select the Open&amp;Download option, click on the document to choose between opening the file or downloading it on your computer.</p> <ul style="list-style-type: none"> <li>• Open</li> <li>• Open&amp;Download</li> <li>• Download</li> </ul> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p><b>NOTE</b></p> <p>The supported file formats for preview (open) are: .jpg, .jpeg, .pdf.</p> </div>

Field	Description
UI Template Options	<p>This field is shown only if the UI Template is selected. You can customize the selected placeholder by modifying the code based on your preferences.</p> <p>For the <b>Phone Placeholder</b> UI templates, the following options are available:</p> <ul style="list-style-type: none"> <li>• type – "tel"</li> <li>• autocomplete – true/false. Selects the use of the browser's autocomplete feature.</li> <li>• placeholder – Displays a grayed out value for exemplification purposes when the field is empty.</li> <li>• default_country – Phone number format based on the <a href="#">ISO 3166-1 alpha-2</a> country codes used by default.</li> <li>• onlyCountries – List of allowed phone number formats based on the <a href="#">ISO 3166-1 alpha-2</a> country codes. When empty, all country-specific phone number formats are available.</li> <li>• countryOrder – Phone number formats based on the <a href="#">ISO 3166-1 alpha-2</a> country codes displayed at the top of the list in the specified order.</li> <li>• excludeCountries – Phone number formats based on the <a href="#">ISO 3166-1 alpha-2</a> country codes excluded from the list.</li> <li>• validationNumberType - Indicates if the UI template allows landline and/or mobile phone numbers. Available options are: "FIXED_LINE", "MOBILE", and "FIXED_LINE_OR_MOBILE".</li> </ul> <p>For instance, the template options below</p> <pre>return {   type: "tel",   autocomplete: true,   placeholder: "123456789",</pre>

Field	Description
	<pre> default_country : "ro", onlyCountries: [], countryOrder: ['ro', 'us', 'gb'], excludeCountries: [], validationNumberType: ['FIXED_LINE_OR_MOBILE'] } </pre> <p>will create the following field:</p> 
Attribute Change Event	Code to be executed each time a form field is updated. For instance, you can dynamically populate a lookup field based on the value selected in another field. For more information, see <a href="#">"Configure Field Rules" below</a> .

### Configure Field Rules

To create field rules, on the entity linked to your form driven flow, you should have at least two attributes defined, one for the condition and the second for the action. For example, if you want to display relevant cities based on the selected country, on the **accountInsertForm** digital journey of the account entity, you have to add the country field and the city field on which you define the action (where both fields are lookup fields). If the data form does not have the two fields, add them.

#### 1 Add field for action

Configure the field used as filtering criterion (eg. country) as described in the ["Customize Form Fields" on page 249](#) section.

#### 2 Add field for condition

Add the field which is filtered (e.g., city) and configure it as described in the ["Customize Form Fields" on page 249](#) section.

Then, in the Attribute Change Event field, provide the piece of code to be executed each time the lookup value of the country field changes, for instance:

```

var accountCountryId = ebs.getFormAttributeValue
("ebsContainerContent", "Country")
ebs.getByQuery({
  entity: {
    alias: "a",
    name: "accountCountryId",
    attributelist: [
      {
        name: "Cities",
      }
    ]
  },
  "where": {
    "type": "and",
    conditionlist: [
      {
        first: "a.accountCountryId",
        type: "equals",
        second: "val("+ accountCountryId +")"
      }
    ]
  },
function(e){
  if(e.Records != null && e.Records != undefined &&
e.Records.length > 0)
  {
    ebs.setFormAttributeValue("ebsContainerContent",
e.Records[0].a_accountCountryId);
  }
})

```

Save the field, then save the form driven flow.

In the user interface, each time you select a country, the list of cities will include only cities that belong to that country (i.e., the cities relevant for the country).

## Form Actions



Form actions allow you to:

- change a record's business status
- generate a report
- run a server side script

- run a business decision matrix
- call a formula.

Once defined, you can use form actions that can be triggered on-demand, for example by adding a form action to a form button. For more details on how to add the form actions to the buttons, check the ["Buttons" on page 347](#) section.

### Create a form action

1. On the configuration page of the form, click the **Actions** tab. Two grids will be shown: one is **Form Actions** and the other is **Action Group**.
2. At the top of the **Form Actions** section, click **Insert**.
3. In the **Form Action** window:
  - a. Enter a name for the form action.
  - b. Click the Plus (  ) sign next to the execute label to add a command.
  - c. Click the labels in the command to select the desired operands such as change business status from status/ generate digital document/ call custom processor/ call business matrix/ call a formula with mapping. Then, select the status/ the document/ the processor/ the matrix.
  - d. Go back to step b. if you wish to add additional commands. To remove commands from the list, click  .
  - e. Click **Save** .

Form Action

Hire

execute

Change Business Status from Status

candidate -> hired

Call Business Matrix

Select...

Employee Score

Save

#### Available form action commands

Command	Description	Operands	Examples
Change Business Status from Status	Changes the record's workflow status based on the entity's attached business workflow. For details, see the <a href="#">Business Workflows Processor documentation</a> .	<ul style="list-style-type: none"> <li>Initial status</li> <li>Final status</li> </ul>	Change the status from Draft to Active
Generate Digital Document	Generates a predefined report. For details, see <a href="#">"Analytics" on page 1015</a> .	Report name.	Generate a contract or an agreement
Call Custom Processor	Runs a predefined on-demand server automation script. For details, see <a href="#">"Create On-demand Server Automation Scripts" on page 1168</a> .	Server automation script name.	Call the E-sign processor or an endpoint.
Call Business Matrix	Runs a predefined business decision matrix. For details, see the <a href="#">Business Decisions Processor documentation</a> .	Business decision matrix name.	Call the eligibility matrix.
Call Formula with data mapping	Runs a predefined formula for calculation of input data. See <a href="#">"Use Business Formulas in a Digital Journey" on page 521</a> .	Business Formulas	Call the formula for calculating the policy of an insurance.

Alternatively, it is possible to trigger an action from the actual step of a digital journey.

### Add an action to a specific step in a Form Driven Flow

1. From the main menu, select **Digital Experience > Customer Journeys > Form Driven Flows**.
2. Select the flow you wish to work on. Click the **Steps** tab. Select the step you wish to modify.
3. Inside the step, select the **Flow Control** tab.

4. In the **Actions to be performed** section, select the action you wish to have in that particular step. The available actions are those configured earlier from the **Available form action commands** table.
5. Click **Save and Reload**.
6. Optionally, if you have selected the Generate contract action, go to the **UI** tab and add the button for this action and the file attribute created in the Data Model of the entity. For more information, see the [Digital Documents Processor](#) documentation.

7. Click **Save and Reload**.

### Attach an endpoint in Form Action

In Form Driven Flow, in form Actions, a user can map between an automation script input/output parameters and form attributes.

1. Create one Platform Data Entity that is default for a FDF.
2. Create a server script with Input Parameters and the Output structure is one of the following, depending on your needs:
  - [none]
  - "Entity"
  - "Custom"
  - "Boolean".

If the endpoint has no output structure type, then no output mapping form will be displayed. For more details, see ["Create On-demand Server Automation Scripts" on page 1168](#).

3. Navigate to the Form Driven Flow, **Actions** tab and in the first grid click **Insert**.
4. Insert a name for the action.
5. Click the execute button and select **Call custom processor**.
6. Select for the drop-down the endpoint created.
7. Map the input attributes.

## 8. Map the output attributes.

Form Action

---

Action bool endpoint

execute +

× Call Custom Processor :

Custom Processor BD\_DPA\_13086\_04\_endpoint ▼

MAP INPUT PARAMETERS

input\_p1\_bool\_bool Is In Black List ▼

MAP OUTPUT STRUCTURE

Result type: Boolean

output Is In Black List ▼

Apply

9. Click **Apply**, then **Save**.

10. Click **Save and close**.

## Defining Action Groups

FintechOS Studio allows you to create a custom group of actions that can be triggered on demand on data form driven flows, when a button is clicked.

This section walks you through the steps you need to follow to create an action group.

### Prerequisite

- You need to have an on-demand automation script defined on the entity for which you create the form driven flow.

### 1 Add action group

On the configuration page of the form driven flow, click the **Action Groups** tab.

At the top of the Action Groups section, click **Insert** . The Add Action Group page appears. Provide the following properties:

Property	Description
Name	The name of the action group. Make sure that you use the following naming convention: pascal case, no special characters and no blank spaces.
Display Name	The name of the action group that will displayed in the Digital Experience Portal.
Entity Form	Select the data form for which you define the action group.

Click **Save and reload** . The Add Action Group page is replaced by the Edit Action Group page and the Actions section becomes available.

## 2 Add endpoints

Add the endpoints on which actions defined in the selected on-demand automation scripts will be run on button click. To do so, from the Endpoints section, click the **Insert** button. The Add Endpoint page will be displayed. Provide the following action properties:

Property	Description
Name	The name of the endpoint that is used by the system
Display Name	The name of the endpoint that will be displayed in the Portal UI.
Script	From the <b>Script</b> drop-down, select the on-demand automation script which will be run on button click. For more information on automation scripts, see Server Scripts. If you have no on demand scripts defined, you can add one by clicking the Script drop-down, clicking the Insert button in the page listing the available scripts and in the Add Scripts page providing the script properties.
Executes Save	Saves after the script execution.

At the top-right corner of the page, click **Save and close**.

**Edit Endpoint**

Name: Endpoint1

Display Name: Endpoint1

Action Group: Test

Script: af\_context.result=true

☐ Executes Save

Security Roles

☐ Name

testrole

+ Insert existing X Remove existing

You can add as many endpoints as you need. The endpoints are displayed in order of their index. To change their order index, drag and drop the desired action row.

The figure below shows the actions defined in the Action Groups for the credit bureau interrogation in the user interface.

**EDIT CREDIT APPLICATION - FINANCIAL INFO AND SCORING**

[Credit Bureau Interrogation](#)

[Get Credit Bureau Result](#)

Credit Bureau Result: [none]

Total Owed

Oldest Info Date

Expenses

FICO Score

CB Phone Number

CB Number Of Interrogations

CB Last Interrogation Date

CB District

CB City

CB Address

[Financial obligations](#)

### Hide the action button

To hide the action button, click the Advanced tab, then click the After Events tab and in the field type the following JavaScript code:

```
$("#div[data-action-group-name=" <the name of the action group you want to hide>"]).hide();
```

At the top-right corner of the page, click one of the save icons to save the changes. The action button will be hidden.

## Defining Filtered Fields

The **Filtered Fields** tab allows you to dynamically filter the options available in a lookup attribute field based on another field. For instance, you can limit the cities available in a drop-down based on a previously selected country. Thus, you can ensure data accuracy and streamline the user experience when the lookup entities you are referencing are also referencing each other.

### Set up a Filtered Field in a Data Form, Form Driven Flow, or App Data Form

For instance, you may need to create a user registration form to collect each of your users' name, country, and city. The available cities and countries are stored in separate entities, and the city entity has a lookup to the country entity to reference which country the city belongs to. To create your registration form:

1. Create an entity that stores all country names (for details, see ["Business Entities" on page 58](#)). The name of this entity is **countries**.
2. Create an entity that stores all city names. The name of the entity is **cities**. This entity should also include a lookup attribute called **country** which references the countries entity (for details, see ["Attributes" on page 71](#)).
3. Create your registration form source entity that includes an attribute for the user name and two lookup attributes called **city** and **country**, referencing the cities and countries entities respectively.

At this point, you should have the following entities:

Entity	Attribute	Type	Description
countries	name	text	Name of the country
cities	name	text	Name of the city
	country	lookup	Reference to the countries entity
regForm	name	text	Name of the user
	city	lookup	References the cities entity
	country	lookup	References the countries entity.

4. When creating "Data Forms" on page 142, "Form Driven Flows" on page 221, or "App Data Forms" on page 933, use the **Filtered Fields** tab to insert a filter for the city attribute based on its country.

ENTITY regForm FORM default

Save and close Save and reload Open in code

1 General 2 UI 3 Steps 4 Field Options 5 Filtered Fields 6 Advanced 7 Security Roles

Filter Fields

+ Insert X Delete Export Refresh

Attribute To Filter	Attribute To Filter By
<input checked="" type="checkbox"/>	
<input type="text" value="city"/>	<input type="text" value="country"/>

The filtered field should have the following settings:

- a. Attribute To Filter - **city**, since we are filtering the values that will be visible in the city lookup attribute of the registration form entity.
- b. Attribute To Filter Reference - **country**, since the lookup attribute in the cities entity used to reference the countries entity is called country.
- c. Attribute To Filter By - **country**, since we are filtering cities based on the value selected in the country attribute of the registration form entity.

Save and close Save and reload

Edit Filtered Field

Filtered Field

Attribute To Filter  
(the name of the attribute on this form that you want to filter) - child

city

Attribute To Filter Reference  
(the name of the attribute on child entity that references the parent)

country

Attribute To Filter By  
(the name of the attribute on this form that you want to filter by) - parent

country

Based on the records you have populated in the countries and cities entities, the city attribute field will be filtered based on the country.

The screenshot shows a form titled "ADD REGISTRATION FORM" with a sub-header "REGISTRATION FORM". It contains three fields: "Name" (a text input), "Country" (a dropdown menu with "Romania" selected), and "City" (a dropdown menu with "Select a value..." displayed). The "City" dropdown is open, showing a list of cities: "Bucuresti", "Cluj", and "Iasi".

The screenshot shows the same "ADD REGISTRATION FORM" but with the "Country" dropdown set to "United States". The "City" dropdown is open, showing a list of cities: "Chicago", "Los Angeles", and "New York".

### Filtered Fields on Editable Views

The filtered fields also apply on editable views. If you set up inline editing for a view, the filtered fields settings that you find on the data form will also apply on the view.

In the Attribute To Filter Reference field, you can also specify the alias when you use a custom view for the lookup that will be filtered and this field is not on the main entity (the entity of the lookup).

When using a custom view for the lookup field that will be filtered and this field is not on the main entity (the entity of the lookup), in the Attribute To Filter Reference field, you can also provide the alias.

The screenshot shows a dialog box titled "Add Filtered Field". Inside, there are three input fields with labels and descriptions:

- Attribute To Filter** (the name of the attribute on this form that you want to filter) - child: Field1Id
- Attribute To Filter Reference** (the name of the attribute on child entity that references the parent): f2.Field3Id
- Attribute To Filter By** (the name of the attribute on this form that you want to filter by) - parent: Field3Id

In the example above, **Attribute to filter reference** is "f2.Field3Id" and "f2" is the alias from the entity Field3 which is related to Field1 which is the entity from the lookup.

## Header Items

When creating user journeys, the **Header Items** tab allows you to provide users visibility to the most relevant information (attributes). It also allows you to make the form driven flow's header sticky on scroll, which is useful when a data form has many attributes and users have to scroll-down to complete it.

### Add a new header item

1. Click **Insert**. The Add New Header Item page will be displayed.
2. Fill-in the following fields:

Field	Description
Label	The name of the header item which will be displayed on the form driven flow or app data form.
Use Virtual Attribute	Select if you wish to display a virtual attribute from the entity's extended data model, instead of a regular attribute. This will replace the Attribute field with a drop-down called Virtual Attribute which will allow you to select virtual attributes instead of regular attributes.

Field	Description
Attribute/Virtual Attribute	The value of the entity attribute which will be displayed on the header item. If you selected the Use Virtual Attribute checkbox, you will be able to choose virtual attributes from the entity's extended data model.
Is Primary Attribute	Tick only if the attribute to be displayed in the digital journey or app data form header is also the entity's primary attribute.

The screenshot shows a dialog box titled 'Add Entity Form Header Item'. It contains four fields: 'Label' with the value 'Last Name', 'Use Virtual Attribute' which is unchecked, 'Attribute' with the value 'LastName', and 'IsPrimaryAttribute' which is unchecked. There are also small icons for adding and deleting items next to the Attribute field.

3. Click **Save and close**. The page refreshes and the header item is displayed in the Entity Form Header Items list.

By default, the header item has the order index set to 1. Add as many header items as you need, then from the **Entity Form Header Items** list set their display order on the data form, by drag and dropping rows. The header items order is ascendant, whereas the first row (header item) has the order index set to 1.

The data form header is by default sticky on scroll. If you do not want to have a sticky data form header, tick clear the **Sticky Header Items** checkbox.

The screenshot shows a table titled 'Entity Form Header Items'. At the top right are buttons for '+ Insert', 'X Delete', 'Export', and 'Refresh'. The table has four columns: 'Label', 'Attribute', 'Virtual Attribute', and 'Order Index'. The first row contains 'Last Name' in the Label and Attribute columns, and '1' in the Order Index column. Below the table, there is a checkbox labeled 'Sticky Header Items (default true)' which is checked.

## Link Labels to Attributes (deprecated)

### IMPORTANT!

This method of linking UI elements to entity attributes ensures backward

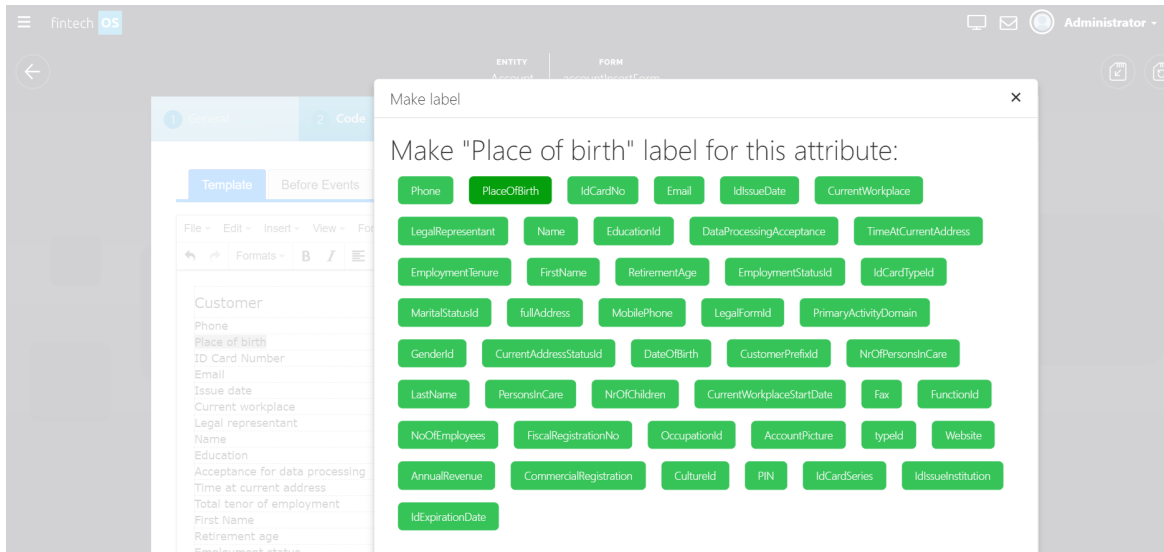
compatibility with legacy forms and steps created prior to v.24. Even on legacy UIs, it is recommended to use the ["UI Designer" on page 323](#) to create attribute fields if possible.

When creating HTML templates (**UI** tab) you can link labels (elements) to attributes. You can do that by using either the HTML editor or the Source code.

### Link labels to attributes using the HTML editor

To link a label to an attribute using the HTML editor, in the template, right-click on the text which will be label and select Make Label. A pop-up will be displayed, listing the attributes you can choose from.

The figure below presents how the pop-up might look like:



Select the desired attribute. The pop-up closes and the selected text will be label for the selected attribute.

To remove the link, right-click on the text (label) and select Remove Label.

### Link labels to attributes using the Source code

To link a label to an attribute control using the source code, from the Tools menu, click **Source code**. In the Source code page, add the HTML attribute data-label-for to that label, set its value to the name of the attribute and click **OK**.

Source code for linking the label "My Atribute" to the attribute "myAttribute"

```
<div class="row">
  <div class="col-xs-12 data form-label" data-label-
for="myAttribute">My Atribute</div>
  <div class="col-xs-12">{myAttribute}</div>
</div>
```

To remove the link, delete from the source code the `data-label-for` HTML attribute.

## Display View from Another Entity

Let's say you want to show a view from one entity data form on another. For example, you would like to see on a data form specific details available in a view on a different entity, saving the extra clicks of having to switch from a data form to another one to check those details.

Starting with FintechOS Platform 18.1.10, you can achieve this by using a token on the data form where you want to render the view from the other entity.

### Prerequisites

- You have two entities.
- One entity has a view defined (can be the default view) and the other entity has a digital journey or a data form defined.

### Display a view from another entity

1. On the entity where you want to render the view from another entity, go to the edit configuration page of the digital journey or data form. Click the **UI** tab.
2. On the left panel of the UI Designer select the **Relation Data Template** and drag it to the container where you wish to place it.
3. After dropping it, click on it to open the right panel and configure it. Select the relation from the drop-down list.
4. Replace the relation tag with the token which will render the view from another entity. The token should follow this format: **{? EntityName, view: viewName ?}**.

**NOTE**

If you copy/paste the token, the HTML editor might break its formatting and it might not work as intended. We recommend you to either copy/paste the token using a text editor or within the HTML Editor from the Tools menu, select <>Source code and check the token formatting.

The table below describes the token elements you can use when rendering the view.

Token Element	Description
entityName	The name of the entity from which you will render the view.
view:	The name of the view to be displayed.
noheader	Does not display the view header. Display the default view from entity FTOS_CMB_AccountType without header using token <b>{? FTOS_CMB_AccountType, view: default, noheader ?}</b>
nofilter	Does not display the view filtering / search. Display the default view from entity FTOS_CMB_AccountType without filtering using token <b>{? FTOS_CMB_AccountType, view: default, nofilter ?}</b>
noinset	Does not display the Insert button on the view toolbar. Display the default view from entity FTOS_CMB_AccountType without the Insert button using token <b>{? FTOS_CMB_AccountType, view: default, noinsert ?}</b>
nodelete	Does not display the Delete button on the view toolbar. Display the default view from entity FTOS_CMB_AccountType without the Delete button using token <b>{? FTOS_CMB_AccountType, view: default, nodelete ?}</b>

Token Element	Description
noexport	<p>Does not display the Export button on the view toolbar.</p> <p>Display the default view from entity FTOS_CMB_AccountType without the Export button using token <b>{? FTOS_CMB_AccountType, view: default, noexport ?}</b></p>
norefresh	<p>Does not display the Refresh button on the view toolbar.</p> <p>Display the default view from entity FTOS_CMB_AccountType without the Refresh button using token <b>{? FTOS_CMB_AccountType, view: default, norefresh ?}</b></p>
notoolbar	<p>Does not display the view toolbar.</p> <p>Display the default view from entity FTOS_CMB_AccountType without the toolbar using token <b>{? FTOS_CMB_AccountType, view: default, notoolbar ?}</b></p>
collapse:	<p>The text displayed as a collapse panel.</p> <p>Display the default view from entity FTOS_CMB_AccountType after clicking on the Name text using token <b>{? FTOS_CMB_AccountType, view: default, collapse: Name ?}</b>.</p>
data form:	<p>If the entity from which you render the view has multiple forms and you want a specific data form on edit directly from this view, use <b>form:</b> in the token followed by the data form name.</p> <div> <p><b>NOTE</b> Do not use no insert within the token when using <b>form:</b>.</p> </div>

Token Element	Description
insertForm:	<p>If the entity from which you render the view has multiple forms and you want a specific data form on insert directly from this view, use <b>insertForm:</b> in the token followed by the data form name.</p> <div> <p><b>NOTE</b> Do not use no insert within the token when using <b>data form:</b>.</p> </div>

- After you finish customizing the view layout, save the data form and the entity.

### Filter the view results

You can filter the results displayed in the view by using a custom fetch for the view.

### Pass default value

If you want to pass a default value for the insert data form, you must use the `context.on("goToInsert", function(e){})` function;

```
context.on("goToInsert", function(e){
  var pId = "18352c17-0ca4-4b6d-8037-28510e6186d1"
  e.options.defaultVals = "parentId*" + pId;
})
```

### Refresh the view

If you want to programmatically refresh the view, use the `ebs.refreshGrid(gridName)` function.

If no **view:** is specified, `gridName` will be the `entityName` token element.

```
For {? myEntity ?}
ebs.refreshGrid("myEntity")
```

If **view:** is specified, `gridName` will be the `entityName` token element concatenated with `_` and the **view:** token element.

```
For {? myEntity, view: myEntityView ?}  
ebs.refreshGrid("myEntity_myEntityView")
```

## Render Custom Data Extensions

Adding custom data extensions to forms and user journeys is a powerful and flexible technique which allows you to customize the user interface.

### Prerequisites

- You have extended the entity with data extensions. For more information, see ["Extend the Data Model" on page 1](#).
- In the Data Model section of the form driven flow, add the entity extension which contains the data extensions that you want to render.

To render a data extension on a form driven flow, in FintechOS Studio, follow these steps:

1. Go to the configuration page of the form driven flow on which you want to render the data extension. The configuration page appears by default on the **General** tab.
2. Click the **Advanced** tab, then the **After Events** tab and provide the code to get the values of the attributes existing in the DB and set the value of the data extension.
3. Click the **Field Options** tab and add the attributes whose values will be used by the data extension. Make sure to tick the **Use Virtual Attribute** checkbox when adding the attributes in the list of field options.
4. In the UI template of the form driven flow (journey configuration page > UI tab or in the UI template of the step where you want to render the data extension (data form configuration page > Steps tab > section configuration page > UI tab) add the data extension similar to normal entity attributes by using tokens.

The values of data extensions flow with the save data request, and their values are available for processing in server side scripts inside the `context.AdditionalAttributes.VirtualAttributes` array.

## Example

This section teaches you how to automatically calculate Total Expenses as the sum of Unreported Expenses and Expenses, and display the amount in the Exposure section of a loan application.

## Prerequisites

- The **FTOS\_BAPer\_LoanApplication** entity has the attributes **unreportedExpenses** and **expenses**.
- The entity **FTOS\_BAPer\_LoanApplication** has been extended with the data extension **TotalExpenses** relating to attributes **unreportedExpenses** and **expenses**.
- On the **LoanApp\_ConsumerFlow** journey, in the **Data Model** section, linked to the entity extension is added.

To use the Total Expenses data extension on the **LoanApp\_ConsumerFlow** journey, follow these steps:

1. Go to the configuration page of the **LoanApp\_ConsumerFlow** journey.
2. Click the **Advanced** tab, then the **After Events** tab. Provide the code for getting the values of the **UnreportedExpenses** and **Expenses** attributes and setting the **TotalExpenses** data extension so that it returns the sum of the two attributes.

```
var unreportedExpenses = ebs.getFormAttributeValue
('ebsContainerContent', 'UnreportedExpenses');
var expenses = ebs.getFormAttributeValue
('ebsContainerContent', 'Expenses');

ebs.setFormAttributeValue('ebsContainerContent',
'TotalExpenses', unreportedExpenses + expenses);
```

3. Click the **Field Options** tab and to the list of **Entity Form Fields**, add the two attributes: **unreportedExpenses** and **expenses**. When adding each of the attributes, in the **Attribute Change Event** field, provide the following code:

```
var unreportedExpenses = ebs.getFormAttributeValue
('ebsContainerContent', 'UnreportedExpenses');
var expenses = ebs.getFormAttributeValue
('ebsContainerContent', 'Expenses');

ebs.setFormAttributeValue('ebsContainerContent',
'TotalExpenses', unreportedExpenses + expenses);
```

- Click the **Steps** tab, and in the sections list, double-click the Exposure section (record) and in the **UI** tab, add the Total Expenses data extension by using the token: `{TotalExpenses}`.

In the Portal, when updating any of the fields Unreported Expenses or Expenses, the Total Expenses field will be automatically filled-in (recalculated).

## Create Custom Search Forms

To create a custom search data form based on input fields, search button and list results, follow these steps:

- On the UI template (UI tab) of the journey / journey step, add a data form container and a button element to the HTML source code:

```
<div id="searchAccount" style="display: inline-block; width: 80%; "></div>
```

```
<div id="BtnId" style="cursor: pointer; color: white; border:
2px solid rgba(0, 0, 0, 0);
background: #25a4dc; text-align: center; vertical-align:
middle; padding: 3px; width: 125px;
height: 100%; display: inline-block; margin-bottom: 30px;
margin-left: 15px;">Search</div>
```

2. On the form driven flow, click the **Advanced** tab, then click the **After Events** section and provide the code to generate the search.

```
/Generate search data form based on a custom data form
(AccountSearch) with only one input field
(Name):
$("#searchAccount").html("<div id='searchAccountForm'
display: inline-block;'></div>");
var formData = {
    entityName: "account",
    formName: "AccountSearch",
    mode: "insert",
    name: "searchAccountForm",
    mainHtmlId: "searchAccountForm",
    disableControlNavigation: true,
    noRenderIntent: true
}
ebs.generateForm(formData, function(e) {
});
//Generate grid
var fetchResults = new Object();
fetchResults.entity = new Object();
fetchResults.entity.name = "TestEntity";
fetchResults.entity.alias = "base";
var viewResults = new Object();
viewResults.fetch = fetchResults;
viewResults.viewName = "default";
var afterGenerateCallback = function (e) {
    var gridId = "AccountResult";
    var dataGrid = $("#" + gridId).dxDataGrid("instance");
    dataGrid.option("paging", {pageSize: 10});
    dataGrid.option("onSelectionChanged", function (g) {
        if (g.selectedRowsData != null &&
g.selectedRowsData.length > 0) {
            $.each(g.selectedRowsData, function(index,value)
{
                currentAccountName = value.base_Name;
```

```

        $("#ClientsSelected").text
(currentAccountName);
    });
}
});
};
ebs.generateGrid("AccountResult", viewResults, "auto", {},
afterGenerateCallback, {
selectionMode: "single" });
//search function and grid refresh as search result
var onClickSearch = function() {
    var where = {
        type: "and",
        conditionlist: []
    };
    var productName = ebs.getFormAttributeValue
("searchAccountForm", "Name");
    if (productName !== null && productName !== "") {
        where.conditionlist.push({
            first: "base.Name",
            type: "contains",
            second: "val(" + productName + ")"
        });
    }
    if (where.conditionlist.length == 0) where = null;
    var fetch = $("#body").data("AccountResultmyData_
dxquery").fetch;
    fetch.where =
where;
//new updated fetch
    $("#AccountResult").dxDataGrid("instance").option
("dataSource", ebs.getDataStore(ebs.
getEbsDataUrl(), fetch));
    $("#AccountResult").dxDataGrid("instance").refresh();
}
$("#BtnId").click(onClickSearch);

```

## Form Driven Mock-up Flows

Form driven mock-up flows allow you to design a form driven flow without having an underlying data model. This lets consultants & developers quickly define the general layout of the user interface. Developers can then attach a data model to the mock-up,

map entity attributes to the corresponding form fields, and work on any additional back-end configurations.

## Create a form driven mock-up flow

1. From the main menu, select **Digital Experience > Customer Journeys > Form Driven Flows**.
2. In the Form Driven Flow screen, click **Create Mock-up**.

Form Driven Flow

+ Create

+ Create Mock-up

Delete

<input type="checkbox"/>	Name	Description	Entity Name	Entity Display Name	Digital Asset
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	afflow1		afent1	afent1	afjda
	afflow4		afent4	afent4	afjda
	AS_Ent2FDF		AS_Ent	AS_Ent	
	AS_EntFDF		AS_Ent	AS_Ent	
	BD_entity_1111_FDF		BD_entity_1111	BD_entity_1111	BD_DA_001
	BD_entity_2222_FDF		BD_entity_2222	BD_entity_2222	BD_DA_001
	BD_entity_333_FDF		BD_entity_333	BD_entity_333	BD_DA_333
	BD_entity_444_FDF		BD_entity_444	BD_entity_444	BD_DA_333
	BD_entity_FDF_111		BD_entity_111	BD_entity_111	BD_digital_asset_111
	BD_FDF_001		BD_entity_001	BD_entity_001	BD_digital_journey_001

5

10

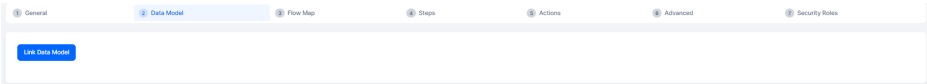
20

Page 1 of 2 (16 items)

12

1. Follow the instructions in the ["Form Driven Flows"](#) on [page 221](#) chapter to create a form driven flow mock-up. The following exceptions apply:

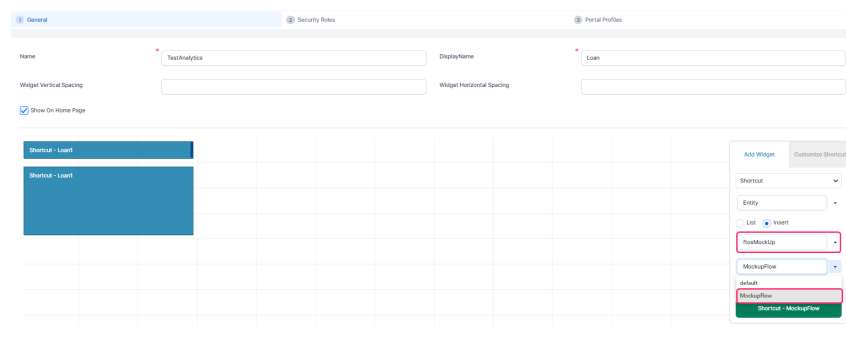
- The Data Model tab contains only a button called **Link Data Model**. This button is used to attach the flow to a business entity, once you are ready to turn the mock-up into a regular flow.



- The UI, Field Options, and Filtered Fields tabs are not available in form driven flow mock-ups.
- The ["UI Designer"](#) on [page 323](#) allows you to add only mock-up attribute templates (see ["Add attributes to the UI"](#) on [page 327](#)).

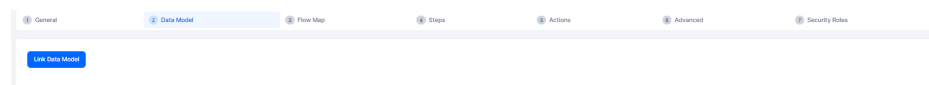
# Display a form driven mock-up flow

- To display a form driven mock-up flow in a digital experience portal, follow the instructions in the ["Digital Experience Portals" on page 402](#) chapter.
- Form driven mock-up flows are always attached to the **ftosMockUp** entity. In other respects, they behave like regular form driven flows.



## Convert a form driven mock-up flow into a regular form driven flow

1. Create the corresponding business entity your flow will attach to. For details, see ["Evolutive Data Core" on page 48](#).
2. Open the form driven mock-up flow and, in the Data Model tab, click **Link Data Model**.



3. Select the business entity you created at [step 1](#).

4. Click **Apply**.

The screenshot shows the 'Data Model' tab in the FintechOS Studio interface. At the top, there is a navigation bar with tabs: 1 General, 2 Data Model (active), 3 Flow Map, 4 Steps, 5 Actions, 6 Advanced, and 7 Security Roles. Below the navigation bar, there is a form configuration area. On the left, the label 'Entity' is followed by a dropdown menu showing 'TestEntity\_AT'. Below this, there is a blue button labeled 'Link Data Model'. On the right side of the form configuration area, there is a green button labeled 'Apply'.

5. You now have a regular form driven flow. In the "UI Designer" on [page 323](#), you can now replace the dummy fields with actual attributes from your business entity.

## Custom Flows

A custom flow is an ordered collection of components which address a singular need in the direction of componentization. It represents a business flow that can be the base for a digital journey, but it is not associated with an entity. It is an implementation of a business sub-process that addresses a single business need of the process.

FintechOS Studio enables you to create custom flows and use them in the following the following business cases:

- to create custom URLs which redirect the user to a specific data form or view.
- to generate custom filtered views based on security roles.
- to add buttons which trigger specifics actions.

## Create a Custom Flow

### 1 Provide custom flow general information

From the menu, click **Digital Experience > Digital Journeys > Custom Flows**. The Custom Flows page appears.

Click **Create**. The custom flow configuration is displayed which consists of three sections (tabs). The configuration page is displayed by default on the **General** tab.

1 General 2 Code 3 Security Roles

Name FTOS\_DFP\_ESign

Display Name ESign Processor

External Url

Description ESign Processor

Fill-in the fields based on your needs:

- **Name** - The name of the custom flow to be used by the system. The field is mandatory.
- **Display Name** - The name of the custom flow that will be displayed in the Digital Experience Portal. The field is mandatory.
- **External Url** - If the custom flow redirects to an external Url, provide the URL here.
- **Description** - Description of the custom flow.

Click **Save and reload** and new tabs will be available.

## 2 Design the custom flow layout

Click the **Code** tab. The Code section is displayed on the Template tab. In the Template section, provide the HTML code which defines the layout of the custom flow.

1 General 2 Code 3 Security Roles

Template After Generate Js

History

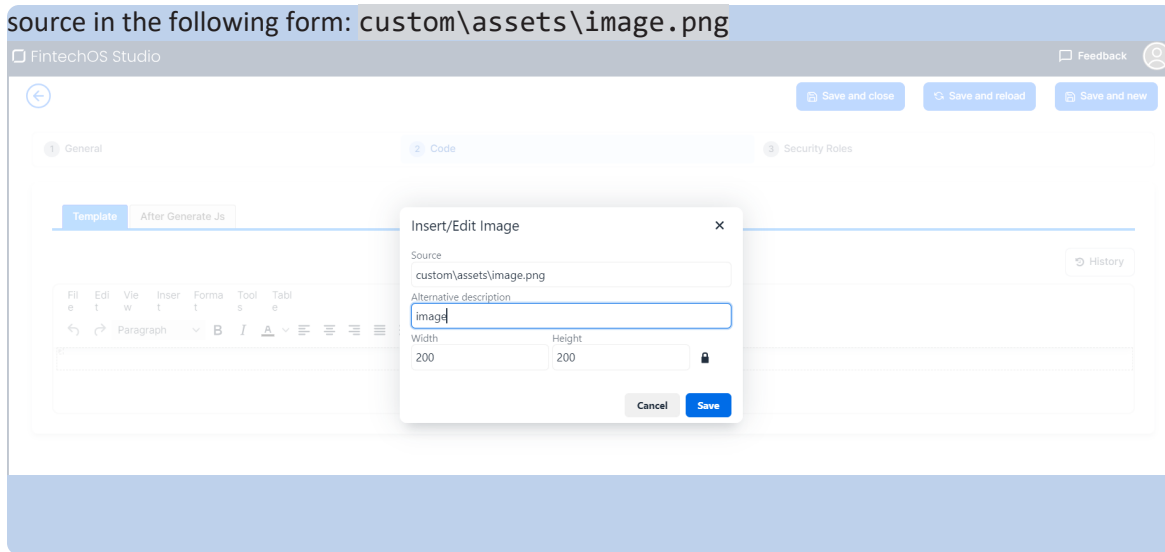
File - Edit - Insert - View - Format - Table - Tools -

Formats - B I A - [Rich Text Editor Icons]

Te rugam sa astepti

### NOTE

When inserting an image from blob storage, make sure to add the name in the



You can also create the HTML template of a form driven flow by using the [Code Editor](#) or the [UI Designer](#).

Using custom flows you can also create custom controls. For more information, see [Creating Custom Controls](#).

### 3 Define the custom flow

In the **Code** section, click the **After Generate Js** tab and provide the JavaScript code which defines the digital journey (redirect users to a specific data form or view, generate custom filtered views, add buttons which trigger specific actions).

Generate contract to be signed off by the customer and add button for the customer to sign off the document.

```
var params = sessionStorage.getItem("sessionParameters");
params = JSON.parse(params);
params.fileIsBase64 = false;
var signContract = function()
{
    if(params.existingFile){
        ebs.showLoadingPanel();
        if(params.externalURL == true){
            params.settings.maskNextStepURLSuccess =
params.settings.maskNextStepURLSuccess;
        } else {
            params.settings.maskNextStepURLSuccess = ebs.getBaseUrl
() + "/" + params.settings.maskNextStepURLSuccess;
```

```

    }
    console.log("params.settings.maskNextStepURLSuccess " +
params.settings.maskNextStepURLSuccess);
    ebs.callActionByName("FTOS_DFP_ESign_Endpoint", {params:
params}, function(result) {

        ebs.hideLoadingPanel();
        params = result.UIResult.Data.params;

        //remember to delete
        //console.Log(JSON.stringify(params));

        sessionStorage.setItem("sessionParameters", JSON.stringify
(params));
        if(params.settings.redirecttoNamirialLink == true){

            window.location.href = params.site;
        } else {
            window.history.back();
        }

    });
} else ebs.showMessage("There is no file to sign!", "error");
}
signContract();
// Upload document button
$('#btnUpload').click(function() {
    if(!params.existingFile){
        $('#fileInput').trigger('click');
        $('#fileInput').on('change', function(event) {
            var fileNotBase64 = event.target.files[0];
            params.existingFile = true;
            $('#fileInput').removeAttr('style');

            var reader = new FileReader();
            reader.onload = function(fileLoadedEvent) {
                var fileBase64 = fileLoadedEvent.target.result;
                fileBase64 = fileBase64.substring
(fileBase64.indexOf(",")+1);
                params.file = fileBase64;
                params.fileIsBase64 = true;
            };
            reader.readAsDataURL(fileNotBase64);
        });
    } else ebs.showMessage("A file was already uploaded!",
"error");
});

```

```
// Sign document button
$("#btnSign").click(function(e) {
    signContract();
});
```

To have a custom flow rendered on a data form-driven journey, you should include an empty div within the Monaco editor: `<div>&nbsp;</div>`.

#### 4 Define who has access to the journey

If your business case requires that the custom flow is available to designated roles within your organization, click the **Security Roles** tab and add the security roles who should have access to them. If no security roles are added here, all users will be able to view the journey. For more information, see ["Security Roles" on page 1245](#).

#### 5 Save the journey

If you want to save and close the journey, click **Save and close**.

If you want to save the journey and continue working on it, click **Save and reload**.

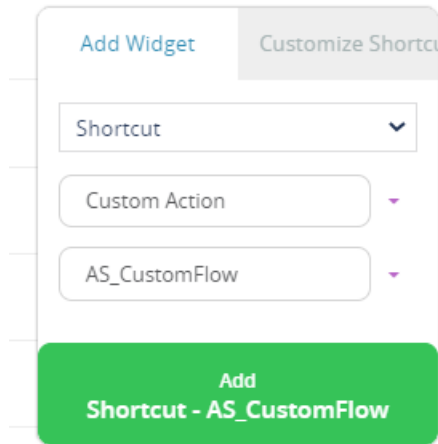
#### 6 Display the Custom Flow on the Portal

To add the flow to the **dashboard**, complete the following steps:

1. In the main menu, navigate to **Digital Experience > Digital Frontends > Dashboards**.
2. From the list, choose the dashboard name where you wish to place the flow.
3. The layout of the dashboard opens. Navigate to the right panel where there is a control table which adds the widgets.

#### HINT

If the dashboard does not have the checkbox "Show on homepage" equal true, then the dashboard will not be shown in the FintechOS Portal. Thus, any widget added to that dashboard will not be displayed.



4. Select the type "Shortcut", followed by the "Custom Action", and lastly the entity.
5. Click "Add Shortcut-*EntityName*".

To add the flow as a **menu item**, complete the following steps:

1. In the main menu, navigate to **Digital Experience > Digital Frontends > Menu Items**.
2. Click **Insert**.

3. The "Add menu item" page opens. Fill in the following:

Field	Description
Type	Select from the list: <ul style="list-style-type: none"> <li>• Entity</li> <li>• <b>Custom journey</b></li> <li>• Report</li> <li>• Menu section.</li> </ul>
Custom Journey	This field will open once you have chosen the type. Select the flow you wish to add.
Display Name	Insert the name of the flow.
Parent menu item	This field is optional, only if you are adding the flow inside another menu item. It shows the name of the menu item inside where the flow will appear.
Icon URL	Select the desired icon.
Disabled	Tick this checkbox if you wish to disable it.

4. Click **Save and reload**. The Edit menu item page appears.
5. Add security roles by clicking the **Insert existing**.

## Create Custom Controls

You can create custom controls you can use in your custom flows, by using either DevExtreme widgets or jQuery.

### Create Custom Controls Using DevExtreme Widgets

Based on the control you want to add, use the corresponding DevExtreme widget. For the complete list of supported widgets, see [DevExtreme UI\\_Widgets](#).

To add a custom control using widgets:

1. Add the control holder element.
2. Generate the desired control.
3. Set the value for the generated control.

```
//Add control holder element
$("table").append("<tr><td><div> Age </div></td><td><div
id='controlHolder '></div></td></tr>");

//generate desired control
$("#controlHolder").dxNumberBox({
    min: 0,
    max: 100,
    showSpinButtons: true,
    step: 1
});

//set value for generated control
$("#controlHolder").dxNumberBox("instance").option("value", 20);

//change property
$("#controlHolder").dxNumberBox("instance").option("step", 3);

//set event listener
$("#controlHolder").dxNumberBox("instance").option("onChange",
function() {
    console.log("value changed")
}
);
```

### Display Tabular Data Using DevExtreme Widgets

You can use the [dxDataGrid](#) UI component to create interactive tables that support sorting, grouping, filtering, data exports and other advanced capabilities. E.g.:

```
<div class="grid"></div>

<script>
    await ebs.injectKnownScriptAsync("dxDataGridExportReferences");
    $(".grid").dxDataGrid({
        dataSource: [
```

```

        { ContractNumber: 1, ContractDate: "5/10/2024",
Customer: "Customer 1" },
        { ContractNumber: 2, ContractDate: "6/10/2024",
Customer: "Customer 2" },
        { ContractNumber: 3, ContractDate: "7/10/2024",
Customer: "Customer 3" },
    ],
    columns: [
        { dataField: "ContractNumber", caption: "Contract
Number", dataType: "number", alignment: "center" },
        { dataField: "ContractDate", caption: "Contract Date",
dataType: "date" },
        { dataField: "Customer", caption: "Customer", dataType:
"string" },
    ],
    paging: {
        pageSize: 10,
    },
    pager: {
        visible: true,
        showPageSizeSelector: true,
        allowedPageSizes: [10, 25, 50, 100],
    },
    allowColumnReordering: true,
    width: "100%",
    export: {
        enabled: true,
        ignoreExcelErrors: true,
        texts: {
            exportAll: "Export all data",
            exportTo: "Export",
        },
    },
    onExporting: function (e) {
        let workbook = new ExcelJS.Workbook();
        let worksheet = workbook.addWorksheet("Main sheet");
        DevExpress.excelExporter
            .exportDataGrid({
                worksheet: worksheet,
                component: e.component,
            })
            .then(function () {
                workbook.xlsx.writeBuffer().then(function
(buffer) {
                    saveAs(new Blob([buffer], { type:
"application/octet-stream" }), "MatrixData.xlsx");
                });
            });
    }

```

```
});
    },
});
</script>
```

Contract Number	Contract Date	Customer
1	5/10/2024	Customer 1
2	6/10/2024	Customer 2
3	7/10/2024	Customer 3

102550100

1

IMPORTANT!

To enable the export functionality, you must import the **dxDataGridExportReferences** library.

Modify Controls' Advanced Properties

You can modify the advanced properties of an existing control created using DevExpress widgets. For information on the controls available properties, see the specific [control documentation](#) provided by DevExpress.

```
//get control
var control = $("#controlHolder").dxNumberBox("instance")
//change properties
control.option("step", 3);
control.option("onChange", function() {
    console.log("value changed")
});
control.option("visible", false);
```

## Create custom controls using JQuery

**IMPORTANT!** We do not guarantee the Client -side methods functionality when using jQuery.

To create a custom control using jQuery, follow these steps:

1. Add the control holder element.
2. Generate the desired control.
3. Set the value of the generated control.
4. Set the event listeners.

```
createCustomControl
//Add control holder element
$("table").append("<tr><td><div> Age </div></td><td><div
id='controlHolder'></div></td></tr>");

//Generate desired control
ebs.generateGenericControl({
    AttributeTypeValue: 5
}, "controlHolder"
);

//Set value of generated control
$("#controlHolder input").val(20);

//Set event listeners
$("#controlHolder input").change(function() {
    alert("Handler for controlHolder called.");
});
```

## Custom UI

Custom UIs are generic containers that enclose forms/flows/steps and that run in the context of the enclosed component. Since custom UIs are reusable (you can associate a custom UI with multiple forms for instance), you can use them to segregate form

layout and styling from the forms' content.

For instance, you can create a custom UI that contains a company header, followed by the enclosed component, followed by a company footer. Thus, all form driven flows associated with that custom UI will be displayed preceded by the company header and followed by the company footer.

With custom UIs, you can customize the user interface without having to modify the layout/styling-related code in your client-side forms and steps each time a package is updated. All form driven flow and app data form UI customizations can be fully decoupled from the client-side functional logic.

### IMPORTANT!

To work with custom UIs, you need the **Developer** security role. For more information, see the ["Default Security Roles" on page 1245](#) documentation.

## Configuration

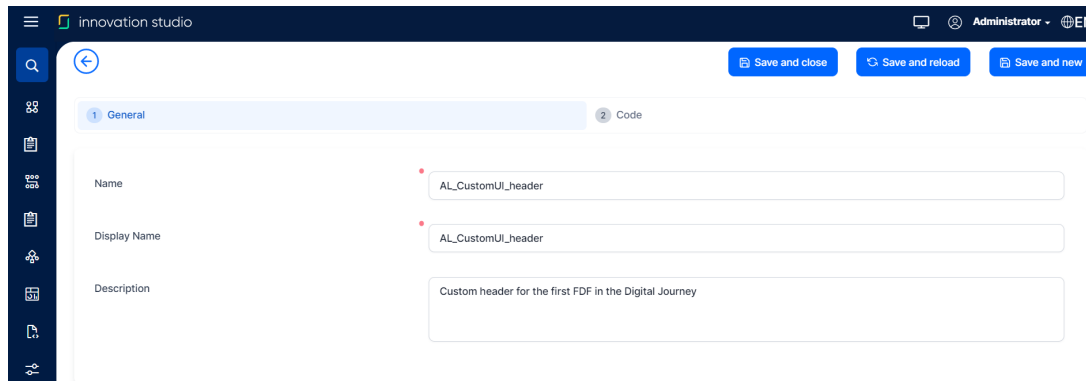
There are two main tasks for setting up a Custom UI: creating the Custom UI and associating it with a Form/Flow/Step.

### NOTE

You can use out-of-the-box templates for UI customization. Read more about them on the [Custom UI Template Kit](#) page.

### Create a Custom UI

1. Navigate to **Digital Experience > Digital Frontends > Custom UI**.
2. Select **+Create**.
3. Fill in the **Name** and **Display Name** fields (the Description field is optional).

4. Select one of the **Save** options.

In the **Code** tab, enter the custom UI HTML and JavaScript code. The code must return an object with 2 properties (methods) that the platform will know to execute in a Form Driven Flow code execution flow (beforeRendering and afterRendering):

```
return = {
  beforeRendering: function(formName),
  afterRendering: function(formName, stepName)
}
```

Example:

```
var customControl = ebs.importClientScript("myControls")
return {
  beforeRendering: function(formName){
    console.log("before render: ", formName, formData,
    formScope);
    ebs.rendering.registerInterceptor("name", new
    customControl.textEditor());
  },
  afterRendering: function(formName, stepName){
    console.log("after render", formName, stepName, formData,
    formScope);
  },
}
```

The beforeRendering and afterRendering functions can also return Promises, for example:

```

return {
  beforeRendering: function(formName){
    return ebs.getByIdAsync("cuitest", "a11d2f5f-4ff9-4904-
8efd-0790470e6850")
    .then(function(){
      console.log("setting marker from form custom
after");
      formScope.marker = "set from testCA(form after
custom)";
    });
  }
}

```

Use a {standardUI} token to inject the content of the associated form/flow/step in the Document Object Model (DOM) of the custom UI.

### Clone a Custom UI

If you wish to create a variation based on an existing Custom UI, you can clone the desired Custom UI and use it as a starting point for further modifications.

To clone a custom UI:

1. Open your Custom UI and in the General tab, select **Clone Custom UI**.

The screenshot shows the 'General' tab of a Custom UI configuration. At the top, there are three buttons: 'Save and close', 'Save and reload', and 'Open in code'. Below these are two tabs: '1 General' (active) and '2 Code'. In the 'General' tab, there are two input fields: 'Name' and 'Display Name', both containing the text 'AL\_DemoUI2'. A red rectangular box highlights the 'Clone Custom UI' button located in the top right corner of the configuration area.

2. In the pop-up window that opens, provide a name for the clone.
3. Select **Clone** to create your new Custom UI.

### Associate a Custom UI with a Form/Flow/Step

1. Navigate to **Digital Experience > Form Driven Flows**.
2. Select the flow (or step) you want to apply your Custom UI to.

- Go to the **Advanced** tab > **Set Custom UI** > select the Custom UI from the drop-down list.

The screenshot shows the 'Set Custom UI' dialog box. On the left, there is a sidebar with a tree view containing 'Custom UI', 'Client-Side app logic', 'UI Logic', 'Event handlers', and 'Server-side app logic'. The main area of the dialog has a 'Set Custom UI' section with a dropdown menu labeled 'Select a value...'. The dropdown is open, showing a list of templates: 'defaultUITemplate', 'templateKraken', 'templateKrakenLanding' (which is highlighted), and 'templateKrakenResolution'. Below the dropdown, there is a 'Name' input field.

- Select **Yes** in the pop-up confirmation window.

This action does not alter in any way the Form Driven Flow or App Data Form metadata. It is a means of linking them to use the Custom UI. The link is created in a new configuration item of type **formUICustomization** and is automatically added to your context Digital Asset, which can be different from the Digital Asset containing the Form Driven Flow.

### NOTE

formUICustomizations are treated as root configuration items and you can move them from one Digital Asset to another. We recommend you to keep the UI customization configuration items in a separate Digital Asset - in case you need to deploy patches or fixes, this will speed up the process.

The screenshot shows the 'Configuration Items' tab. At the top, there are tabs for 'General', 'Configuration Items' (which is selected and highlighted with a red box), 'Related Digital Solutions', 'Data Import Files', 'Custom Files', 'Dependencies', 'Configuration Items Migration', and 'Advanced'. Below the tabs, there are buttons for '+ Add', 'Branch Digital Asset', and 'Remove'. The main area contains a table with the following data:

Entity Name	Record Name	Parent Name	Modified On	User Name
<input checked="" type="checkbox"/> formUICustomization	customUI_EntForCustUI_AL_F...		12/10/2022 12:19	Administrator
<input type="checkbox"/> customAction	AL_DemoUI		12/10/2022 12:21	Administrator

## Events flow execution

1. If there is a configuration made on **Form** level:

The HTML template used will be the one of the configured **customForm**.

While rendering:

- The `beforeRendering` function will be called after the `BeforeEvents` of the **Form**
- The `afterRendering` function will be called after the `AfterEvents` of the **Form**

*AfterEvents(Form) -> AfterEvents(Step) -> afterRendering(customUI)*

2. If there is a configuration made on **Step** level too:

- The HTML template used on the Step will be the one of the configured **customForm**
- If the configured HTML Template is null and there is a **Custom UI** configured on the **Form**, then the HTML template used will be the one from the configured **customForm** of the **Form**
- The `afterRendering` function will be called after the `AfterEvents` of the Step (if the configuration exists on Form level too, the `afterRendering` configured on Form level will also be executed)

### NOTE

The `beforeRendering` function from a *Step* Custom UI is not executed, as `beforeRendering` is linked in execution with the *Form* `beforeEvent`.

You can also register your custom event to be triggered on step navigation (when the step changes, before the `AfterEvents` from Form and Step), including forward / backward navigation. E.g.:

```
formScope.myChangeSection = formData.on("stepNavigation", function
(formName, stepName){
  console.log("changeSection", formName, stepName, formData,
formScope);
});
```

And when needed, you can unregister your event:

```
formScope.myChangeSection.dispose()
```

### IMPORTANT!

Your code is executed in form context, so besides having access to `formName` and `stepName`, you will have direct access to `formScope` and `formData`.

For UI manipulation, you need to write your code in `afterRendering`. Having in mind the execution flow and that this event will be called any time the step after event is executed, if you want to avoid multiple executions for each step, you need to use flags in `formScope`.

The execution sequence follows the pattern below:

*AfterEvens(Form) -> AfterEvens(Step) -> afterRendering(customUI configured on Form)  
-> afterRendering(customUI configured on Step)*

For more details about events flow execution, see [Code Execution Sequence](#).

## Custom UI on Custom Flows (B2C)

This feature exposes a function that can be called from the After Generate Js of a custom action that will load a customUI by name. It is exposed as a member of a context object available in the After Generate Js of the *customAction* entity: `context.setCustomUI("customUIName")`. The `context.setCustomUI(customUIName)` function called from the After Generate Js of the customFlow will:

- Fetch the data for the specified customUI
- Check if the template contains the `FTOS_B2C_UI_Wrapper` marker and then check if the same customUI already wraps the UI (and if not, use it to wrap the UI)
- Execute the `afterRendering` of the customUI
- Execute the `stepNavigation` event handler with the parameters: `formName: null`, `stepName: null`, `customFlowName: the name of the custom flow`

To ensure that the loader is disposed of after the customUI is fully rendered, the After Generate Js of the Custom Flow must return a Promise. As `context.setCustomUI` is a Promise, we recommend using it like this:

```
return context.setCustomUI("customUIName").then(function(){
    //any other logic
});
```

## Custom UI on B2C journeys

B2C journeys exposed in B2C portals are built to provide anonymous customer interactions, not requiring a standard user authentication. Usually, these journeys are subject to a full UI Customization.

To view more details about unauthenticated user access, see ["Anonymous Frontends \(deprecated\)" on page 391](#).

When the Digital Experience Portal runs in B2C User Journey mode, the standard portal layout components are removed by default. This means no menu, no header, no footer, etc. Only the exposed journeys flows are rendered. However, with the Custom UI functionality, you can customize the look and feel of your journeys in B2C mode as well.

## Custom UI on Automation Blocks

You can further enhance your digital journeys by applying the Custom UI functionality to automation blocks. To associate a Custom UI to an automation block, go to the **Processor Settings** section of your desired automation block and select your Custom UI. For more information, see [Automation Blocks](#).

## Custom UI on the entire UI (B2C)

You can wrap the entire digital journey UI (layout, header, footer, logo, etc.) to match your business needs, with the CustomUI functionality. If there is a custom UI set on the form and the HTML template of the associated Custom Action contains an element with the id “FTOS\_B2C\_UI\_Wrapper“, you can use that template to replace the {standardUI} token and wrap the whole UI with the customUI.

### Scenarios

- Assuming you have a Digital Journey with 2 form driven flows, both having the FTOS\_B2C\_UI\_Wrapper in their respective configured customUI:

- If the user starts the Digital Journey on the first form driven flow, then the journey UI is wrapped with the first customUI template.
- If the user progresses to the second form driven flow, then the journey UI is wrapped with the second customUI template.
- In case of a Digital Journey with 2 form driven flows, with only the first flow having the FTOS\_B2C\_UI\_Wrapper in the configured customUI (the second flow does not have one):
  - If the user starts the Digital Journey on the first form driven flow, then the journey UI is wrapped with the first customUI template.
  - If the user progresses to the second form driven flow, then the first customUI template still wraps the UI.
- In a Digital Journey with 2 form driven flows, where only the second flow has the FTOS\_B2C\_UI\_Wrapper in the configured customUI:
  - If the user starts the Digital Journey on the first form driven flow, nothing happens.
  - If the user progresses to the second form driven flow, then the journey UI is wrapped with the second customUI template.

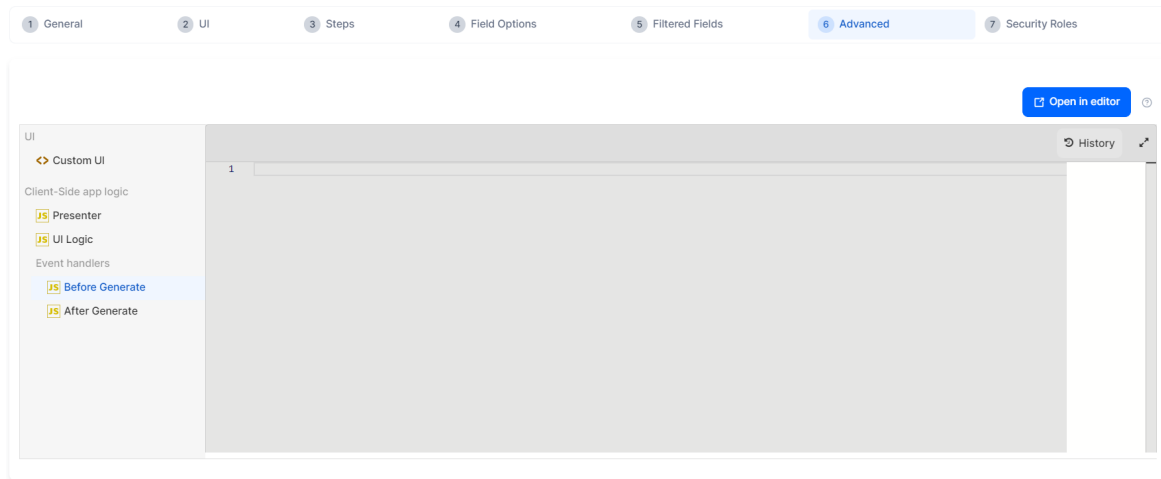
## Code Execution Sequence

In FintechOS Studio, you can attach client-side and server-side JavaScript code in multiple places within forms, flows, and views. The script's location determines when the code is executed, for instance, after a view is loaded, before a form is displayed, or between two form steps.

### HINT

See the [Client SDK](#) and [Server SDK](#) reference documentation for information on how you can make the best use of the FintechOS Platform's coding capabilities.

## Data Forms



In the **Advanced** section of "Data Forms" on page 142, under Event Handlers, there are two tabs called **Before Generate** and **After Generate** where you can add client code that runs when the form is loaded.

In addition, there are the following two tabs:

- **Presenter:** add client-side code when the App Data Logic is encapsulated in methods providing the desired functionalities. These methods are invoked when needed in events scripts or other UI events (button pressed, etc.).

FTOS Client Side SDK offers methods that help build Presenter functionalities, such as [change Business Status](#), [generate Report](#), [refresh Data](#), [call Action](#), accessing and changing the context data in the [form](#), [upload/download file](#), [Save](#), [goToRoute](#), [goToUrl](#), [goToDashboard](#).

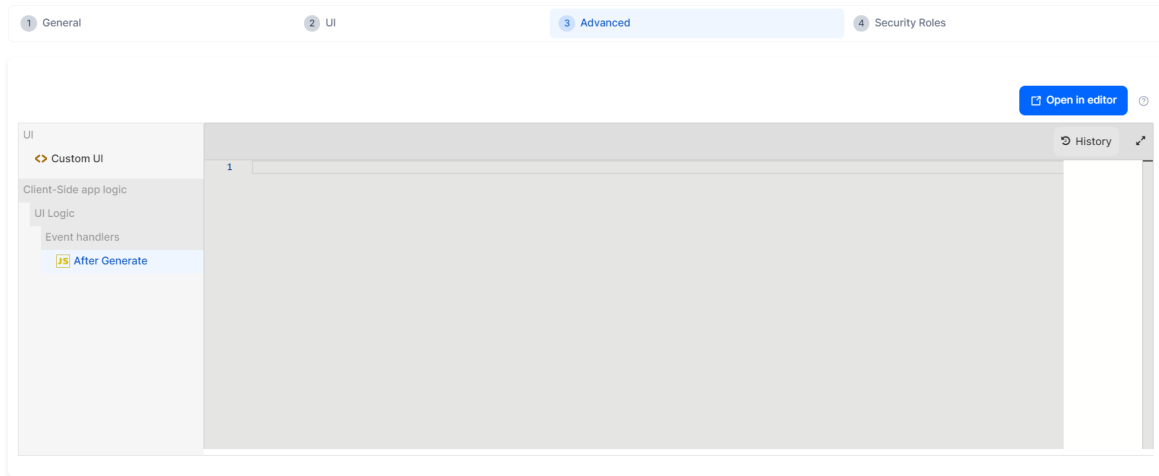
- **UI Logic:** add client-side code when the UI Logic is encapsulated in methods providing the desired functionalities. These methods are invoked when needed in events scripts or other UI events (button pressed, etc.).

FTOS Client Side SDK offers methods that help build UI Logic, such as [setFormAttributeReadOnly](#), [setFormAttributesReadOnly](#), [setFormSectionAttributesReadOnly](#), [setFormAttributeRequired](#), [removeAttributeEditButton](#), [removeFormEditButton](#), [generateGrid](#), [generateHTMLWidget](#), [showMessage](#), [alert](#), [confirm](#), [show/hide LoadingPanel](#), [makeOptionSetRadioGroup](#), [createWizardObject](#), etc.

When a data form with no steps is loaded, the code executes in the following sequence:

1. Code in the **Before Generate** tab.
2. Form HTML code.
3. Code in the **After Generate** tab.

### Data Form Steps



In the **Advanced** section of data form steps, there is a tab called **After Generate** where you can add client code that runs when the step is loaded.

When an entity form with steps is loaded, the code executes in the following sequence:

1. Code in the form's **Before Generate** tab.
2. HTML code for all steps.
3. When a step is displayed:
  4. Step HTML code.
  5. Form **After Generate** code.
  6. Step **After Generate** code.
7. Repeat from Step 3 for each subsequent step that is displayed.

#### IMPORTANT!

Step **After Generate** code is executed only the first time that the step is displayed. If,

in the entity form, you navigate away from a step and later return to the same step, the step's **After Generate** code will not run again.

## Form Driven Flows

In the **Advanced** section of form driven flows, there are multiple tabs that allow you to attach client-side and/or server-side JavaScript code to the flow.

- **UI**
  - **Custom UI** - Create generic containers that enclose forms/flows/steps and that run in the context of the enclosed component. Read more on the [Custom UI](#) page.
  - **Navigation styles** - Settings for the flow's appearance, including labels, navigation button positions, colors, and title bar. Refer to the [Client SDK](#) documentation for details on using the [ebs.createWizardObject](#) function.
- **Client-side app logic**
  - **Presenter:** add client-side code when the App Data Logic is encapsulated in methods providing the desired functionalities. These methods are invoked when needed in events scripts or other UI events (button pressed, etc.).  
  
FTOS Client Side SDK offers methods that help build Presenter functionalities, such as [change Business Status](#), [generate Report](#), [refresh Data](#), [call Action](#), accessing and changing the context data in the [form](#), [upload/download file](#), [Save](#), [goToRoute](#), [goToUrl](#), [goToDashboard](#).
  - **UI Logic:** add client-side code when the UI Logic is encapsulated in methods providing the desired functionalities. These methods are invoked when needed in events scripts or other UI events (button pressed, etc.).  
  
FTOS Client Side SDK offers methods that help build UI Logic, such as [setFormAttributeReadOnly](#), [setFormAttributesReadOnly](#), [setFormSectionAttributesReadOnly](#), [setFormAttributeRequired](#), [removeAttributeEditButton](#), [removeFormEditButton](#), [generateGrid](#), [generateHTMLWidget](#), [showMessage](#), [alert](#), [confirm](#), [show/hide LoadingPanel](#), [makeOptionSetRadioGroup](#), [createWizardObject](#), etc.

- **Before Generate** - Code executed before the user interface is initially rendered. The script runs once per flow per mode (insert/edit). This means that, for an insert flow, Before Events may run twice: at the flow's start and when transitioning from insert to edit mode (typically when the user clicks Next for the first time to advance from the first step to the second step).
- **After Generate** - Code executed after a step is initially rendered and before the step's After Events script (see ["Form Driven Flow Steps" below](#)). The script runs once per step per mode (insert/edit). This means that if a step is rendered for the first time when the flow is in the insert mode, then revisited (re-rendered) after the flow transitioned from the insert mode to the edit mode (e.g. by navigating back in the flow using the Previous button), the After Events will be triggered twice.
- **Navigation styles** - Settings for the flow's appearance, including labels, navigation button positions, colors, and title bar. Refer to the [Client SDK](#) documentation for details on using the `ebs.createWizardObject` function.
- **Server-side app logic**
  - **Before Save** - Code executed before each step's Before Save Server Event script (see ["Form Driven Flow Steps" below](#)).
  - **After Save** - Code executed before each step's After Save Server Event (see ["Form Driven Flow Steps" below](#)).

**NOTE**

Form driven flows always include steps. Check the ["Form Driven Flow Steps" below](#) section below for further details about the code execution sequence.

**Form Driven Flow Steps**

In the **Advanced** section of each form driven flow step, there are multiple tabs that allow you to attach client-side or server-side JavaScript code to each step.

- **UI**
  - **Custom UI** - Create generic containers that enclose forms/flows/steps and that run in the context of the enclosed component. Read more on the [Custom UI](#) page.
- **Client-side app logic**
  - **After Generate** - Code executed after the step's user interface is initially rendered. The script runs once per step per mode (insert/edit). This means that if a step is rendered for the first time when the flow is in the insert mode, then revisited after the flow transitioned from the insert mode to the edit mode (e.g. by navigating back in the flow using the Previous button), the step's After Events will be triggered twice.
  - **Before Save** - Code executed before the step's data is saved to the database (e.g.: when you advance to the next step or, if it's the final step in the flow, when you click the Finish button).
  - **After Save** - Code executed after the step's data is saved to the database (e.g.: when you advance to the next step or, if it's the final step in the flow, when you click the Finish button).
- **Server-side app logic**
  - **Before Save** - Code executed before the step's data is saved to the database.
  - **After Save** - Code executed after the step's data is saved to the database.

### Good Coding Practices

- The After Generate scripts are executed after a step is rendered for the first time in two stages: first the form driven flow's After Generate (which are common for all steps in the flow), then the step's After Generate (which are specific to the step). This allows you to implement good coding practices, by reusing code that is common for all steps.

- The server-side Before/After Save code enables you to decouple business logic related to database CRUD operations into journey-specific and entity-specific (event triggered automation scripts) categories.

### Use Case Scenarios

The examples below show detailed code execution sequences for various form driven flow scenarios, including the execution sequence for elements such as ["Form Actions"](#) on page 258, ["Flow Control"](#) on page 241 rules, or ["Event Triggered Automation Scripts"](#) on page 1163.

## Start a Form Driven Flow and Render the First Step

### Advance to the Next Step (in Edit Mode)

### Finalize a Form Driven Flow

## End-to-End User Journey

The examples below show the code execution sequence for a two step form driven flow where the user:

1. **Starts the flow.**
2. Advances from the first step to the second step (clicks **Next**).
3. Returns from the second step to the first step (clicks **Previous**).

4. Goes back to the second step (clicks **Next**).
5. Finalizes the flow (clicks **Finish**).

The journey is illustrated for both the insert and edit scenarios.

Insert Record	Edit Record

### HINT

Notice the additional Before Generate and After Generate scripts that were "reset" when the insert flow transitioned to the edit mode.

## App Data Forms

In the **Advanced** section of app data forms, there are multiple tabs that allow you to attach client-side or server-side JavaScript code to the form.

- **UI**
  - **Custom UI** - Create generic containers that enclose forms/flows/steps and that run in the context of the enclosed component. Read more on the [Custom UI](#) page.
- **Client-side app logic**
  - **Before Generate** - Code executed before the user interface is initially rendered. If the app data form is re-rendered (e.g.: the user clicks the Save and Reload button), the script will run again.
  - **After Generate** - Code executed after a step is initially rendered and before the step's After Events script (see "[App Data Form Steps](#)" on the next page). If the step is re-rendered (e.g.: the user clicks the Save and Reload button), the script will run again.

- **Server-side app logic**
  - **Before Save** - Code executed before form data is saved to the database (e.g.: the user clicks Save and Close or Save and Reload).
  - **After Save** - Code executed after the form data is saved to the database (e.g.: the user clicks Save and Close or Save and Reload).

### App Data Form Steps

In the **Advanced** section of app data form steps, there is a tab called **After Events** that allows you to attach client-side code to the step.

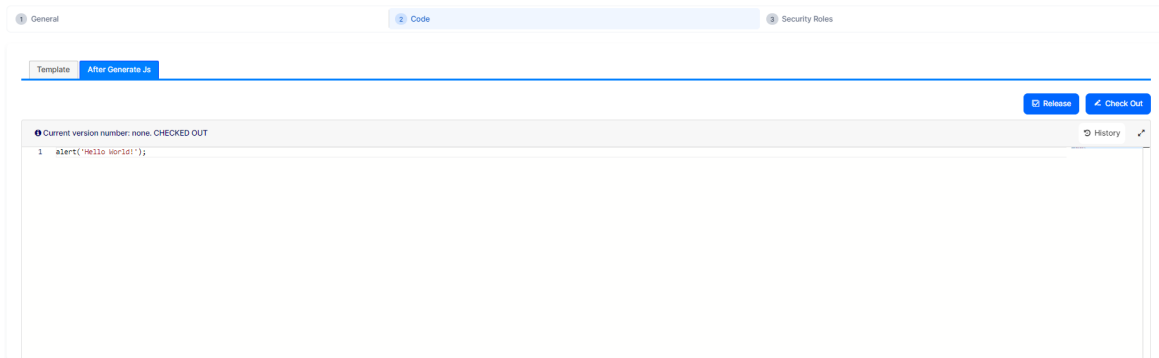
The After Events code is executed after a step is initially rendered and the app data form's After Events script is run. If the step is re-rendered (e.g.: the user clicks the Save and Reload button), the script will run again.

## Use Case Scenario

The example below illustrates the code execution sequence for a two step app data form where the user:

1. **Opens the form.**
2. Advances from the first step to the second step (clicks **Next**).
3. Saves the form data (clicks **Save and Reload**).
4. Closes the form (clicks **Save and Close**).

## Custom Flows



In the **Code** section of custom flows, there are two tabs called **Template** and **After Generate Js** where you can add client code.

The code in the **Template** tab contains the flow's HTML code. Since this is a custom design, you must provide the form's entire content in the **Template** tab.

The code in the **After Generate Js** tab is executed after the flow's HTML code is rendered.

1. Flow **Template**.
2. Flow **After Generate Js**.

## Custom UIs in Forms/Steps

Depending on the level where a Custom UI is applied, the execution sequence follows the patterns below:

1. If there is configuration made on **Form** level:

- The HTML template used will be the one of the configured **customForm**.

While rendering:

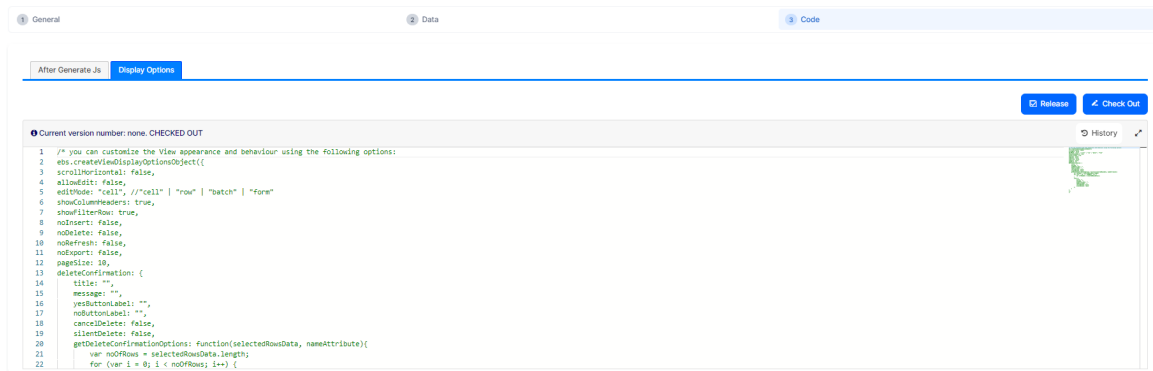
- The `beforeRendering` function will be called after the `BeforeEvents` of the **Form**
- The `afterRendering` function will be called after the `AfterEvents` of the **Form**  
`AfterEvens(Form) -> AfterEvens(Step) -> afterRendering(customUI)`

2. If there is configuration made on **Step** level too:

- The HTML template of the Step used will be the one of the configured **customForm**
- If the configured HTML template is null and there is a **Custom UI** configured on the **Form**, then the HTML template used will be the one from the configured **customForm** of the **Form**
- The `afterRendering` function will be called after the `AfterEvents` of the Step (if the configuration exists on Form level too, the `afterRendering` configured on Form level will also be executed)

*AfterEvens(Form) -> AfterEvens(Step) -> afterRendering(customUI configured on Form)  
-> afterRendering(customUI configured on Step)*

## Entity Views



In the **Code** section of entity views, there are two tabs called **After Generate Js** and **Display Options** where you can add client code.

The code in the **After Generate Js** tab runs after the view is loaded.

The code in the **Display Options** tab allows you to define settings for the view's look and feel, such as scrolling, headers, filter row, page size, or the ability allow records editing in the view. The **Display Options** tab is pre-populated with the available settings commented out which you can modify according to your preference. These settings supplement the view customizations.

1. Display view using any configured **Display Options**.
2. Run the **After Generate Js** code.

Charts

Edit Chart

Name

MKT\_Dashboard

Chart Title

MKT\_Dashboard

Chart Base Type

series

Chart Type

bar

Name field

camp

Value field

act

Series argument field

channel

Show Legend

☒

Legend horizontal alignment

center

Legend vertical alignment

bottom

Show Labels

☒

Axis X Title

Axis Y Title

Render Type

[none]

Container CSS Class

Container CSS Inline Style

After Generate Js

1

alert('Hello World!');

Fetch value

Fetch expression return object

1

return {

2

"entity": {

3

"alias": "base",

4

"name": "MKT\_Stats",

5

"attributeList": [

6

{

7

"name": "CampaignType",

8

"alias": "camp",

9

"attributeType": 3

10

},

11

{

12

"name": "ChannelProvider",

13

"alias": "channel",

14

"attributeType": 3

15

},

16

{

17

"name": "sum(activitiesNumber)",

18

"alias": "act",

19

"attributeType": 6

20

}

21

]

22

}

23

}

Chart

Preview

In the chart editing windows, there is a section called **After Generate Js** where you can add client code that is executed when the chart is rendered.

## Logic Blocks

Logic blocks allow you to configure event handlers through an intuitive no-code interface. You can define complex actions and conditional logic that can be triggered by attribute field updates, custom events, or at different stages of a step's execution sequence. Logic block are supported in both ["Form Driven Flows" on page 221](#) and ["App Data Forms" on page 933](#) steps.

### Add, Rename, Find, and Delete Logic Blocks

To manage logic blocks, open the form driven flow or app data form step where you wish to configure event handlers, then select the **Logic Blocks Designer** tab. Use the panel on the left to:

- **Create a logic block:** Click **+ Add New Logic Block** at the bottom of the panel.
- **Rename a logic block:** Click the logic block's name to select it, then type to overwrite the existing name.
- **Find a logic block:** Use the **Search logic block** field at the top of the panel to enter search keywords.
- **Delete a logic block:** Hover over the logic block's name and click the trashcan icon that appears next to it.

### Configure a Logic Block's Executable Logic

The executable logic defines the actions or computations that the logic block performs when its trigger conditions are met (see ["Configure a Logic Block's Trigger\(s\)" on page 319](#)). To add executable logic to a logic block:

1. In the Logic Block Designer, select the logic block by clicking its name in the left panel.
2. In the Logic Block Designer's main panel, click the **+** buttons to add conditional blocks or actions to the executable logic.

## Conditional Block

Conditional blocks are decision-making operators that allow selective execution of logic or a choice between alternate execution paths based on whether a condition is true or false. Conditional blocks contain three operands: **IF**, **THEN**, and **ELSE**:

Operand	Description
IF	<p>This is the condition, an expression that evaluates to either true or false based on the values of various form fields. Click + inside the IF node and use the point-and-click interface to define your expression.</p> <p>In its simplest form, the IF condition is based on a single form field evaluation (e.g.: Name is not blank). The available operators depend on the data type of the form field: generic operators such as <i>is blank</i> or <i>is not blank</i>, text specific operators such as <i>starts with</i> or <i>contains</i>, date specific operators such as <i>years since</i> or <i>days until anniversary</i>, etc.</p> <p>You can also create complex boolean logic by grouping conditions hierarchically with logical operators:</p> <ul style="list-style-type: none"> <li>• AND - All conditions are true.</li> <li>• OR - At least one condition is true.</li> <li>• NOT AND (NAND) - Not all conditions are true (at least one is false).</li> <li>• NOT OR (NOR) - All conditions are false.</li> </ul>
THEN	<p>This is the execution path taken when the IF condition evaluates to true. Click + inside the THEN node and use the point-and-click interface to define this path.</p> <p>You can include a sequence of <a href="#">"Data Actions" on the next page</a>, <a href="#">"Field Actions" on the next page</a>, <a href="#">"Logic Actions" on page 316</a>, <a href="#">"Navigation Actions" on page 317</a>, <a href="#">"UI Actions" on page 317</a>, or even other conditional blocks to create complex, nested decision trees.</p>

Operand	Description
ELSE	<p>This is the execution path taken when the IF condition evaluates to false. Click + inside the ELSE node and use the point-and-click interface to define this path.</p> <p>You can include a sequence of <a href="#">"Data Actions" below</a>, <a href="#">"Field Actions" below</a>, <a href="#">"Logic Actions" on the next page</a>, <a href="#">"Navigation Actions" on page 317</a>, <a href="#">"UI Actions" on page 317</a>, or even other conditional blocks to create complex, nested decision trees.</p>

## Data Actions

Data actions synchronize the user interface with the data store, either by refreshing the form's state from the database or by committing the current form data to the database.

Data Action	Description
Refresh Grid	Refreshes a relation UI component by querying the database. For more information, see <a href="#">"Add relations to the UI" on page 333</a> .
Refresh Business Status	Retrieves the form record's business status from the database. For more information, see <a href="#">"Business Workflows" on page 418</a> .
Refresh Form Data	Reloads the full form state from the database, including attribute fields, header items, form grids, and business status.
Save Data	Saves the form data to the database, ensuring all changes are committed.

## Field Actions

Field actions allow you to control the value and properties of form fields.

Field Action	Description
Set Value	Assigns a given value to a form field.
Set Empty	Clears the content of a form field, leaving it blank.
Set Visibility	Shows or hides a form field. Set to <code>true</code> to make the field visible, set to <code>false</code> to hide it.
Set Editable	Makes a field editable or read-only. Set to <code>true</code> to allow editing, set to <code>false</code> to make it read-only.

Field Action	Description
Set Required Level	<p>Sets the mandatory status of a form field:</p> <ul style="list-style-type: none"> <li>• <b>Required</b> – Displays a red dot in the upper-left corner, marking the field as mandatory. The record cannot be saved if this field is empty.</li> <li>• <b>Optional</b> – The field may be filled or left blank without restriction.</li> <li>• <b>Recommended</b> – Displays a blue dot in the upper-left corner, to indicate that it might be useful to fill in the field, but not mandatory.</li> </ul>

## Logic Actions

Logic actions allow you to trigger action handlers for state transitions, calling endpoints, generating reports, or client-side app logic.

Logic Action	Description
Change Business Status	Transitions the form record to the specified business status. For more information, see <a href="#">"Business Workflows" on page 418</a> .
Execute Endpoint	Calls a specific endpoint (see <a href="#">"Endpoints" on page 1213</a> ). If the endpoint has predefined input parameters or output structure, an <b>I/O Parameters</b> field is displayed, allowing you map form attributes to specific inputs or outputs.
Generate Report	Generates a report (see <a href="#">"Analytics" on page 1015</a> ). The report must have an Entity scope set to the form's main entity.
Call UI Logic Method	<p>Calls a public method defined in the form's UI logic (see <a href="#">"Code Execution Sequence" on page 301</a>). E.g.:</p> <pre> this.trimNames = function() {   formData.model.firstName =   formData.model.firstName.trim();   formData.model.lastName =   formData.model.lastName.trim(); }; </pre>

Logic Action	Description
Call Presenter Method	<p>Calls a public method defined in the form's Presenter section (see <a href="#">"Code Execution Sequence" on page 301</a>). E.g.:</p> <pre> this.goToStepTwo = function(){      var myState = {routeData: {         entityName: formData.entityName,         formId: formData.id,         formName: "default",         pageNo: "2",         type: "edit"}     };     EbsRouter.continueRoute(myState) } </pre>

## Navigation Actions

Navigation actions allow you to refresh the current page or redirect the user to a different destination within the platform.

### IMPORTANT!

Navigation actions are final actions. You cannot insert additional actions or conditional blocks in the executable logic after a navigation action.

Navigation Action	Description
Reload Page	Reloads the current page.
Navigate to Dashboard	Navigates to a specified dashboard (see <a href="#">"Dashboards" on page 362</a> ).
Navigate to Record	Opens a record referenced by the current form record in a specified form.
Navigate to List	Opens a specified entity view (see <a href="#">"Data Views" on page 112</a> ).
Navigate Home	Navigates to the FintechOS Portal homepage.
Navigate to Custom Flow	Navigates to a specified custom flow (see <a href="#">"Custom Flows" on page 283</a> ).

## UI Actions

UI Actions allow you to manipulate the user interface or control UI behavior by showing or hiding loading screens, enabling or disabling user interactions, and displaying messages.

UI Action	Description
Loading Panel	Shows or hides the loading screen animation.
Remove Edit Button	Removes the Edit button for an option set or lookup attribute field. This allows users to only select preexisting values and prevents them from editing the option set or the related entity of the lookup attribute.
Set Step Read Only	Changes the read-only status of all attributes on the form step: <ul style="list-style-type: none"> <li>• <code>true</code> - Attributes are read-only.</li> <li>• <code>false</code> - Attributes are editable.</li> </ul>
Show Message	Displays one of the following message types: <ul style="list-style-type: none"> <li>• <b>Alert</b> - Displays a modal window with one button. You can click + inside the <i>Function to be run</i> node and use the point-and-click interface to define "Data Actions" on page 315, "Field Actions" on page 315, "Logic Actions" on page 316, "Navigation Actions" on the previous page, "UI Actions" on the previous page, or conditional blocks to execute when the button is pressed.</li> <li>• <b>Confirmation</b> - Displays a modal window with two buttons. For each button, you can click + inside the <i>Function to be run</i> node and define executable logic in the same way as above.</li> <li>• <b>Toast</b> - Displays an Info, Success, Warning, or Error toast message.</li> </ul>

## Configure a Logic Block's Trigger(s)

A trigger is an event that causes a logic block's executable logic to run. You can configure multiple triggers for each logic block.

To configure a trigger:

1. In the **Logic Block Designer**, select the logic block by clicking its name in the left panel.
2. Click the **gear icon** in the top-right corner of the Logic Block Designer to open the right panel.
3. Fill in the sections in the **Triggers & Functionality** tab:
  - **Functionality**: Optionally enter a description of the logic block's triggers and executable logic.
  - **Triggers - Data Change**: Click **+Add Data Change Trigger** to run the executable logic when a specific form field is updated.
  - **Triggers - On Event**: Click **+Add On Event Trigger** run the executable logic based on one of the following events:
    - **After Data Refresh** - Any time form state is loaded from the database.
    - **On Step Enter** - Each time the step is displayed, after the step's user interface is rendered.
    - **Before Save** - Before the form data is saved to the database (e.g.: when clicking Next or Finish, but not when clicking Previous).
    - **Custom Event** - A custom event defined using the [formData.registerEvent](#) Client SDK method.

## Preview the Logic Block's Code

The code underlying each logic block is generated automatically as you configure the block from the point-and-click interface. To view this code:

1. In the **Logic Block Designer**, select the logic block by clicking its name in the left panel.
2. Click the **gear icon** in the top-right corner of the Logic Block Designer to open the right panel.
3. Select the **Code Preview** tab to display the generated code.

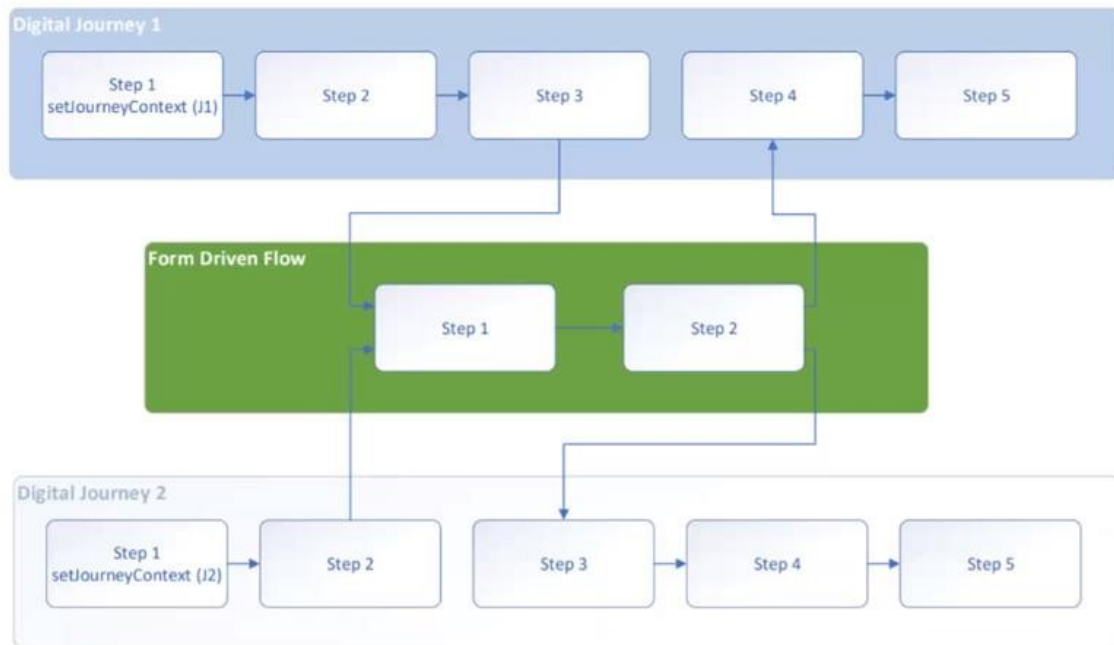
## Review the Logic Block Triggers

To review the step's logic blocks grouped by their triggers:

1. In the **Logic Block Designer**, click the **gear icon** in the top-right corner to open the right panel.
2. Select the **Triggers & Blocks** tab to view all triggers configured for the form step, along with their associated logic blocks.

## Digital Journey Context

Form driven flows can be shared between multiple digital journeys. The digital journey context allows you to customize a flow's behavior based on the journey that initiated it.



To set a digital journey context, use the [ebs.setJourneyContext](#) function on a form driven flow step (ideally on the first form driven flow step of the journey). This value will be propagated in subsequent journey steps.

To read the digital journey context, use the [ebs.getJourneyContext](#) function (typically in a downstream form driven flow that is shared between multiple journeys).

You can also use the [ebs.cleanJourneyContext](#) function to clear the digital journey context from session storage.

## Apply flow control rules only for specific digital journeys

To apply a flow control rule only for a specific digital journey, use the **Evaluate only for Journey** setting in the flow control rule editor.

Flow Control Rule

Name

goToJourney1

Form Section

SelectProduct

Description

Evaluate only for Journey

DJ API Demo

Define rule expression

And

+

×

Status

Equals

Active

Action

Cancel Navigation

Close Flow

○

Navigate to another Step

●

Navigate to another Flow

Select Digital Journey Flow

CopperLoan

Select Digital Journey Flow Step

SimulateLoan

Reference Attribute

the value of this attribute will be used as the primary key value for the target entity

Use Virtual Attribute

Attribute

LastName

Actions to be Performed

Select items to include

## IMPORTANT!

In a prior step of the journey, you must use the [ebs.setJourneyContext](#) function to set the digital journey context to the exact name of the digital journey. Otherwise, the evaluation will not work.

TABLE OF CONTENTS

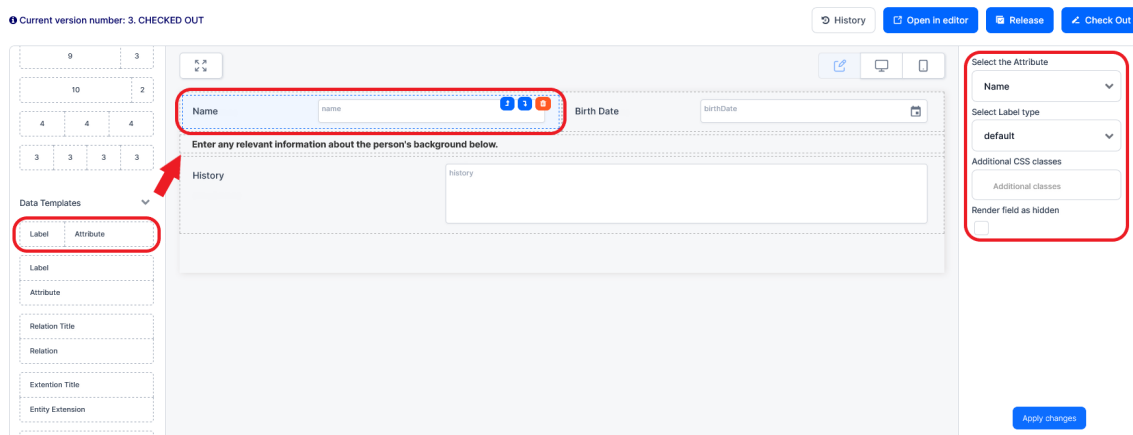
322

## UI Designer

The UI Designer is a tool that allows you to build responsive HTML user interfaces for your flows and forms by dragging and dropping the desired containers, attributes, relations, and other UI elements. It also allows you to separate your UI content from the design elements, by focusing only on the structure of your interfaces (use the FintechOS Platform's [UI customization](#) features to control the styling).

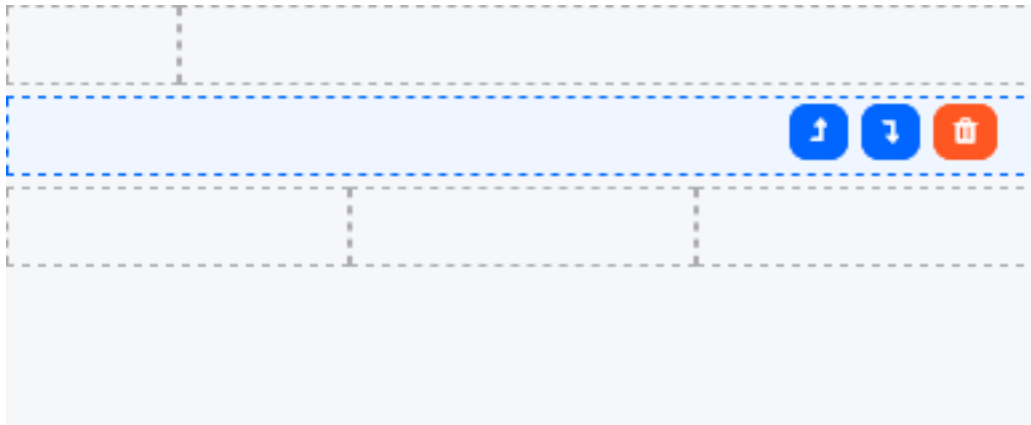
### Adding UI elements to the user interface

The UI Designer has two side panels. The left panel lists the available UI elements. The right panel controls the settings of the element currently selected in the editor's working area. To add an element to the user interface, drag it from the left panel to the editor's working area in the desired position, then edit its properties from the right panel.



### Moving and deleting UI elements

To move an UI element in the user interface, drag it in the editor or click the **up** or **down** arrow buttons visible on mouse hover to position it before the previous element or after the next element. To delete a UI element, click the **delete** trashcan button visible on mouse hover.



## Where can you access the UI Designer

The UI Designer is available in the following components:

Entity	Attribute
Entity Form (entity and form driven flow)	Template
Entity Form Step (entity and form driven flow)	Template

You can access the UI Designer, as follows:

- For entity forms, entity form steps, and custom flows, from the **UI** tab, by clicking UI Designer from the HTML Editor toolbar.
- For Widgets, from the **Code** tab of the Digital Frontends section to the menu item Widget.

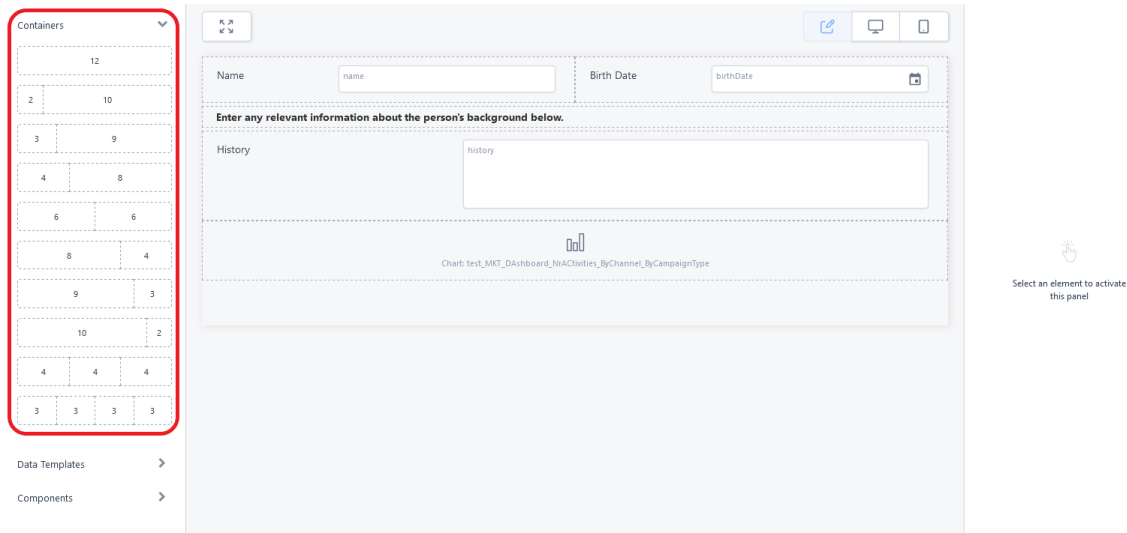
When using the UI Designer on custom flows and widgets, it will have two sections: HTML and JavaScript.

You can also access the UI Designer from an HTML field when inserting or editing entity records, but only with limited functionality. Using the UI Designer when inserting or editing records, you will not be able to:

- Change or remove attributes and relations
- Configure relations
- Add automation scripts to UI elements
- No automatically generated on click events after adding UI elements

## Design the UI layout with containers

You can use the **Containers** in the UI Designer's left panel to position the content in each form section (row) using responsive design. Each container is divided based on a grid of 12 columns that are grouped together in various combinations to create wide or narrow cells (three 4-column cells, a 3-column cell next to a 9-column cell, two 6-column cells, etc.).



To add a row template to the interface, drag and drop its corresponding container from the left panel to the editor's working area (see ["Adding UI elements to the user interface" on page 323](#)). Each cell of the template allows you to place data fields, UI components, or even other row templates to further refine your page layout.



Cells on the same row template are responsive, which means they appear on the same line on desktop devices, but are rearranged one on top of the other on mobile devices. Empty cells take up space in desktop (horizontal) mode, but are collapsed in mobile (vertical) mode.

Button A	Button B	Button C	fifty	fifty	Empty Cell	1	2	3
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.								

VS.

Button A	
Button B	
Button C	
fifty	
fifty	
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	
1	
2	
3	

## Collapsible Containers

A collapsible container is a variation of the one-column container that allows users to expand or collapse its content by clicking the arrow icon in the top-left corner. You can use them to manage screen space efficiently, organize information hierarchically, and allow users to control what they see.

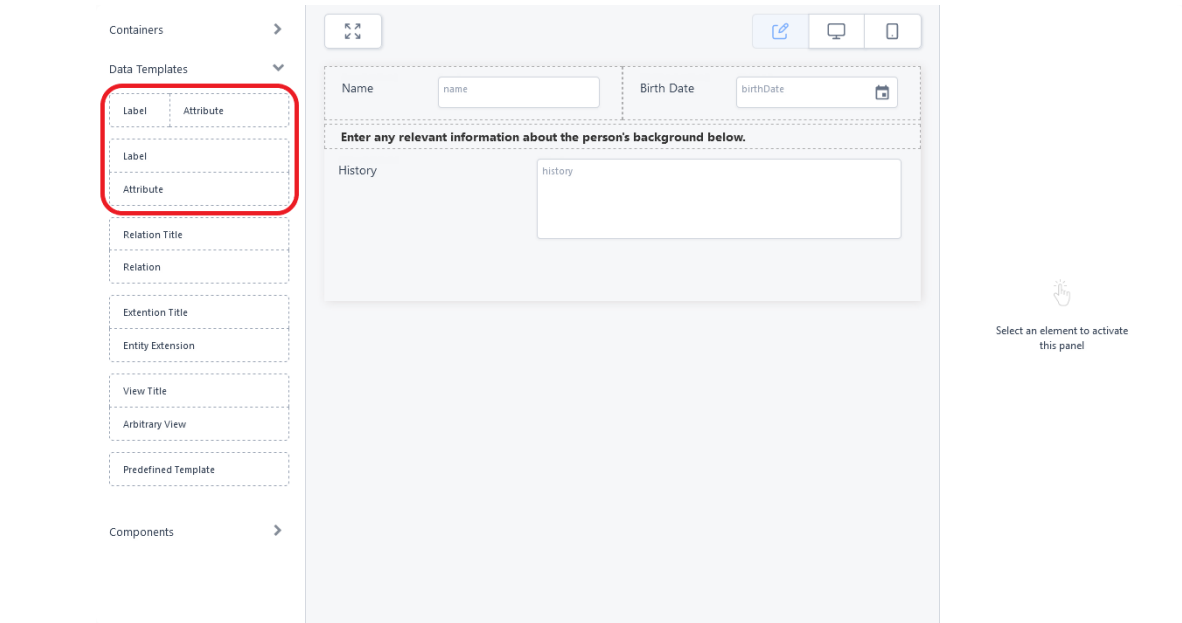
After you add a collapsible container to the UI, while the container is selected, fill in its properties in the right panel:

- **Title** - Enter the text to be displayed at the top of the container. To provide translations for each platform [language](#), click the translation button placed to the right of the title field. In this case, you must also specify a localization resource key (see below).
- **Enter the resource key name that will store the localizable text** - Enter a name for the resource key that will save this title in the [Localization Resources](#) page.
- **Initial State** - Choose whether the container should be expanded or collapsed by default.
- **Additional CSS Classes** - Add any CSS class names to style the container. These classes must be defined in one of the form's associated ["Style Sheets" on page 1209](#). You must press **Space** or **Enter** after typing each class name to confirm it.

Once configured, you can add content to the collapsible container just like any other container.

## Add attributes to the UI

1. In the ["UI Designer" on page 323](#)'s left panel, from the Data Templates section, select a vertical or horizontal attribute template and drag it to the working area in the desired position. For more information, see ["Adding UI elements to the user interface" on page 323](#).



2. While the attribute field is selected, fill in its properties in the right panel.

Property	Description
Select the Attribute	<p>Select the name of the attribute that you want to display. The attribute must belong to the form's data model. Otherwise, select the <b>** Mockup Attribute **</b> option to temporarily create a field in the user interface that is not yet linked to an attribute.</p> <div><p><b>NOTE</b> "Form Driven Mock-up Flows" on page 280 and "App Data Mock-up Forms" on page 944 allow only mockup attribute templates.</p></div>

Property	Description
Select Label Type	<ul style="list-style-type: none"> <li>• <b>default</b> - The attribute's display name is used as a field label.</li> <li>• <b>custom</b> - Manually enter the text for the field label. You can also manually add translations for each platform <a href="#">language</a>, by clicking the toggle translation button placed in the <b>Enter the desired text</b> field. In this case, you also need to choose a name for the resource key that will save this text in the <a href="#">Localization Resources</a> page.</li> <li>• <b>none</b> - No label is displayed and no label space reserved in the user interface when rendering the attribute field.</li> </ul>
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated "<a href="#">Style Sheets</a>" on <a href="#">page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

Property	Description
Render field as hidden	<p>The element will not be visible in the user interface. This allows you to bring an attribute value in the form's context without displaying it (e.g.: to retrieve the value of a birth date attribute to calculate age).</p> <div> <p><b>IMPORTANT!</b></p> <p>Hidden elements are still listed in the form's document object model. Do not disclose sensitive information in hidden fields.</p> </div>
UI Control	Allows you to customize the appearance of the attribute field. See <a href="#">"Custom UI Controls" on the next page.</a>
Fill in the Attribute Name	<p>(Mockup attributes only)</p> <p>Enter a name for the mockup attribute.</p>
Add comma separated items	<p>(Mockup Custom List attributes only)</p> <p>Allows you to populate a mockup list field manually.</p>
Select the Lookup Entity	<p>(Mockup Lookup attributes only)</p> <p>Allows you to populate a mockup lookup field with a specific entity's primary attributes.</p>
Select the Option Set	<p>(Mockup Option Set attributes only)</p> <p>Allows you to populate a mockup option set field with a specific option set's values.</p>

3. Click **Apply changes** to save the field's properties.

4. Click **Save and reload** to save the form.

To modify an attribute field, select it in the main area and edit its properties from the right panel.

To move an attribute in the user interface or to delete it, see "[Moving and deleting UI elements](#)" on page 323.

### Custom UI Controls

Some attribute fields have a **UI Control** property which allows you to customize their appearance. Click the gear icon to select your customizations:

#### Custom UI Controls for Boolean Attributes (Non-Mockup)

- **Default** - Standard checkbox control.
- **Ftos Switch** - Toggle switch with the following options:
  - **Label** - Text label for the switch.
  - **Extra** - Optional text displayed between the label and the switch. The text is highlighted when the switch attribute value is set to True and grayed out when set to False
  - **On Text** - Text displayed on the switch when the attribute value is True.
  - **Off Text** - Text displayed on the switch when the attribute value is False.

Each option field has a **Translations** button that allows you to provide translations for each platform [language](#).

- **Ftos Button Switch Boolean** - A pair of radio buttons with the following options:
  - **Use Radios** - Displays a radio button selector inside each button.
  - **Label** - Text label for the attribute field.
  - **True Label** - Text displayed on the switch when the attribute value is True.
  - **False Label** - Text displayed on the switch when the attribute value is False.

Each option field has a **Translations** button that allows you to provide translations for each platform [language](#).

## Custom UI Controls for Numeric and Whole Number Attributes (Non-Mockup)

- **Default** - Standard text box control.
- **Ftos Slider** - Text box bound to a slider with the following options:
  - **Label** - Text label for the control. Includes a **Translations** button that allows you to provide translations for each platform [language](#).
  - **Icon** - HTML markup for additional visual elements such as formatted text or images. E.g.: `<i class="fa fa-percent" aria-hidden="true"></i>`
  - **Min** - Minimum slider value (leftmost position).
  - **Max** - Maximum slider value (rightmost position).
  - **Step** - Value increment/decrement between slider positions.
  - **Currency Code** - Allows you to display a currency symbol label based on the ISO 4217 standard.
  - **Input Width** - CSS-compatible width specification for the control.
- **Ftos Decorated Numeric** - Text box with customizations for additional visuals and adjustable width:
  - **Icon** - HTML markup for additional visual elements such as formatted text or images. E.g.: `<i class="fa fa-percent" aria-hidden="true"></i>`
  - **Input Width** - CSS-compatible width specification for the control.
- **Ftos Quantity Buttons** - Spin box with increment/decrement buttons (+/-):
  - **Label** - Text label for the control. Has a **Translations** button that allows you to provide translations for each platform [language](#).
  - **Step** - Value change per click on the +/- buttons.
  - **Input Width** - CSS-compatible width specification for the control.

**Custom UI Controls for Option Set Attributes (Non-Mockup)**

- **Default** - Standard drop down box control.
- **Ftos Button List** - Radio buttons with the following options:
  - **Use Radios** - Displays a radio button selector inside each button.
  - **Label** - Text label for the control. Includes a **Translations** button that allows you to provide translations for each platform [language](#).
  - **Items Extra Label** - Additional label for each option displayed after the "Display Name" of the option.

## Add relations to the UI

The relations element allows you to display and/or edit "[Data Views](#)" on [page 112](#) with entity records that reference the record currently displayed in the UI. The field is based on a view of the referencing entity filtered by the lookup attribute that points to the form's base entity to match only the current record.

For instance, you may have in your data model a Deposits entity that references the Customers entity (has a lookup attribute pointing to the Customers entity to indicate the customer that owns the deposit). If you display the Deposits-to-Customers relation on a Customers form, you will be able to see all the deposits that the customer owns.

Name

John Doe

Birth Date

30/01/1940

Enter any relevant information about the person's background below.

History

n/a

DEPOSITS

+ Insert

× Delete

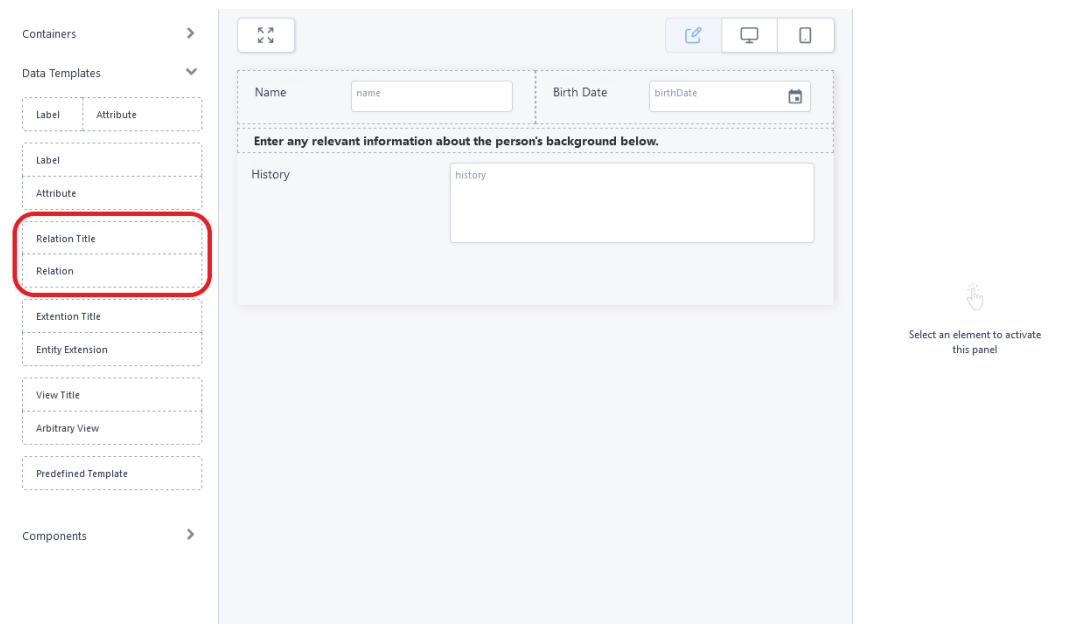
Export menu

Refresh

<input type="checkbox"/>	Account No.	Currency	Amount
	001	RON	250.00
	002	EUR	2,000.00

To add a relation element to the user interface:

1. In the **"UI Designer"** on [page 323](#)'s left panel, from the Data Templates section, select the relation template and drag it to the working area in the desired position. For more information, see **"Adding UI elements to the user interface"** on [page 323](#).



2. While the relation field is selected, fill in its properties in the right panel.

Property	Description
Select the Relation	Select the name of the relation that you want to display.
Edit Mode	<p>Makes the relation view editable and sets the mode of the inline editing on the view:</p> <ul style="list-style-type: none"> <li>• <b>cell</b> - allows you to edit view records cell by cell.</li> <li>• <b>row</b> - allows you to edit a view record by editing the cells in a row then saving the view record changes.</li> <li>• <b>batch</b> - allows you to edit several view records and then save the changes in batch.</li> </ul>
Collapse	Makes the relation field collapsible and sets the heading of the collapsible panel.
View	<p>The name of the view of the referencing entity to be displayed. If left unchecked, the default view is used.</p> <div> <p><b>IMPORTANT!</b></p> <p>The view must include the referencing entity's primary key and the lookup attribute pointing to the form's base entity.</p> </div>
Form	If the referencing entity has multiple forms and you want to use a specific data form for edits, select this checkbox and provide the form name. Otherwise, the entity's default edit form will be used.

Property	Description
Insert Form	If the referencing entity has multiple forms and you want to use a specific data form for inserts, select this checkbox and provide the form name. Otherwise, the entity's default insert form will be used.
No Header	Does not display the view header.
No Filter	Does not display the view filtering / search.
No Insert	Does not display the <b>Insert</b> button on the view toolbar.  <div> <p><b>NOTE</b></p> <p>Do not select <b>No Insert</b> when using <b>Form</b>. and <b>Insert Form</b>; otherwise, issues might occur.</p> </div>
No Delete	Does not display the <b>Delete</b> button on the view toolbar.
No Export	Does not display the <b>Export</b> button on the view toolbar.
No Refresh	Does not display the <b>Refresh</b> button on the view toolbar.
No Toolbar	Does not display the view toolbar.
Select Label Type	<ul style="list-style-type: none"> <li>• <b>default</b> - The attribute's display name is used as a field label.</li> <li>• <b>custom</b> - Manually enter the text for the field label. You can also manually add translations for each platform <a href="#">language</a>, by clicking the toggle translation button placed in the <b>Enter the desired text</b> field. In this case, you also need to choose a name for the resource key that will save this text in the <a href="#">Localization Resources</a> page.</li> </ul>

Property	Description
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets" on page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

3. Click **Apply changes** to save the field's properties.

4. Click **Save and reload** to save the form.

### IMPORTANT!

Do not intersect two relation layouts mixing them by moving relations rows up and down; otherwise, errors might occur.

To modify a relation field, select it in the main area and edit its properties from the right panel.

To move a relation in the user interface or to delete it, see ["Moving and deleting UI elements" on page 323](#).

## Add transient data entities to the UI

The extension element allows you to display ["Transient Data Entities" on page 150](#) with collection outputs that are part of the entity's extended data model. The field is based on a view of the transient data entity.

### IMPORTANT!

Before you add a transient data entity to the UI, you must:

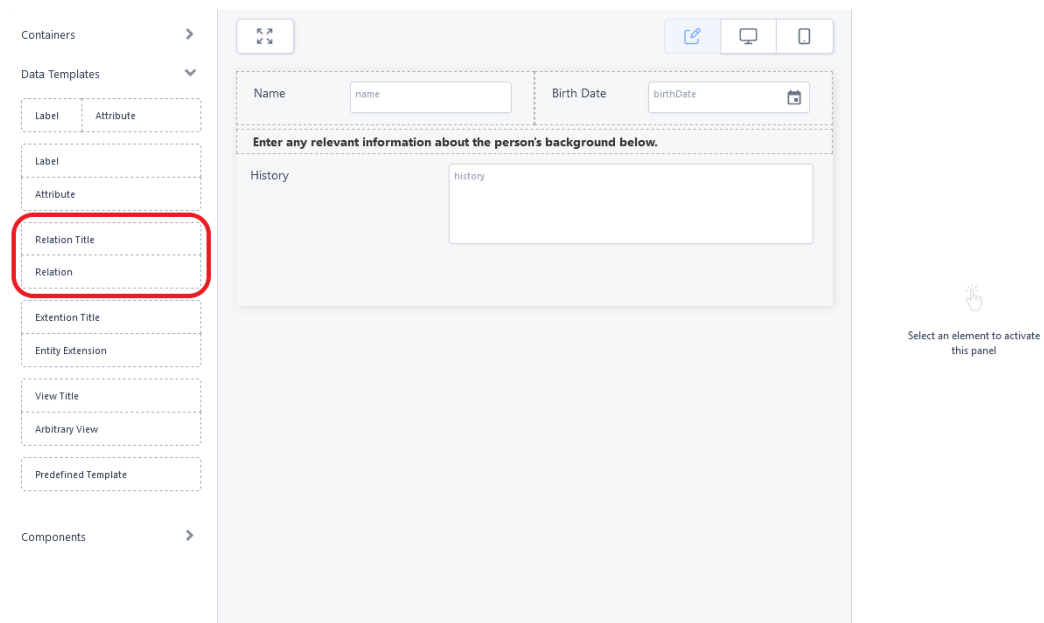
1. Define a transient data entity (see ["Create transient data entities" on page 150](#)) with a collection output (see ["Create On-demand Server](#)

[Automation Scripts" on page 1168](#)).

2. Extend the data model of the base entity with the transient data entity (see ["Extend platform data entities with transient data entities" on page 156](#)).
3. For ["App Data Forms" on page 933](#) and ["Form Driven Flows" on page 221](#), explicitly add the extension to the form's or flow's data model.

To add a transient data entity element to the user interface:

1. In the ["UI Designer" on page 323](#)'s left panel, from the Data Templates section, select the extension template and drag it to the working area in the desired position. For more information, see ["Adding UI elements to the user interface" on page 323](#).



2. While the extension field is selected, fill in its properties in the right panel.

Property	Description
Select the Entity Extension	Select the name of the data model extension that you want to display.
Collapse	Makes the extension field collapsible and sets the heading of the collapsible panel.
View	The name of the view of the transient data entity to be displayed. If left unchecked, the default view is used.
No Refresh	Does not display the <b>Refresh</b> button on the view toolbar.
No Toolbar	Does not display the view toolbar.
Select Label Type	<ul style="list-style-type: none"> <li>• <b>default</b> - The extension's display name is used as a field label.</li> <li>• <b>custom</b> - Manually enter the text for the field label. You can also manually add translations for each platform <a href="#">language</a>, by clicking the toggle translation button placed in the <b>Enter the desired text</b> field. In this case, you also need to choose a name for the resource key that will save this text in the <a href="#">Localization Resources</a> page.</li> </ul>
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated "<a href="#">Style Sheets</a>" on <a href="#">page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

3. Click **Apply changes** to save the field's properties.

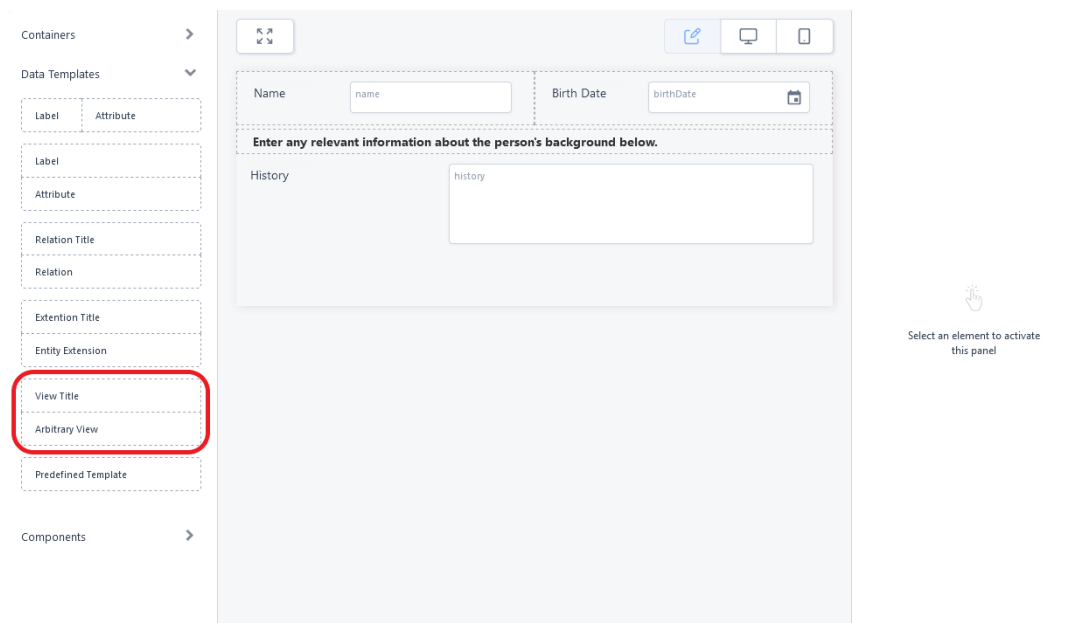
4. Click **Save and reload** to save the form.

To modify an extension field, select it in the main area and edit its properties from the right panel.

To move an extension in the user interface or to delete it, see ["Moving and deleting UI elements" on page 323](#).

## Add views to the UI

1. In the ["UI Designer" on page 323](#)'s left panel, from the Data Templates section, select the view template and drag it to the working area in the desired position. For more information, see ["Adding UI elements to the user interface" on page 323](#).



2. While the view field is selected, fill in its properties in the right panel.

Property	Description
Entity	Select the name of the entity the view belongs to.
View	The name of the view of the selected entity to be displayed. If left unchecked, the default view is used.

Property	Description
Grid Id	<p>Allows you to assign an ID suffix/ending to the HTML element that holds the grid view.</p> <p>This can be used as a discriminator if you want to display multiple views of the same entity with different behaviors.</p>
Collapse	Makes the field collapsible and sets the heading of the collapsible panel.
No Header	Does not display the view header.
No Filter	Does not display the view filtering / search.
Form	If the view entity has multiple forms and you want to use a specific data form for edits, select this checkbox and provide the form name. Otherwise, the entity's default edit form will be used.
Insert Form	If the view entity has multiple forms and you want to use a specific data form for inserts, select this checkbox and provide the form name. Otherwise, the entity's default insert form will be used.
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets"</a> on page 1209.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

3. Click **Apply changes** to save the field's properties.

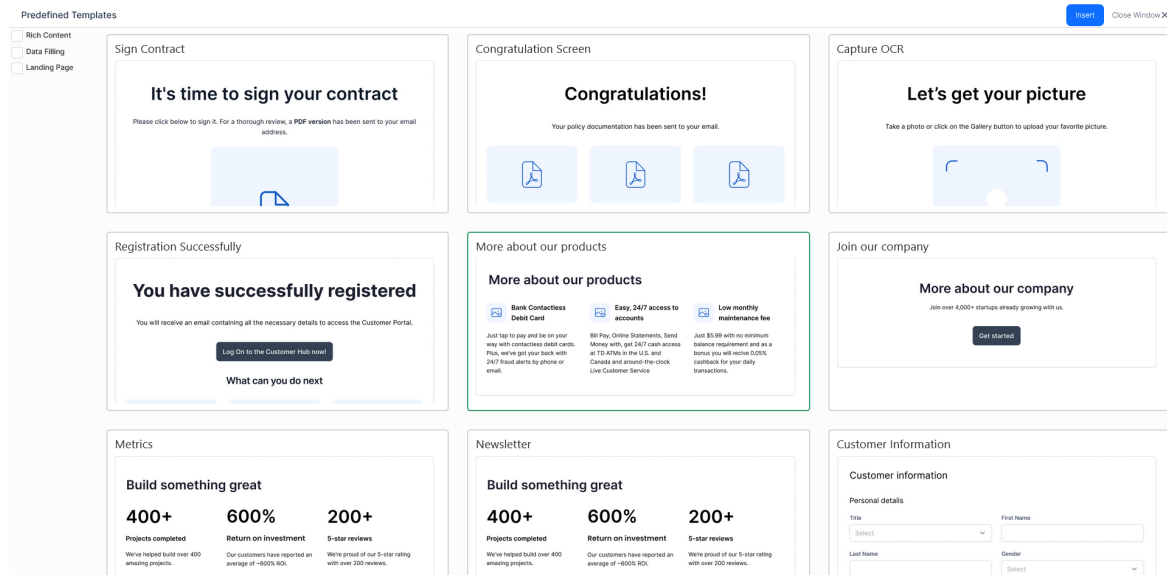
4. Click **Save and reload** to save the form.

To modify a view field, select it in the main area and edit its properties from the right panel.

To move a view in the user interface or to delete it, see ["Moving and deleting UI elements"](#) on page 323.

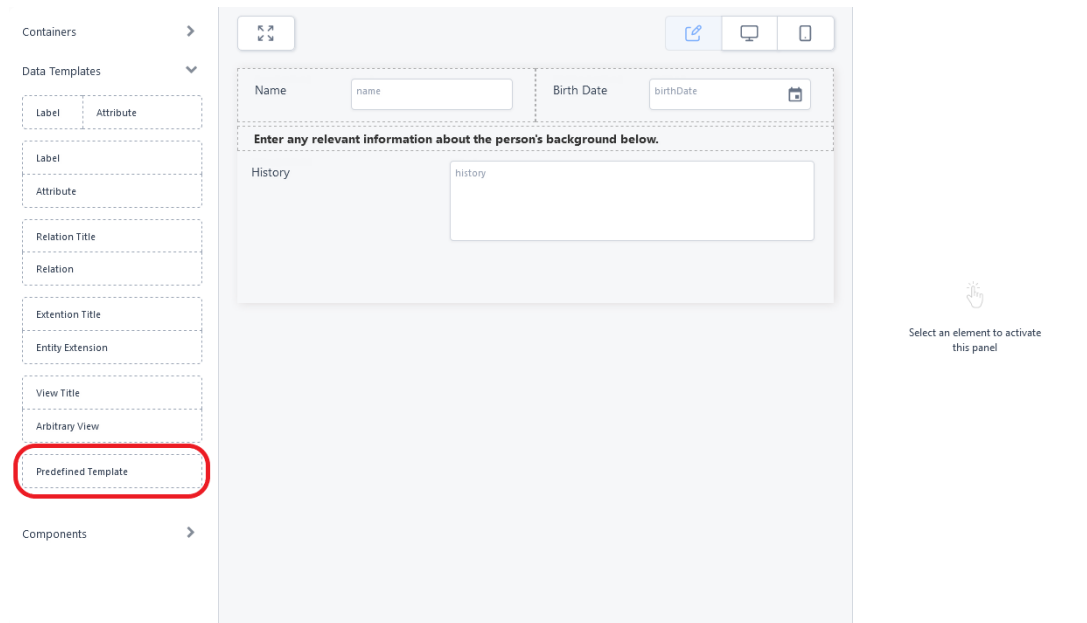
## Use predefined UI templates

The UI Designer comes with several preset UI templates for frequent use cases such as product information pages, contact forms, customer registration, identity validation, or landing pages, that you can use as a starting point in laying out the UI.



To add a predefined template to the user interface:

1. In the ["UI Designer"](#) on page 323's left panel, from the Data Templates section, select the Predefined Templates element and drag it to the working area in the desired position. For more information, see ["Adding UI elements to the user interface"](#) on page 323.



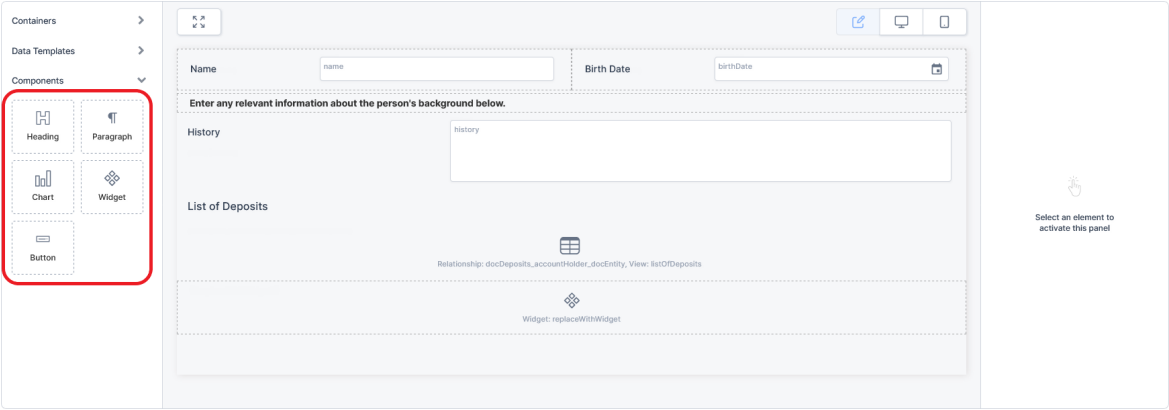
2. Select the desired template from the previewer and click **Insert**. You can use the checkboxes on the left to filter the available templates.
3. Customize the inserted UI elements according to your requirements. To modify a UI element, select it in the main area and edit its properties from the right panel.
4. Click **Save and reload** to save the form.

### HINT

You can import multiple templates in the same UI, then mix elements between them (see "[Moving and deleting UI elements](#)" on page 323).

## Use UI components (headings, paragraphs, charts, widgets, buttons)

The **Components** in the left panel of the UI designer allow you to add static or interactive elements that are not related to the form's data model to the user interface. These elements include headings, paragraphs, charts, widgets, and buttons.



To add a component to the interface, drag and drop it from the left panel to the editor's working area (see ["Adding UI elements to the user interface"](#) on page 323).

Headings

Headings separate and label the various sections in your user interface. To add a heading to the user interface:

- 1. From the left panel, drag the heading component to the desired position in the UI.
- 2. Use the right panel to configure the heading's properties:

3.

Property	Description
Select Heading Type	Select the type of heading from H1 through H5.
Enter the desired text	Enter the text for the field heading. You can also add translations for each platform <a href="#">language</a> , by clicking the toggle translation button .
Enter the resource key name that will store the localizable text	Choose a name for the resource key that will save this label in the <a href="#">Localization Resources</a> page.
Select the text alignment	Select to align the heading left, center, or right.

Property	Description
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets" on page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

Click **Apply changes**.

4. Click **Save and reload**.

## Paragraphs

Paragraphs are static blocks of text that describe your user interface. To add a paragraph:

1. From the left panel, drag the paragraph component to the desired position in the UI.
2. Click the paragraph to insert the desired text. A pop-up toolbar allows you to format the text.



3. Click **Apply changes**.
4. Click **Save and reload**.

## Widgets

To add ["Widgets" on page 370](#) to the user interface:

1. From the left panel, drag the widget component to the desired position in the UI.
2. Use the right panel to configure the widget's properties:

Property	Description
Select the Widget	Select the desired widget you wish to display.
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets" on page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>
Widget identifier	Allows you to assign an ID to the HTML element where the widget is displayed.
Hide widget title	Does not display the widget title.

3. Click **Apply changes**.

4. Click **Save and reload**.

## Charts

To add ["Charts" on page 1019](#) to the user interface:

1. From the left panel, drag the chart component to the desired position in the UI.

2. Use the right panel to configure the chart's properties:

3.

Property	Description
Select the Chart	Select the desired chart you wish to display.

Property	Description
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets" on page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

Click **Apply changes**.

4. Click **Save and reload**.

## Buttons

To add a button to the user interface:

1. From the left panel, drag the button component to the desired position in the UI.
2. Use the right panel to configure the button's properties:

3.

Property	Description
	Behaviour

Property	Description
Select the Button type	<ul style="list-style-type: none"> <li>Custom - Button that you configure by writing your own code. When you create a custom button, a boilerplate event handler code is created in the After Events section of the Advanced tab for the button's click events. <div data-bbox="630 506 1369 785" data-label="Text"> <pre>/* Click event for the testButton button */ \$('#testButton').on('click', function (event) {     console.log("The button testButton was clicked"); });</pre> </div> </li> <li>Call Custom Processor - Button that calls predefined "Endpoints" on page 1213.</li> <li>Call Form Action - Button that calls a form action (see "Form Actions" on page 258 and "Define Form Actions" on page 949 for reference).</li> <li>Notify Event - Button that triggers a custom event.</li> </ul> <div data-bbox="552 1182 1369 1640" data-label="Text"> <p><b>NOTE</b></p> <p>When removing a custom button from the template, the corresponding on-click event in the <b>After Events</b> tab will not be removed, as it might be useful in case you accidentally delete the button.</p> <p>Also, the localization resource key is not erased, as you still might need the translations.</p> </div>

Property	Description
Fill in the Button Text	<p>Enter the text for the button label.</p> <p>You can also add translations for each platform <a href="#">language</a>, by clicking the toggle translation button.</p>
Fill in the Button ID	Enter a HTML ID for the button.
Resource key name	<p>Name of the resource key that designate the the button label localizations in the <a href="#">Localization Resources</a> page.</p> <p>This key is generated automatically based on the button ID.</p>
Custom Processor (Call Custom Processor buttons only)	Select the endpoint that will be called when clicking the button.
Select the Form Actions (Call Form Actions buttons only)	Select the form action that will be called when clicking the button.
Navigate to Next Step (Call Form Actions buttons only)	If the action is triggered, and the result is true, trigger the navigation to the next step.

Property	Description
Fill in the Form Actions Message (Call Form Actions buttons only)	Once the action is triggered, display this message.
Fill in the Custom Event Name (Notify Event buttons only)	Name of the custom event you wish to trigger. <div> <b>NOTE</b>            Custom events are configured using the <a href="#">formData.registerEvent</a> Server SDK method.         </div>
Appearance	
Select the Color	Choose a color for the button.
Style	Choose the button styling: <ul style="list-style-type: none"> <li>• Block - The button occupies the entire width of its container.</li> <li>• Filled - The button color is applied to the entire button, not just the edges.</li> <li>• Round - The margins of the button are rounded.</li> <li>• Simple - The button shape is not visible, only its text label is displayed. Not compatible with Filled and Round.</li> </ul>
Select the Size	Select the size of the button label's text.

Property	Description
Select the Icon	Select an icon to be displayed before the button label.
Additional CSS classes	<p>Write any CSS class names you wish to apply to the field. The classes must be defined in one of the form's associated <a href="#">"Style Sheets" on page 1209</a>.</p> <div> <p><b>NOTE</b></p> <p>You must press <b>Space</b> or <b>Enter</b> after typing a class name to confirm it.</p> </div>

Click **Apply changes**.

- Click **Save and reload**.

## Example: Creating a Search Customer button

This example describes how to configure a **Search Customer** button on a simulated digital journey for unsecured loan applications. This enables agents to check if a customer applying for a loan is an existing customer and if so, get the customer's information from the database and auto-fill in specific customer information. This example presents the button's click event handler.

```
// Search Customer by PIN
$('#getPersonalData').click(
  function() {
    var PIN = $("#inputSearchPIN").val();
    ebs.getByQuery(
      {
        "entity": {
          "alias": "a",
          "name": "Account",
          "attributelist": [
            {"name": "Accountid"},
            {"name": "LastName"}
          ]
        }
      }
    )
  }
)
```

```

        {"name": "FirstName"},
        {"name": "UniqueID"},
        {"name": "DateOfBirth"},
        {"name": "PlaceOfBirth"},
        {"name": "Age"},
        {"name": "IdCardNo"},
        {"name": "IdCardSeries"},
        {"name": "IdCardIssuedBy"},
        {"name": "IdCardIssueDate"},
        {"name": "IdCardExpiryDate"},
        {"name": "GenderId"}
    ],
    },
    "where": {
        "type": "and",
        "conditionlist": [
            {
                "first": "a.PIN",
                "type": "equals",
                "second": "val(" + PIN + ")"
            }
        ]
    }
}, function(e){
    if(e.Records.length > 0){
        ebs.setFormAttributeValue("ebsContainerContent",
"LastName", e.Records[0].a_LastName);
        ebs.setFormAttributeValue("ebsContainerContent",
"FirstName", e.Records[0].a_FirstName);
        ebs.setFormAttributeValue("ebsContainerContent",
"PIN", e.Records[0].a_UniqueID);
        ebs.setFormAttributeValue("ebsContainerContent",
"BirthDate", e.Records[0].a_DateOfBirth);
        ebs.setFormAttributeValue("ebsContainerContent",
"BirthPlace", e.Records[0].a_PlaceOfBirth);
        ebs.setFormAttributeValue("ebsContainerContent",
"Age", e.Records[0].a_Age);
        ebs.setFormAttributeValue("ebsContainerContent",
"IDNumber", e.Records[0].a_IdCardNo);
        ebs.setFormAttributeValue("ebsContainerContent",
"IDSeries", e.Records[0].a_IdCardSeries);
        ebs.setFormAttributeValue("ebsContainerContent",
"IDIssuedBy", e.Records[0].a_IdCardIssuedBy);
        ebs.setFormAttributeValue("ebsContainerContent",
"IDValidFrom", e.Records[0].a_IdCardIssueDate);
    }
}

```

```

        ebs.setFormAttributeValue("ebsContainerContent",
        "IDValidTo", e.Records[0].a_IdCardExpiryDate);
    }
    else{
        ebs.showMessage("The customer was not found in
        our database. Please proceed to OCR.", "warning");
        ebs.setFormAttributeValue('ebsContainerContent',
        'LastName', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'FirstName', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'BirthPlace', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'BirthDate', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'Age', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'BirthDate', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'IDNumber', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'IDSeries', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'IDIssuedBy', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'IDValidFrom', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'IDValidTo', null);
        ebs.setFormAttributeValue('ebsContainerContent',
        'PIN', $("#inputSearchPIN").val());
    }
    });
});

```

To check if the customer is an existing customer, the agent will provide the PIN and click the **Search Customer** button. If the PIN exists in the database, the customer's Personal Data is automatically filled-in.

## Code Snippets Support

Code snippets are small sequences of reusable code that can be inserted using a combination of shortcut keys. They are very useful as they make it easier to remember specific formatting of functions and avoid spending time on typos and syntax errors.

Code snippets are an aid of type IntelliSense/auto-complete and most of the times they are incomplete and / or syntactically incorrect and require further processing thereof.

Code snippets support is available via the Monaco Intelisense for text boxes and controls of JavaScript. and for the HTML editor.

### NOTE

Backward compatibility is not necessary, so the snippets, their name, their content and the way they are organized in drop-downs may change from one FintechOS Studio release to another without prior notice.

## Code Snippets Support for the HTML Editor

Code snippets are also available in the HTML editor available on forms and user journeys, via the Monaco Editor.

In order to use these code snippets on the data form or digital journey, follow the steps below:

1. Go to the **Code** section, **Template** tab.
2. From the HTML editor toolbar, click **Tools** and select **Source code**. The Source code page displays using the Monaco Editor.
3. Press the **CTRL+Space** keys to open a drop-down which contains not only entries corresponding to certain code snippets, but also attributes and relations belonging to the entity.

## Code Snippets Placeholders

The code snippets might contain hint words, known as placeholders which can be easily selected and changed. The placeholders are highlighted with a background color.

### Navigating placeholders

Upon snippet insertion, the cursor will be placed on the first placeholder (if any). If no placeholder is available, the cursor will go to the end of the snippet.

To navigate from one placeholder to another, press the **TAB** key or **SHIFT+TAB**.

Pressing **TAB** when the cursor is over the last placeholder will move the cursor at the end of the snippet.

### Replacing placeholders

There might be placeholders without underlying text, visible due to a narrow gray line on that position.

When the cursor is over a placeholder, starting to type will remove the placeholder text.

#### **IMPORTANT!**

Replace all placeholders, otherwise errors might occur as the code snippet syntax might be broken or the syntax remains unbroken but it is logically incorrect.

The presence of one or more cursors indicate that those placeholders will be filled in simultaneously with the same text. Additionally, such group of placeholders might have identical texts beneath.

Make sure to replace the placeholders and click **OK** to save the source code changes.

### Deactivating placeholders

When the cursor falls outside the snippet, the placeholders will be deactivated and any text which has not been replaced will remain in the snippet.

When the placeholders are deactivated:

- They are no longer highlighted with a background color.
- Navigating from one placeholder to another is no longer possible using **TAB** and **SHIFT+TAB**.
- Typing when cursor is on top of the placeholders will not remove the placeholder text.

### Series of placeholders

Some snippets may have for some fields' series of placeholders in a row, like: true false

To eliminate the unwanted variants and keep the desired one, navigate placeholders using **TAB** or **SHIFT+TAB** and when on top of the placeholder that you want to eliminate, press the **delete** key.

### Nested code snippets

Snippets might be nested, meaning that in a placeholder you can insert another snippet, without deactivating the placeholders of the first snippet.

### Examples of code snippets

Pressing **CTRL+Space** and selecting **attribute rows by 2** from the drop-down will insert the following code snippet:

```
<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12" >
    <div class="row">
      <div class="col-lg-4 col-md-4 col-sm-4 col-xs-12
data form-label"></div>
      <div class="col-lg-8 col-md-8 col-sm-8 col-xs-
12"></div>
    </div>
  </div>
```

```

</div>
<div class="col-lg-6 col-md-6 col-sm-6 col-xs-12" >
  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-4 col-xs-12
data form-label"></div>
    <div class="col-lg-8 col-md-8 col-sm-8 col-xs-
12"></div>
  </div>
</div>
</div>

```

Pressing **CTRL+Space** and selecting `<=""></>` from the drop-down will insert the following code snippet:

```
<tag = " ">text</tag>
```

**NOTE** In cases like the previous example, two or more placeholders might be written simultaneously, more than one cursor being available.

Examples of code snippets for attributes and relations.

- Pressing **CTRL+Space** and selecting `{#relView#} ebs_AAA_Ent_8966_businessunit` will insert `{#ebs_AAA_Ent_8966_businessunit#}`
- Pressing **CTRL+Space** and selecting `{attrLabel|attribute} businessunitid` will insert `{businessunitid|attribute}`
- Pressing **CTRL+Space** and selecting `{attrName} businessunitid` will insert `{businessunitid}`
- Pressing **CTRL+Space** and selecting `{relName|entities} ebs_AAA_Ent_8966_businessunit` will insert `{ebs_AAA_Ent_8966_businessunit|entities}`

## Code Snippets Support for JavaScript

Using Monaco IntelliSense available via the Monaco Editor, the platform provides two mechanisms for using code snippets in Java Script text boxes, as follows:

- **\$s.** - displays code snippets
- **\$m.** - displays code snippets for entities and attributes.

The sections below describe how to use the two mechanisms.

## Code Snippets

Writing **\$s.** opens a drop-down list which offers the possibility to choose a snippet name or another. There might be one or more entries in the list which are not snippets but group names in which case typing a dot after them will open new drop-down menu.

The snippets can be organized in menus or drop-downs on multiple levels in structures like hierarchies.

A snippet name together with its selection path may look like:

```
$s.ebs.functions.callAction
```

Placing the cursor right at the end of the snippet path/name (full name) and pressing the **TAB** key, the system will replace the string with:

```
ebs.callAction( actionId, { id:ebs.getCurrentEntityId() },
function( e ) { function body; });
```

**NOTE** If you do not place the cursor at end of the snippet path / name (full name) before pressing the **TAB** key, the snippet name will be breaking syntax.

In the example provided above, “ebs.” is inserted in front of the function to reduce further typing.

## Code Snippets Placeholders

The code snippets might contain hint words, known as placeholders which can be easily selected and changed. The placeholders are highlighted with a background color.

In the example provided above, the **ebs.callAction** function snippet, “actionId” and “function body”, although not visible in the text above, might be placeholders.

### Navigating placeholders

Upon snippet insertion, the cursor will be placed on the first placeholder (if any). If no placeholder is available, the cursor will go to the end of the snippet.

To navigate from one placeholder to another, press the **TAB** key or **SHIFT+TAB**.

Pressing **TAB** when the cursor is over the last placeholder will move the cursor at the end of the snippet.

### Replacing placeholders

There might be placeholders without underlying text, visible due to a narrow gray line on that position.

When the cursor is over a placeholder, starting to type will remove the placeholder text.

**IMPORTANT!** Replace all placeholders, otherwise, errors might occur as the code snippet syntax might be broken or the syntax remains unbroken but it is logically incorrect.

The presence of one or more cursors indicate that those placeholders will be filled in simultaneously with the same text. Additionally, such group of placeholders might have identical texts beneath.

Make sure to replace the placeholders and click **OK** to save the source code changes.

### Deactivating placeholders

When the cursor falls outside the snippet, the placeholders will be deactivated and any text which has not been replaced will remain in the snippet.

When the placeholders are deactivated:

- They are no longer highlighted with a background color
- Navigating from one placeholder to another is no longer possible using **TAB** and **SHIFT+TAB**.
- Typing when cursor is on top of the placeholders will not remove the placeholder text.

### Series of placeholders

Some snippets may have for some fields' series of placeholders in a row, like:  
left inner right

To eliminate the unwanted variants and keep the desired one, navigate placeholders using **TAB** or **SHIFT+TAB** and when on top of the placeholder that you want to eliminate, press the **delete** key.

### Nested code snippets

Snippets might be nested, meaning that in a placeholder you can insert another snippet, without deactivating the placeholders of the first snippet.

### Code snippets for entities and attributes

The **\$m** mechanism has been introduced to enable FintechOS Platform engineers to easily select entities and attributes names, transforming them in strings if necessary.

Writing **\$m.** opens a drop-down list containing existing entities. Selecting an entity and typing a dot opens a new drop-down containing the attributes of that entity. Pressing **TAB** after the inserted entity or attribute transforms the expression into a string which contains only the name of the entity or attribute depending on the selection(s) made from drop-down(s).

If selecting the current entity from the drop-down, data extensions (if any will also be displayed) in the drop-down list.

- Pressing **TAB** after **\$m.Account.Accountid** transforms the expression into "Accountid"

- Pressing **TAB** after `$m.Account` transforms the expression into "Account"

To eliminate the quotes around the string, press **TAB** a second time .

**NOTE** If there is a dot before “\$m.” (**.\$m.**), the expression is not transformed into a string, keeping the selected name.  
Pressing **TAB** after `.$m.Account.Accountid` transforms the expression into **.Accountid**

# Digital Frontends

FintechOS Studio enables you to define every interaction that your business has with your internal team as well as with the customers. Broadly defined, digital frontends represent your user experience.

Properly defined digital frontends allow you to keep customers happy and loyal while ensuring higher efficiency within your organization.

This section covers the following topics:

Dashboards .....	362
Widgets .....	370
Menu Items .....	371
Custom Themes .....	373
Portal Profiles .....	377
B2C Portals .....	381
Login Screen Themes .....	388
Anonymous Frontends (deprecated) .....	391
Digital Experience Portals .....	402

## Dashboards

Dashboards are customizable panels you can display in the Portal interface that aggregate entity views, reports, charts, shortcuts to frequently used forms, and other custom elements on the same page. These elements are displayed in a grid layout using widgets, which you can customize, position, and resize inside the dashboard at will.

To create a dashboard, follow these steps:

### 1 Create a Dashboard

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Dashboards**.
2. Click **Insert**. The dashboard configuration page appears by default on the General tab.
3. In the **Name** field, provide the name of the dashboard that will be used by the system.
4. In the **Display Name** field, provide the name of the dashboard that will be displayed in the Digital Experience Portals.
5. In the **Widget Vertical Spacing** field, provide the amount of vertical space to be added between the dashboard's elements. Default value is 20. The vertical spacer automatically adjusts with the screen size your page is viewed on.
6. In the **Widget Horizontal Spacing** field, provide the amount of horizontal space to be added between the dashboard's elements. Default value is 20. The horizontal spacer automatically adjusts with the screen size your page is viewed on.
7. Tick the **Show On Home Page** checkbox if you want the dashboard to be displayed on the Portal home page. If the checkbox is false, then the dashboard and everything placed in its layout will not be rendered in the Portal.

8. Click **Save and Reload**. The page refreshes and the dashboard is displayed in the Dashboards List page.

You can add as many dashboards as you need. The **Order Index** column specifies the order in which dashboards appear in the **Change Dashboard** drop-down menu on the Portal home page. You can change the order index of a record by using drag and drop.

You can now add widgets to your dashboard.

## 2 Add Widgets to Dashboards

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Dashboards**.
2. Double-click the desired dashboard from the list to open the dashboard configuration page.
3. In the **Add Widget** area, from the widget type drop-down list, select the desired widget type.

4. Select the widget to be added. You can choose from multiple types of widgets:

Widget	Description
PowerBI Report	Widget based on PowerBI reports. For details, see <a href="#">"Advanced Analytics"</a> on page 1026.
HTML Widget	Fully customizable <a href="#">"Widgets"</a> on page 370 based on your own HTML and JavaScript code.
Shortcut	Button that allows you to open entity views, entity insert forms, or <a href="#">"Custom Flows"</a> on page 283.
Entity View	Displays an entity view on the dashboard.
Charts	Allows you to display <a href="#">"Charts"</a> on page 1019.

5. Click **Add**. The selected widget is displayed on the left side of the page.

### Customize Widgets

After the widget is added to the dashboard, you can further customize its appearance and behavior from the widget editor. To do so, click the widget rectangle and fill in the corresponding fields (depending on the type of widget, some fields may not be available):

Edit Widget

Customize Shortcut

Title

LookupPrerequisites

Description

Insert lookup values

Tag

LookupPrerequisites

Custom CSS Class

Visibility endpoint

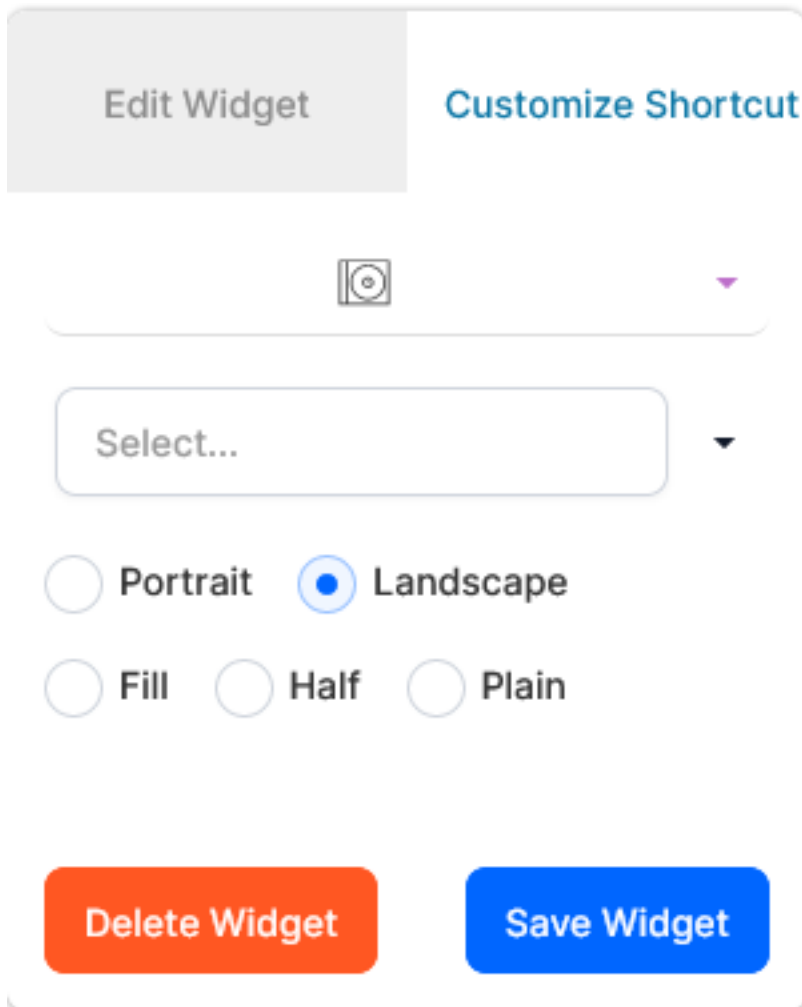
Select visibility endpoint

Delete Widget

Save Widget

Field	Description
Title	Name of the widget as displayed on the dashboard.
Description	Explanation of what the widget does.
Tag	Small label displayed in the corner of the widget. You can use it to classify widgets into categories.
Custom CSS Class	CSS class used to apply styling for your widget. The custom style sheets that contain the class definitions must be imported in your widget's underlying component, depending on the widget type (see <a href="#">"Apply Style Sheets Using Code" on page 1212</a> ). For instance, a shortcut widget that opens a custom flow, will need to have the style sheet imported in the After Generate Js section of the flow.
Visibility endpoint	Allows you to use on-demand server automation scripts with boolean output (see <a href="#">"Create On-demand Server Automation Scripts" on page 1168</a> ) to dynamically enable or disable the visibility of the widget on the dashboard. If the script returns true, the widget will be visible, if the script returns false, the widget will be hidden.

Shortcut widgets have an additional tab in the widget editor called **Customize Shortcut** which allows you to customize its appearance (icon, background color, form factor, and design).



The screenshot shows a configuration panel for a widget. At the top, there are two tabs: 'Edit Widget' (highlighted in grey) and 'Customize Shortcut' (in blue text). Below the tabs is a small icon of a widget with a dropdown arrow. Underneath is a 'Select...' dropdown menu. Below the menu are three rows of radio button options: 'Portrait' and 'Landscape' (selected with a blue dot), 'Fill', 'Half', and 'Plain'. At the bottom are two large buttons: 'Delete Widget' (orange) and 'Save Widget' (blue).

When done, click **Save Widget**.

### Resize Widgets

You can resize dashboard widgets by placing the mouse cursor on the bottom-right corner of the widget that you want to resize and dragging the corner to adjust the size as preferred.

### Move Widgets

You can arrange widgets on the dashboard by dragging it with the mouse to the desired position on the dashboard grid.

## Delete Widgets

To remove a widget from a dashboard, in the widget editor, click the **Delete Widget** button.

## 3 Attach Security Roles to Dashboards

Your business case may require the dashboard to be available only to users with designated security roles within your organization. To configure this:

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Dashboards**.
2. Double-click the desired dashboard from the list to open the dashboard configuration page.
3. Open the **Security Roles** tab and add the security roles that should have access to the dashboard. If no security roles are added here, all users will be able to view the dashboard in the Digital Experience Portals.
4. Click **Save and reload**.

## 4 Add Dashboards to Portal Profiles

To make a dashboard visible only on FintechOS Portal instances that use a specific portal profile:

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Dashboards**.
2. Double-click the desired dashboard from the list to open the dashboard configuration page.
3. Open the **Portal Profiles** tab.
4. Click **Insert** at the top of the Dashboards on Portal Profiles section.
5. Select the **Portal Profile** on which the current dashboard will be available.

Add Dashboard On Portal Profile

Dashboard: Test ↓ Portal Profile: MB\_PortalProfile ↓

Disabled: ☐

6. Click the **Save and close** icon. The **Add Dashboard on Portal Profile** page closes and the record is displayed in the **Dashboards on Portal Profiles** section.

1 General 2 Security Roles 3 Portal Profiles

Dashboards On Portal Profiles

+ Insert X Delete Export Refresh

Dashboard	Portal Profile	Disabled
Test	MB_PortalProfile	<input type="checkbox"/>

You can repeat the process to associate the dashboard to additional portal profiles.

### HINT

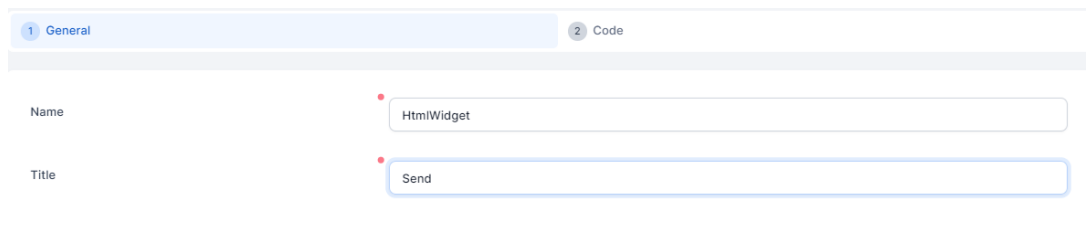
You can also assign dashboards to portal profiles from the ["Portal Profiles" on page 377](#) page. This is useful if you wish to assign multiple dashboards to the same portal profile.

## Widgets

HTML Widgets are custom components you can include in your Dashboards. With full control over the HTML and JavaScript code, you can fully customize the look, feel and function of HTML widgets. This allows you to expand your dashboards beyond the out-of-the-box features provided by PowerBI reports, shortcuts, entity views, or charts.

To create an HTML widget:

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Widgets**.
2. Click **Insert**. The wizard configuration page opens by default on the **General** tab.
3. In the **Name** field, provide the name of the HTML widget to be used by the system.
4. In the **Title** field, provide the widget title to be displayed in the Digital Experience Portals.



The screenshot shows the 'General' tab of the HTML Widget configuration wizard. It has two tabs: '1 General' (active) and '2 Code'. Below the tabs are two input fields. The first field is labeled 'Name' and contains the text 'HtmlWidget'. The second field is labeled 'Title' and contains the text 'Send'. Both fields have a red dot to their left, indicating they are required.

5. Click the **Code** tab.
6. Provide the widget's HTML code in the **Html** tab and the JavaScript code in the **JavaScript** tab.
7. Click **Save and close**.

The widget is saved and is listed the **Widgets List** page.

Now you can add the HTML widget to a dashboard (for more information, see [Adding Widgets to Dashboards](#)).

## Menu Items

You can customize the menu displayed in the FintechOS Portal instances to include shortcuts to entity lists, custom forms, and reports. To better organize your menu, you can group multiple menu items into menu sections.

### Create Menu Items

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Menu Items**.
2. Click **Insert**.
3. Fill in the following fields (depending on the type of menu item, some fields may not be available):

- **Type** - The type of menu item

Item Type	Description
Entity	Displays a view of a business entity.
Custom Journey	Opens a custom flow.
Report	Displays a report.
Menu Section	Creates a container menu item where you can group other menu items.

- **Business Entity** (for Entity menu items) - Business entity that the menu item displays.
- **Custom Journey** (for Custom Journey items) - Custom flow that the menu item displays.
- **Report** (for Report items) - Report that the menu item displays.
- **Display Name** - Name of the menu item as it is displayed in the user interface.

- **Parent Menu Item** (for Entity, Custom Journey, and Report menu items) - If you have already created a menu section where you wish to include the menu item, you can select it here.
  - **Entity View** (for Entity menu items) - The entity view that is displayed when opening the menu item. If not filled in, the entity's default view is used.
  - **Use Menu Display Name as View Title** (for Entity menu items) - Overrides the entity view's display name with the menu item's display name when opening the menu item.
  - **Insert Entity Form** (for Entity menu items) - Entity form used to insert new entity records. If not filled in, the entity's default insert form is used.
  - **Edit Entity Form** (for Entity menu items) - Entity form used to edit entity records. If not filled in, the entity's default edit form is used.
  - **Icon Url** - Icon for the menu item.
  - **Color** (for Menu Section menu items) - Color for the menu section's items.
  - **Disabled** - Tick if you wish to remove the menu item without deleting it, for instance if you wish to disable it temporarily.
4. Click **Save and reload**.
  5. If you wish the menu item to be available only for specific security roles, add them to the **Security Roles** grid. If the grid is left empty, only users with the Admin security role will be able to access the menu item.

**NOTE**

The security role must also have CRUD privileges assigned for the component (entity, form, etc.) that the menu item opens.

6. In Menu Section menu items, you can use the **Menu Items Children** grid to add new menu items to the section or to edit its existing menu items.

## Manage Menu Items in Menu Sections

When creating an Entity, Custom Journey, or Report menu items, if you don't select a parent, the menu item displays at the first level of the menu. Once you assign a menu item to a menu section, you are no longer be able to access it from the menu items list. You must open its menu section and access it from the Menu Items Children grid.

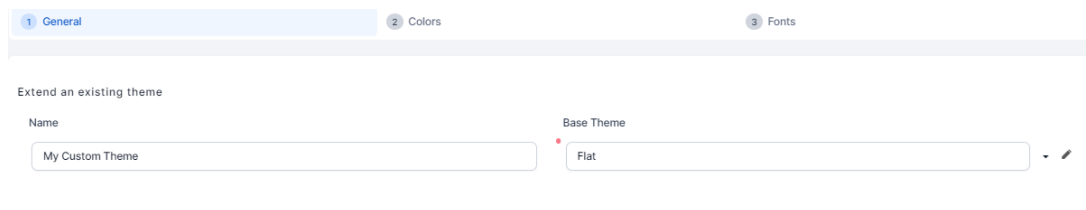
You cannot return a menu item to the first menu level once it has been assigned to a menu section. You can only assign it to a different menu section by selecting a different parent for it.

## Custom Themes

Custom themes allow you to adjust the default fonts and color palette of your FintechOS Portal user interface.

### Create a Custom Theme

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Custom Themes**.
2. Click **Insert**. The theme configuration page opens on the General tab.



The screenshot shows the 'General' tab of the 'Extend an existing theme' configuration page. At the top, there are three tabs: '1 General' (active), '2 Colors', and '3 Fonts'. Below the tabs, the text 'Extend an existing theme' is displayed. There are two input fields: 'Name' with the value 'My Custom Theme' and 'Base Theme' with the value 'Flat'. A red asterisk is visible next to the 'Base Theme' label. A small edit icon is located to the right of the 'Base Theme' field.

3. Provide a **Name** for your custom theme.
4. Select the **Base Theme** that your custom theme will extend.
5. Click **Save and Reload**.
6. Open the **Colors** tab.

7. In the **Match the new state colors** section, set the matching colors for different system notification messages. Read the on-screen information on the recommended colors.

Match the new state colors

\* info, success, warning, error classes will be generated having the following colors

Info Color  
(We suggest a blue color)

Success Color  
(We suggest a green color)

Warning Color  
(We suggest a yellow/orange color)

Error Color Value  
(We suggest a red color)

Blue

Green

Orange

Red

8. In the **Add the new colors** section, you can customize 16 colors from the base theme previously selected by providing the desired color hexadecimal values.

1 General 2 Colors 3 Fonts

Add the new colors

\* State colors (second input in a color group) are used in the FinTechOS Design Engine to animate an element's different states. Ex: hover, active

Color Purple #9b59b6	Color Blue #34495e	Color Azure #1abc9c
Color Green #2ecc71	Color Orange #e67e22	Color Red #e74c3c
Color Pink #f08f90	Color Gray #95a5a6	

#8e44ad #2c3e50 #16a085 #27ae60 #d35400 #c0392b #f47983 #7f8c8d

9. In the **Generate the new palette colors** section, select three colors of the 16 colors you added earlier. Based on these colors, a palette will be generated to the new custom theme.

Generate the new palette colors

Palette First Color

Palette Second Color

Palette Third Color

Purple

Pink

Gray

10. Once you select the matching colors for the different system notifications, a new section is displayed at the bottom of the configuration page and lists the new CSS color classes.

New CSS colors classes			
Color1 - Info Color	Color2 - Success Color	Color3 - Warning Color	Color4 - Error Color
#34495e	#2ecc71	#e67e22	#e74c3c
#2c3e50	#27ae60	#d35400	#c0392b
Color5	Color6	Color7	Color8
#9b59b6	#1abc9c	#f08f90	#95a5a6
#8e44ad	#16a085	#f47983	#7fbc8d

\* State colors are used in the FintechOS Design Engine to animate an element's different states. Ex: hover, active

## NOTE

The colors listed in the **New CSS color classes** section are non-editable. If you want to modify them, you can do so from the **Match the new state colors** section.

11. If you wish to customize the default fonts:

- Open the **Fonts** tab.
- Expand the **Custom Font** field and select the desired custom font. If no custom fonts have been defined or you wish to use a different font, click **Insert** and add a **Name** for your custom font, then upload font files for the various font weights.

Add Custom Font

Custom Font

Name

Thin Font

Add file or Drop file here

Thin Italic Font File

Add file or Drop file here

Light Font

Add file or Drop file here

Light Italic Font

Add file or Drop file here

Regular Font

Add file or Drop file here

Regular Italic Font

Add file or Drop file here

Bold Font

Add file or Drop file here

Bold Italic Font

Add file or Drop file here

Black Font

Add file or Drop file here

Black Italic Font

Add file or Drop file here

HINT

For UI consistency, try to use fonts from the same font family.

12. Save the custom theme by clicking the save icons at the top-right corner of the page.

## Apply the Custom Theme to a FintechOS Portal Instance

## To apply a custom theme to a FintechOS Portal instance

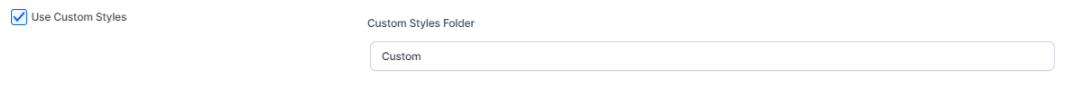
1. Open the [Configuration Manager](#).
2. Navigate to the `kv/<environment name>/<portal instance>/app-features` node.
3. Set the `feature.customTheme` secret to the name of your custom theme.

## Apply a Custom CSS Style Sheet to the Portal UI

1. Save the CSS file which contains the custom visual design and layout of the user interface on the server where the FintechOS Platform installation package is located, in

the **custom** folder.

2. In FintechOS Studio, go to **Main Menu > Admin > Settings** . The Portal UI configuration page will be displayed on the **General** tab.
3. Tick the **Use Custom Styles** checkbox and type the name of the folder.



The screenshot shows a configuration interface. On the left, there is a checkbox labeled 'Use Custom Styles' which is checked. To its right, there is a text input field labeled 'Custom Styles Folder' containing the text 'Custom'.

### NOTE

If you're using a custom CSS style sheet for the Portal UI, the options to choose the theme and palette will no longer be available in the Portal's top-right **Settings** menu.

## Portal Profiles

Portal profiles allow you to customize the look and feel of FintechOS Portal instances by changing the: company logo, backgrounds, favicon, system parameters, menus, dashboards, etc. This helps you personalize the user experience you provide to different types of users. For instance, an internal portal that your employees use at work will look and behave differently from a customer facing portal that is used to expose your products and services.

To customize your portals, you define portal profiles and assign them to the desired FintechOS Portal instances. You can reuse your portal profiles by assigning the same portal profile to multiple portals. In addition, you can import your portal profile settings, including background image and company logo, to another environment, by creating your portal profile in the context of a [digital solution](#), exporting and importing the package into the destination environment. These resources are visible in the [FintechOS Editor](#).

### Create a Portal Profile

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Portal Profile**.
2. Click **Insert** to display the Add Portal Profile page.

FintechOS Studio

Feedback

Save and close Save and reload Save and new

Add Portal Profile

Name TestDocs

Description Creating a new portal profile

Company Logo picture.png Add file or Drop file here

Background Image picture.png Add file or Drop file here

Use Full Width Forms

Use Dashboard as Sticky Footer

Favorite Icons

3. Enter a **Name** and **Description** for your portal profile.
4. Upload the **Company Logo** and **Background Image**.
5. Check the **Use Full Width Forms** checkbox if you wish the forms displayed in the portal to cover the full width of the screen.
6. If you want one of your "Dashboards" on page 362 to always be visible at the bottom of the screen, select it in the **Use Dashboard as Sticky Footer** dropdown.
7. To customize the favicon, use the **Favorite Icons** control to insert a new one or select it from the list. When creating a new favicon, you have the option to provide links to

different favicon variations depending on the resolution and type of device.

Add Favicon

Name
Demo

Apple Touch Icon 57×57		Apple Touch Icon 60×60	
Apple Touch Icon 72×72	<a href="https://external-content.duckduckgo.com/iu/?u=http">https://external-content.duckduckgo.com/iu/?u=http</a>	Apple Touch Icon 76×76	
Apple Touch Icon 114×114		Apple Touch Icon 120×120	
Apple Touch Icon 144×144		Apple Touch Icon 152×152	
Apple Touch Icon 180×180			
Fav Icon 16×16	<a href="https://external-content.duckduckgo.com/iu/?u=http">https://external-content.duckduckgo.com/iu/?u=http</a>	Fav Icon 32×32	
Fav Icon 96×96			
Android Icon 192×192		MS Icon 144×144	

8. Click **Save and Reload**.

9. If you want to override specific "[System Parameters](#)" on [page 1285](#), use the **System Parameter Values on Portal Profile** grid to insert your changes. You can also disable such a system parameter if you want to temporarily revert it to the global system parameter value.

## HINT

Some of the system parameters you may use frequently to customize your portal instances are described below:

- sys-hide-dashboard-tabs - Shows/hides the dashboard tabs.
- sys-hide-shortcuts-navbar-left - Shows/hides the navigation bar.
- sys-hide-footer - Shows/hides the page footer.
- sys-hide-menu - Shows/hides the menu.

- sys-hide-company-logo - Shows/hides the company logo.
- sys-minimal-css - Features a clean, minimalistic styling and design for all basic page elements.

10. If you want to filter or disable specific ["Menu Items" on page 371](#), use the **Show only selected menu items** grid to specify the available menu items and if they are temporarily disabled or not. If the grid is empty, all menu items will be displayed.
11. If you want to filter or disable specific ["Dashboards" on page 362](#), use the **Show only selected dashboards** grid to specify the available dashboards and if they are temporarily disabled or not. If the grid is empty, all dashboards will be displayed.
12. Use the **Restrict access to selected Security Roles** grid to specify the ["Security Roles" on page 1245](#) that allow access to the portal. If the grid is empty, all security roles will be able to access the portal instance.
13. Click **Save and Close**.

## Assign a Portal Profile to a FintechOS Portal Instance

To customize a FintechOS Portal to use a specific portal profile, go to the [Configuration Manager](#) and add the following secret:

Key Path	Key Name	Key Value
kv/<environmentName>/<portalName>/app-configurations	portal-profile	Portal profile name

### IMPORTANT!

The value for the `portal-profile` secret must match a previously configured portal profile name.

You can reuse your portal profiles by assigning the same portal profile to multiple portals.

## B2C Portals

B2C Portals, formerly named Anonymous Frontends, expose specific "Digital Journeys" on page 189 to unauthenticated users. This allows you to provide access to specific user journeys (for instance getting a quote for a car insurance) without having to create a FintechOS Platform user account in advance.

### Expose a Digital Journey to Anonymous Users

1. To expose a digital journey to anonymous users, in the [Configuration Manager](#), at the `kv/<environment name>/b2cportal/b2c` path, edit the following key:

Key	Value
Journeys	<pre>[   {     "Name": "JourneyTutorial",     "UseHttpRedirect": true,     "AllowedHeaders": [ "header1", "header2" ],     "RedirectUriInitiate":       "https://myServer.net/b2cportal/#/userjourney/myEntity/insert/form/myForm",     "RedirectUriError":       "https://myServer.net/platform/Content/expired.html",     "RedirectUriResume":       "https://myServer.net/b2cportal/#/entity/myEntity/edit/00000000-0000-0000-0000-0000-000000000000/form/myForm/pageno/1",     "AnonymousUserLifespanDays": 30,     "TemporaryUser": {       "BusinessUnit": "root",       "SecurityRoles": [ "SecRoleTutorial" ]     }   } ]</pre>

Parameter	Description
Name	Name of the digital journey to be exposed anonymously.
TemporaryUser	<p>Configures the temporary user accounts created for each anonymous session.</p> <ul style="list-style-type: none"> <li>SecurityRoles - Security roles assigned to the temporary anonymous users.</li> </ul> <div style="background-color: #f9e79f; padding: 10px; margin: 10px 0;"> <p><b>IMPORTANT!</b></p> <p>These security roles must be also set up on all the flows of the anonymous journey (see <a href="#">"Define access to the flow" on page 229</a>). If regular authenticated users also need access to the journey, they too need to have these security roles assigned.</p> </div> <ul style="list-style-type: none"> <li>BusinessUnit - Business unit associated with the temporary anonymous users.</li> </ul>
AllowedHeaders	HTTP headers to be forwarded in responses to the <a href="#">"Initiate an Anonymous User Journey" on page 384</a> and <a href="#">"Resume an Anonymous User Journey" on page 385</a> requests.

Parameter	Description
AnonymousUserLifespanDays	Lifespan in days of the anonymous user accounts. During this time, the anonymous user can resume an interrupted journey. Default value: 30.
RedirectUriInitiate	FintechOS Portal URI returned when requesting to <a href="#">"Initiate an Anonymous User Journey" on the next page</a> . E.g.: An insert form.
RedirectUriResume	<p>FintechOS Portal URI returned when requesting to <a href="#">"Resume an Anonymous User Journey" on page 385</a>. E.g.: An edit form.</p> <div> <p><b>NOTE</b></p> <p>If the path includes a record ID, you can use any valid guid, e.g.: 00000000-0000-0000-0000-000000000000.</p> </div>
RedirectUriError	URI returned if a request to <a href="#">"Resume an Anonymous User Journey" on page 385</a> cannot be fulfilled (the session expired).

Parameter	Description
UseHttpRedirect	<p>Determines if the user agent receives a redirect link or the actual page content when requesting to initiate/resume an anonymous user journey.</p> <ul style="list-style-type: none"> <li>• <b>true</b> - Returns an HTTP status code 302 response with the corresponding RedirectUriInitiate, RedirectUriResume, or RedirectUriResume address in the Location header field.</li> <li>• <b>false</b> - Returns an HTTP status code 200 response with the content of the corresponding RedirectUriInitiate, RedirectUriResume, or RedirectUriResume page in the response body.</li> </ul>

2. Add the same exact key from the previous step to the portal or open api path, depending on your scenario. The path to the portal section might look like this: `kv/<environment name>/portal/b2c`. If not added, the error message "Journey not found" will be thrown at user creation with the function `ftos.identity.users.createTemporary`.

## Activate B2C Portals

To activate B2C portals globally, in the [Configuration Manager](#), at the `kv/<environment name>/app-features/B2C` path, set the **feature-b2c-userjourneys** key to **1** (setting the parameter to 0 deactivates the anonymous B2C portals).

## Initiate an Anonymous User Journey

### Request

To initiate an anonymous journey, call the **b2cportal/platform/B2CAuth/InitiateJourney** endpoint using the GET method. Use

the *journeyName* parameter in the call's URL to specify the name of the anonymous digital journey. To localize the journey in a specific language use the *culture* parameter with the value at your choice. The default value is English. E.g.:

```
https://platform-  
url/b2cportal/platform/B2CAuth/InitiateJourney?journeyName=JourneyTutorial&culture=RO-ro
```

### Response

Depending on the ["UseHttpRedirect" on the previous page](#) parameter setting, the response will include either a redirect link or the actual content of the ["RedirectUriInitiate" on page 383](#) page.

The response will also contain a cookie with the temporary user account information. The user will have the business unit and the security roles specified in the journey configuration.

## Resume an Anonymous User Journey

### Request

To resume an anonymous journey, call the **b2cportal/platform/B2CAuth/ResumeJourney** endpoint using the GET method. Use the same *journeyName* parameter in the URL to specify the name of the journey, and the *sessionId* parameter to identify the session belonging to the temporary user:

```
https://platform-  
url/b2cportal/platform/B2CAuth/ResumeJourney?journeyName=JourneyTutorial&sessionId=C175BFE5-EAFD-422F-A57F-7D85E806C38B
```

### HINT

Use the [ftos.context.user.b2CSessionId](#) Server SDK function to get the session ID of the anonymous journey.

### Response

In case the ["UseHttpRedirect" on the previous page](#) parameter value is set to false , and if the resume happens within the of days set in the *AnonymousUserLifespanDays* parameter, the user is redirected to the ["RedirectUriResume" on page 383](#) page.

If the anonymous journey activation timeframe has passed, and the ["UseHttpRedirect" on page 384](#) parameter value is set to true, the endpoint returns the user to the ["RedirectUriError" on page 383](#) page.

## Initiate/Resume Anonymous User Journeys with Digital Journey APIs

If you wish to access anonymous user journeys via custom applications, web applications, front-ends, etc., use the dedicated [initiate](#) and [resume](#) digital journey APIs.

## Resume External Journeys in an Anonymous B2C Environment

To allow the user to resume a pre-populated digital journey in an anonymous B2C environment:

1. Create a new temporary identity and session on the same journey by using the [ftos.identity.users.createTemporary](#) server SDK method in an event triggered automation script.
2. Transfer data ownership from the existing user to the temporary one by adding the following to the script:

```
update(entityName,recordId, {UserId:tempIdentity.temporaryUserID});
```

3. Provide updated ["Resume an Anonymous User Journey" on the previous page](#) with the new session ID to the user.

### HINT

Use the [ftos.messaging.send](#) method to communicate the journey link to your customer.

## Use Custom Parameters in Anonymous User Journeys

You can provide custom parameters in an initiate or resume journey request by appending their names and values at the end of the request (preceded by the **&** symbol). E.g.:

```
https://platform-  
url/b2cportal/platform/B2CAuth/InitiateJourney?journeyName=JourneyTutorial&culture=RO-ro&parameter1=value1&parameter2=value2
```

or

```
https://platform-  
url/b2cportal/platform/B2CAuth/ResumeJourney?journeyName=JourneyTutorial&sessionId=C175BFE5-EAFD-422F-A57F-7D85E806C38B&parameter1=value1&parameter2=value2
```

### Retrieve Custom Parameters in Anonymous User Journeys

Use the `searchParams` property to retrieve custom parameters values in the journey's client side code.

If your portal is running in [Single Page Application mode \(EBS router\)](#) use the following model:

```
EbsRouter.getState().searchParams.parameterName
```

If your portal is using legacy URLs use the following model:

```
var urlobj = new window.URL(window.location.href)  
urlobj.searchParams.get("parameterName")
```

## Manage Style Sheets for Anonymous User Journeys

FintechOS Portal instances that run anonymous user journeys (B2C portals) have fewer default style sheets and don't use themes. Therefore, you may have to apply some of the portal specific styling at the digital journey level.

To manage the style sheets for digital journeys which are exposed to unauthenticated users, follow the basic procedure for managing your ["Style Sheets" on page 1209](#).

### NOTE

There are specific style particularities for anonymous user journeys, described in the table below.

The following CSS elements are available for B2C user journeys:

Parameter	Default Value	Observations
--defaultFontSize	14px	
--defaultTextColor	#333	
--linkColor	#337ab7	
--linkHoverColor	#23527c	
--controlTextColor	#333	The color of the text displayed in controls. E.g.: OptionSet control.
--controlIconColor	#333	
--controlBorderColor	#ddd	
--controlActiveBorderColor	#337ab7	
--errorColor	#d64031	The background color of error toast messages.
--infoColor	#2980b9	
--warningColor	#feb332	
--successColor	#049F0C	

## Login Screen Themes

The FintechOS platform offers a variety of tools to personalize the appearance and functionality of your digital solutions. You can design and customize B2C Portals to present your solutions to customers, or refer to this guide for detailed customization options. Additionally, you can modify the login screen themes for both your Studio and Portal.

### Edit the Theme

1. Go to an environment with v24.4 and create a digital solution package, as explained on the [Digital Solutions Packages](#) page.
2. Include the UI theme associated with the Portal Profile in the package. Export the package.
3. Open the package locally and edit the themes, as explained on the [themes documentation](#) page.
4. Import the edited package in Studio, as explained on the [Digital Solutions Packages](#) page.

#### NOTE

If you're using the SysPackageDeployer to upload the package, make sure to

configure the settings added for IDP themes in the .config file as explained on the [Importing and Exporting Packages](#) page.

Showing all 10 items

<input type="checkbox"/>	Name	Last modified	Access tier	Blob type	Size	Lease state
<input type="checkbox"/>	📁 Evolv					...
<input type="checkbox"/>	📁 EvolvDark					...
<input type="checkbox"/>	📁 Portal					...
<input type="checkbox"/>	📁 Studio					...
<input type="checkbox"/>	📁 baseAdmin					...
<input type="checkbox"/>	📁 baseEvolv					...
<input type="checkbox"/>	📁 basePortal					...
<input type="checkbox"/>	📁 baseStudio					...
<input type="checkbox"/>	📄 README.md	7/8/2025, 9:01:00 AM	Hot (Inferred)	Block blob	291 B	Available
<input type="checkbox"/>	📄 README.txt	7/8/2025, 9:01:00 AM	Hot (Inferred)	Block blob	291 B	Available

## Update and Upgrade

At clean install, the themes are automatically applied to the environment. If you've customized the Studio and Portal themes in versions prior to v24.3, you should considering several changes when updating or upgrading. One of the change is that the themes are now hosted in the container named **container-keycloak-themes** in Azure, and not the **idpthemes** container as in previous versions.

- When you're updating from a minor version to another, for example from v24.3 to v24.4, the IDP themes should automatically be updated with the latest changes.
- When upgrading from one major version to another, for example from v22.x to v24.4, the theme files are overwritten with the v24.x ones. If you have custom themes on v22.x, it is mandatory to manually adjust them for the upgrade.

Be aware that a number of files have changes in the 24.3 version, for example, in the baseStudio/login/login.ftl file, the "loginAccountTitle" parameter was removed. So you should be aware of this at upgrade. Another changed file is index.ftl, the productNameFull was removed.

Starting with 24.4, the Evolv and EvolvDark themes are available. Before update or upgrade, take into consideration the following:

- At upgrade or update to 24.4, the Evolv, EvolvDark and baseEvolv folders are automatically deployed
- You can use one theme, Evolv or EvolvDark, for both Studio and Portal

- The Studio and Portal folders, which contain the themes for previous Platform versions, are still there, in the container, but they get overwritten at upgrade or update

### IMPORTANT!

If you've made changes to the custom theme, i.e. to folders Studio and/or Portal, before updating or upgrading to 24.4, make sure to **backup** your files, as the two folders, Studio and Portal will be overwritten.

### NOTE

In the likely scenario that you need to make custom changes to login themes, as a best practice, make sure to create a duplicate of default themes and work on those, so that your custom changes don't get overwritten at environment upgrade or update.

Environment variables have changed, you must check that you have the correct ones put in place:

- the KC\_PROXY=edge environment variable from Keycloak 22 was switched to KC\_PROXY\_HEADERS=xforwarded
- the KEYCLOAK\_ADMIN and KEYCLOAK\_ADMIN\_PASSWORD environment variables were switched to KC\_BOOTSTRAP\_ADMIN\_USERNAME and KC\_BOOTSTRAP\_ADMIN\_PASSWORD, respectively

If you want to start your new Keycloak in production mode and encounter the hostname strict error, please update the configuration start command of your Azure App Service the like the one below:

```
start --hostname-strict false
```

General settings Path mappings Error pages (preview)

#### Stack settings

Startup Command

```
start --hostname-strict false
```

 Provide an optional startup command that will be run as part of container startup. [Learn more](#)

## Anonymous Frontends (deprecated)

### IMPORTANT!

This implementation of anonymous frontends is no longer supported. For more information, see ["B2C Portals" on page 381](#). For information on how to migrate the legacy anonymous frontends to the current implementation see the following [migration guide](#).

This information is provided so that you can still reference your legacy anonymous frontends for migration.

Anonymous Frontends expose data from select ["Form Driven Flows" on page 221](#) and ["Custom Flows" on page 283](#) to unauthenticated users. This allows you to provide access to specific user journeys (for instance with a button or widget on your public website) to users who don't have a FintechOS Platform user account.

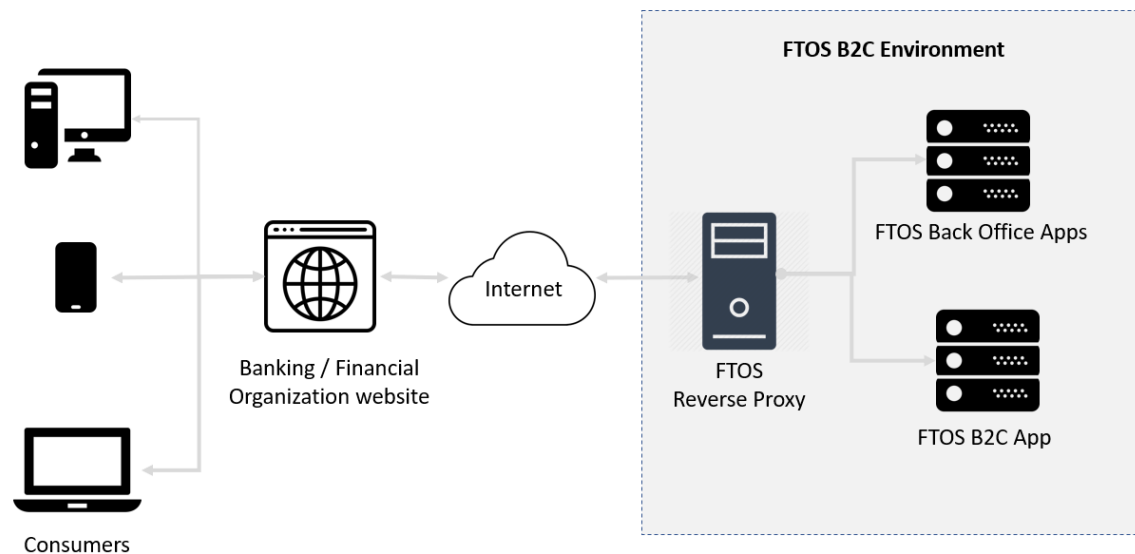
### Anonymous Frontends Security

An anonymous frontend environment with a secure architecture has been designed to allow exposing journeys to unauthenticated users (consumers).

The form driven flow is exposed via an iFrame. As iFrames are vulnerable to hacker attacks, a reverse proxy sits between the internet and FintechOS Platform apps that are placed in a non-public subnet.

The reverse proxy ensures a single point of authentication for all HTTP requests, forwarding the requests to the FintechOS Platform B2C App (that contains the form driven flow to be exposed). It also handles requests to the FintechOS Platform Back Office apps (FintechOS Studio and portals).

Here's a simplified diagram of the traffic flow.



In order to expose digital journeys, you need to set up a B2C environment.

## Set Up a B2C Environment

The B2C environment offers a secure architecture for exposing digital journeys to unauthenticated users. The B2C environment is comprised of the following components:

- FintechOS Platform Back Office applications (Portal and FintechOS Studio)
- FintechOS Platform B2C Application, which is a copy of the back-office Portal with the following keys set in the [Configuration Manager](#):

Key Path	Key Name	Key Value
kv/<environment>/<FintechOS Portal instance>/app-settings	feature-b2c-userjourneys	1
kv/<environment>/<FintechOS Portal instance>/app-settings	feature-b2c-userjourneys-impersonated-user	username

Key Path	Key Name	Key Value
kv/<environment>/<FintechOS Portal instance>/app-settings	feature-b2c-userjourneys-api-key	feature-b2c-userjourneys-api-key

**IMPORTANT!**

- The impersonated user (feature-b2c-userjourneys-impersonated-user) cannot be **host** or **admin**.
- The "Security Roles" on [page 1245](#) for the impersonated user must include only read access to the related dictionary entities (entities referenced by lookup attributes used in the journey).
- Since the impersonated user has limited access to the journey's data model, make sure that you do not use data manipulation functions (such as [ebs.getByID](#), [ebs.getByIdAsync.htm](#), [ebs.getByQuery.htm](#), etc.) in the journey's client-side code. Instead, use server-side scripts which you can access through functions such as [ebs.callActionByNameAsync.htm](#) or [ebs.callActionByName.htm](#).

**NOTE**

To expose multiple user journeys to a B2C portal, add the user journeys API keys in the Configuration Manager, with the values separated by comma:

Key Path	Key Name	Key Value
kv/<environment>/<FintechOS Portal instance>/app-settings	feature-b2c-userjournes-api-key	feature-b2c-userjourneys-api-key, feature-b2c-userjourneys-api-key

- FTOS Reverse Proxy application which ensure secure routing of HTTP requests from Internet to FintechOS Platform apps.

## 1 Install the FTOS Reverse Proxy

1. Install [.NET Core Sdk](#).
2. Install [.NET Core 2.2 Runtime and Hosting Bundle for Windows](#).
3. Copy the FTOS Reverse Proxy files in a folder of your choice.
4. Create an application in Microsoft IIS:
  - a. Create the application pool (.NET CLR=No Managed Code | Managed pipeline mode=Intergrated | Identity=NetworkService)
  - b. Create a web application in IIS using the Application Pool you have just created and having the Physical path as the folder where you copied the FTOS Reverse Proxy files.

## 2 Configure the FTOS Reverse Proxy

In the FTOS Reverse Proxy installation folder, find the **proxy/proxy.config.js** file and make the necessary changes:

```
//the output refers to the __FintechOS B2C Application__
output.scheme = "http";
output.port = 80;
output.host = "192.168.15.15";
```

```
output.application = "FintechOS_B2C";// if there is no application
(FintechOS B2C runs as a root site in IIS) use output.application =
null
```

Configure your routes following the example from the file.

Make sure that you set the B2C header with the value you have set up in the Configuration Manager for the FintechOS B2C application (**feature-b2c-userjourneys-api-key**):

```
output.requestHeaders.set("B2C", "feature-b2c-userjourneys-api-
key");
```

FTOS Reverse Proxy Configuration

```
filter.setup(function (input, output)
{
    /* LOCAL TEST */
    output.scheme = "http";
    output.port = 60130;
    output.host = "localhost";
    output.application = null;
    /* -----
    // mandatory for Fintech, mapping # to Main#
    // -----
    */
    if (output.path.match(/^#/)) {
        output.path = "Main" + output.path;
        return output.redirect();
    }
    if (output.path === "/rca")
    {
        if (output.query.match(/\bsessionId=i))
        {
            output.path = "/Main";
            output.query +=
"#/entity/claimNotification/edit/newEntry/data
form/b2cUJ/pageno/1";
        }
        else
        {
            output.path = "/Main";
            output.query +=
"#/userjourney/claimNotification/insert/data form/b2cUJ";
        }
    }
}
```

```

        return output.redirect();
    }

    output.requestHeaders.set("B2C", "feature-b2c-userjourneys-
api-key");

    return output.go();
});

```

When configuring routes use the following templates:

- Insert Link: `#/userjourney/{entityName}/insert/data form/{formName}_`
- Edit Link: `#/entity/{entityName}/edit/old/data form/{formName}/pageno/{pageNo}`

### 3 Allow Access to the Journey Through the Reverse Proxy

To enable a specific digital journey to be accessible through the reverse proxy, In FintechOS Studio follow these steps:

1. In FintechOS Studio, go to **Main Menu > Digital Experience > Digital Frontends > Anonymous Frontends**.
2. Click **Insert**.
3. In the **Name** field, fill-in the name of the B2C domain and in the API Key field (provide the `feature-b2c-userjourneys-api-key` value you have set up in the Configuration Manger).
4. Click **Save and reload**.
5. Populate the Published Form Driven Flows grid with the "Form Driven Flows" on [page 221](#) you wish to expose.
6. Populate the Published Custom Flows grid with the "Custom Flows" on [page 283](#) you wish to expose.
7. Click **Save and close**.

Once unauthenticated users complete a digital journey, their records are logged in. To review the logs, go to **Main Menu > Security > Anonymous Journey Access Logs**.

#### 4 Override the Default Save Mechanism

Now that you've set up the B2C environment and you exposed journeys to unauthenticated users, you need to override the default save on the digital journey. For information on how to do that, see ["Override the Default Save Mechanism \(deprecated\)" on page 399](#).

### Enable Caching on the FTOS Reverse Proxy (optional)

You can improve the performance of your anonymous frontends by caching resources for your unauthenticated form driven flows or custom flows at the reverse proxy level. To do so, add an entry based on the model below in the reverse proxy's `appsettings.json` configuration file:

```
"Caching": {
  "CachedPaths": [ "/Scripts/**", "/custom/*.css", "/custom/*.js"
],
  "SlidingExpirationTime": "1m 20s"
}
```

Setting	Description
CachedPaths	Contains the paths of the resources you wish to cache relative to the input path of the reverse proxy server. For instance, if you wish to cache the following JavaScript file <code>https://www.myCompany.com/Scripts/bundle-static.js</code> , you must add the <code>/Scripts/bundle-static.js</code> path to the cached resources. You can use the <code>*</code> wildcard to replace strings of text or <code>**</code> to recursively select everything in a folder and its subfolders. If you don't set any value, caching is disabled.
SlidingExpirationTime	Sets the cached resources' expiration time using the following format: <code>1w 1d 1h 1m 1s</code> . Whenever a resource is accessed, its cache expiration time is reset. Default value: 24h.

## Customize the Styles Sheets (optional)

The B2C journeys use the FintechOS Platform default styles by default, so you may want to create and use your own styles sheets. For more information, see ["Manage Style Sheets for Anonymous User Journeys" on page 387](#).

## Customize the Language (optional)

An anonymous frontend can serve in a specific language. For information on how to set anonymous frontends to serve in a specific language, see ["Serving User Journeys in a Specific Language \(deprecated\)" on page 401](#).

## Reset an Anonymous Frontend Session

Resetting an anonymous frontend session is useful when running multiple instances of an anonymous frontend, to ensure that, if users go back to a previous step, they don't lose the data that has already been saved in previous steps.

## Example

An anonymous frontend for a loan application has multiple steps. In one of the steps, the customer is asked to provide the monthly income. Based on this value, the maximum loan is calculated in a different step.

To register the customer loan application without altering the data based on which the loan amount was calculated, reset the anonymous frontend session by using the `ftos.core.resetB2CSession()` function in the last step (**Advanced** tab > **After Section Save** tab) of the loan application digital journey.

To reset an anonymous frontend session:

1. In FintechOS Studio, go to the configuration page of the digital journey exposed to anonymous users.
2. Open the **Steps** tab and in the **Entity Form Steps** section, double-click the last step of the digital journey (the step with the highest order index).
3. Open the **Advanced** tab.

4. Open the **After Section Save** tab and in the JavaScript field, type the function

```
ftos.core.resetB2CSession();
```

5. Click **Save and close**.

## Override the Default Save Mechanism (deprecated)

### IMPORTANT!

This procedure is no longer required in the current anonymous frontends implementation. For more information, see ["B2C Portals" on page 381](#). For information on how to migrate the legacy anonymous frontends to the current implementation see the following [migration guide](#).

The default save method on forms saves into the database the data inserted by users when filling in the forms. If you want to manipulate the data before saving it into the database, you can do so by overriding the default save method.

Overriding the default save method on data form is also useful if the current user does not have privileges on that entity. You can achieve this on the server side by using an endpoint (action) in three steps:

### 1 Create an On-Demand Automation Script

This is the script that will be executed instead of the default save function. For information on how to create on-demand automation scripts, see ["Create On-demand Server Automation Scripts" on page 1168](#).

If you want to override the save on forms exposed to unauthenticated users, use this minimal working code in the on-demand script.

```
setAdminMode(true);
//Log(context);
var activeStatus = getOptionSetItemId("B2CProcessStatus",
"Active");
var entityIdByName = getEntityIdByName(context.EntityName);
var dateName = new Date().toString();
var sessId = server.B2C.SessionId;
var saveData = getEndpointSaveData();
saveData.EntityValues["formName"] = saveData.FormName;
saveData.EntityValues["sectionIndex"] = saveData.SectionIndex;
saveData.EntityValues["sectionName"] = saveData.SectionName;
if(saveData.OperationType == "edit"){
```

```

        update(saveData.EntityName, saveData.Id,
saveData.EntityValues);
    }
    else if(saveData.OperationType == "insert"){
        var generatedId = insert(saveData.EntityName,
saveData.EntityValues);
        setData({"Id": generatedId});

        var epVals = {
            "sessionId": sessId,
            "name": dateName,
            "entityId": entityIdByName,
            "recordId": generatedId,
            "status": activeStatus
        }
        var B2CExternalProcessId = insert("B2CExternalProcess",
epVals);
    }
    function getEntityIdByName(name){
        var a = getByQuery({
            entity: {name: "entity", alias: "a"},
            where: {
                type: "and",
                conditionlist: [{
                    type: "equals",
                    first: "a.name",
                    second: "val(" + context.EntityName + ")"
                }]
            }
        });
        return a[0]["a_entityid"];
    }
}

```

Where:

- `saveData.EntityValues` contains the values that will normally be passed to the default save function.
- FintechOS Platform uses the `sessionId` parameter to retrieve the desired record through the B2C External Process entity.

## 2 Create an Endpoint and Attach the Script to it

Create an endpoint (action) and add the automation script created at step 1 to it.

For information on how to create an endpoint and attach an automation script to it, see ["Endpoints" on page 1213](#).

### 3 Call the Endpoint on the Form Driven Flow

In the Before Events of the form driven flow, call the `setSaveEndpoint` method of the `formData` object, with the name of the endpoint created at step 1:

```
formData.setSaveEndpoint("endpointName");
```

When overriding the default save functionality, you can use the `formData.getEndpointSaveData()` method to retrieve the entity save data and other save context information. This function returns an `IEndpointSaveData` object:

```
interface IEndpointSaveData
{
    Id: string;
    EntityName: string;
    OperationType: "edit" | "insert";
    FormName: string;
    SectionIndex: number;
    SectionName: string;
    EntityValues: any;
}
```

## Serving User Journeys in a Specific Language (deprecated)

### IMPORTANT!

This is no longer supported in the current anonymous frontends implementation. To localize an anonymous journey in a specific language, use the *culture* parameter when calling the **B2CAuth/InitiateJourney** endpoint (see ["Initiate an Anonymous User Journey" on page 384](#) for details).

For more information about the current anonymous frontends implementation, see ["B2C Portals" on page 381](#). For information on how to migrate the legacy anonymous frontends to the current implementation see the following [migration guide](#).

The platform enables you to serve a B2C digital journey in a specific language, that is, the digital journey can be started in a specific language.

For a B2C journey to be started on a specific language, the language should exist in the system and the resources should be localized in that language. For more information on how to add a language and details on localization resources, see ["Localization" on page 1269](#).

You have two options to set up a digital journey to start in a specific language:

- Add the culture query parameter to the proxy URL. For instance, to launch a digital journey in Romanian: *https://proxyurl?culture=ro-RO*
- Configure the FTOS Reverse Proxy by setting the culture. To do so, In the FTOS Reverse Proxy installation folder, find the **proxy/proxy.config.js** file and add the **output.query** property.

```
//the output refers to the __FintechOS B2C Application__  
...  
output.query += "?culture=ro-  
RO#/userjourney/claimNotification/insert/data form/b2cUJ";
```

Once users launch the digital journey, records are automatically added in FintechOS Studio to the B2C External Process entity and FintechOS Platform engineers will be able to see the culture in which the digital journey has been started.

## Digital Experience Portals

FintechOS Studio provides various ways for streamlining the experience of your business users by customizing the **Digital Experience Portals** in accordance to their needs.

The following customization features are available:

- Customize the login and home page
- Use a custom UI theme
- Use custom icons
- Visual branding
- Add digital journey sticky header

- Customize dashboards using widgets
- Show tooltips (if allowed by the Portal customization).

With the use of portal profiles, you can also customize **Digital Experience Portals** with specific elements like background image, menu items, dashboards, or specific values for system parameters. For more information on portal profiles, see ["Portal Profiles" on page 377](#).

### Theme Support

The portals come with multiple options for customizing the layout by uploading a background image, generating a color palette, or using a floating style, global dashboard, or shortcuts on the homepage.

### Custom Theme Support

Custom theme support is useful for streamlining, automating, and merging deployments with FintechOS Platform upgrades.

### Enhanced Security

The portals facilitate adaptive user interface based on role (designer, portal), with available apps displayed accordingly, as well as dedicated data form for self-service user profile management.

### Visual Branding Support

The portals come with the default color palette inherited from FintechOS Platform brand. However, you can switch it over to custom colors and logo or pick one of the available color palettes.

### Enriched Dashboards

There are extensive types of elements supported within user dashboards, such as KPIs, HTML widgets, charts and Fincharts, views, and Power Bi reports. Security wise, all such elements may be restricted to certain user roles.

### Native Analytics

The portals feature a powerful library of charts for displaying business information in a compelling and visual way, serving diverse and complex analytics needs.

The figure below shows an example of how a **Digital Experience Portal** may be customized:



## Using Custom Icons

Personalizing your **Digital Experience Portal** icons is a great way to identify the Portal with your company's brand, making it uniquely yours.

This topic covers everything you need to know in order to use custom icons.

### What files do you need?

The following is a recommended best practice for structuring a file:

```
+-- customIcon
|   +-- css
|   |   +-- customIcon.css
|   +-- font
|   |   +-- customIcon.eot/ttf/woff/woff2/svg
|   +-- customIcon.js
```

You will need your font files (e.g., .eot, .woff, .woff2, etc.). Once you have the font files, import them in your css file and make sure to add the icon classes.

**IMPORTANT!** FintechOS Platform does not support spinning icons or any sort of animated icons if the animation is done with a separate class.

.css file

```
@font-face {
    font-family: 'customIcon';
    src: url('../font/customIcon.eot?48188603');
    src: url('../font/customIcon.eot?48188603#iefix') format
('embedded-opentype'),
    url('../font/customIcon.svg?48188603#fontello') format
('svg');
    font-weight: normal;
    font-style: normal;
}
[class^="customIcon-"]:before, [class*=" customIcon-"]:before {
    font-family: "customIcon";
    font-style: normal;
    font-weight: normal;
    speak: none;
    display: inline-block;
    text-decoration: inherit;
    width: 1em;
    margin-right: .2em;
    text-align: center;

    /* For safety - reset parent styles, that can break glyph
codes*/
    font-variant: normal;
    text-transform: none;

    /* fix buttons height, for twitter bootstrap */
    line-height: 1em;

    /* Animation center compensation - margins should be
symmetric */
    /* remove if not needed */
    margin-left: .2em;

    /* you can be more comfortable with increased icons size */
    /* font-size: 120%; */
}
```

```

/* Uncomment for 3D effect */
/* text-shadow: 1px 1px 1px rgba(127, 127, 127, 0.3); */
}

.customIcon-happy:before { content: '\e800'; } /* ' ' */
.customIcon-wink:before { content: '\e801'; } /* ' ' */
.customIcon-unhappy:before { content: '\e802'; } /* ' ' */
.customIcon-sleep:before { content: '\e803'; } /* ' ' */
.customIcon-thumbsup:before { content: '\e804'; } /* ' ' */
.customIcon-devil:before { content: '\e805'; } /* ' ' */
.customIcon-surprised:before { content: '\e806'; } /* ' ' */
.customIcon-tongue:before { content: '\e807'; } /* ' ' */
.customIcon-coffee:before { content: '\e808'; } /* ' ' */
.customIcon-sunglasses:before { content: '\e809'; } /* ' ' */
.customIcon-displeased:before { content: '\e80a'; } /* ' ' */
.customIcon-beer:before { content: '\e80b'; } /* ' ' */
.customIcon-grin:before { content: '\e80c'; } /* ' ' */
.customIcon-angry:before { content: '\e80d'; } /* ' ' */
.customIcon-saint:before { content: '\e80e'; } /* ' ' */
.customIcon-cry:before { content: '\e80f'; } /* ' ' */
.customIcon-shoot:before { content: '\e810'; } /* ' ' */
.customIcon-squint:before { content: '\e811'; } /* ' ' */
.customIcon-laugh:before { content: '\e812'; } /* ' ' */
.customIcon-wink2:before { content: '\e813'; } /* ' ' */
.customIcon-spin1:before { content: '\e830'; } /* '𐀀' */
.customIcon-spin2:before { content: '\e831'; } /* '𐀁' */
.customIcon-spin3:before { content: '\e832'; } /* '𐀂' */
.customIcon-spin4:before { content: '\e834'; } /* '𐀄' */
.customIcon-spin5:before { content: '\e838'; } /* '𐀈' */
.customIcon-spin6:before { content: '\e839'; } /* '𐀉' */
.customIcon-firefox:before { content: '\e840'; } /* '𐀐' */
.customIcon-chrome:before { content: '\e841'; } /* '𐀑' */
.customIcon-opera:before { content: '\e842'; } /* '𐀒' */
.customIcon-ie:before { content: '\e843'; } /* '𐀓' */
.customIcon-crown:before { content: '\e844'; } /* '𐀔' */
.customIcon-crown-plus:before { content: '\e845'; } /* '𐀕' */
.customIcon-crown-minus:before { content: '\e846'; } /* '𐀖' */

```

There are tools you can use to generate custom icon fonts. We recommend you to use [Fontello](#) as it quickly builds everything you need to include vector images into your web pages and it also provides open-source artworks.

If you opt for using Fontello, after you select/upload the icons that you want to use, make sure that all the icons have the same prefix. You can accomplish this from the **Customize Names** tab.

Icons must the same "customIcon" prefix:

```
customIcon-happy
customIcon-wink
customIcon-unhappy
customIcon-sleep
```

Create the JavaScript file that will do the heavy lifting of delivering your icons into the platform:

```
! function ($) {
  if (!$.iconset_list || !$.iconset_list.length) {
    $.iconset_list = [];
  }
  /* Replace customIcon with your own icons family name */
  /* The way that the app adds your icon class to the icon container is this: class="iconClass iconClassFix-icon" */
  /* The iconClass is not mandatory but we have it because some font families use it (ex. fontawesom). */
  $.iconset_customIcon = {
    iconFamily: "customIcon",
    iconClass: "",
    iconClassFix: "customIcon-",
    icons: ["happy", "wink", "unhappy", "sleep", "devil", "surprised", "sunglasses", "chrome", "opera", "ie"]
  }
  $.iconset_list.push('customIcon');
}(jQuery);
```

### Files Location

Add the files to the following location: *custom/customAssets/icons/customIconSet*. If you have another custom folder path set up, add the files to the following location *yourCustomFolder/customAssets/icons/customIconSet*, where customIconSet is the folder containing your files.

**NOTE** If you add your files to a location other than the ones mentioned above, the icons will not be loaded.

## Use custom icons

If you got all the needed files and you put them into the custom folder previously mentioned, in FintechOS Studio, from the menu, click **Admin > Settings**. On the **General** tab, scroll-down, select **Use Custom Styles** and in the **Custom Styles Folder** field, enter the name of the folder where you have the custom files located, if it is other than "custom".

## Setting Sticky Header

On data form driven flows, in the **Journey Configuration** page, **Header Items** tab, if you defined several header items, select the **Sticky Header Items** checkbox (it is selected by default) and on save, the journey header items will be displayed in the **Digital Experience Portals** on the menu bar.

### NOTE

The Portal users might need to refresh the Portal in order for the header items to appear on the specific digital journey.

The figure below shows how the campaign header items are displayed in the **Digital Experience Portals**:

CAMPAIGN NAME	TYPE	START DATE	END DATE	REMAINING DAYS
ET02 Camp Test 2		10/23/2018	10/30/2018	8

1 Setup

2 Content

3 Audience

4 Schedule

EDIT CAMPAIGN

Name

ET02 Camp Test 2

Start Date

23/10/2018

End Date

30/10/2018

Campaign Type

Campaign Subtype

Campaign Priority

High

Campaign Identifier

Total Number of Days

8

No of days since start

0

Remaining Days

0

## Keyboard Shortcuts

FintechOS Platform provides key combinations you can use from the keyboard as an alternative way to do something that you'd typically do with a mouse.

## General keyboard shortcuts

Press this key	To do this
<b>Ctrl + S</b>	Save record changes on forms, sections, wizard mode, when inline editing views (data form, row, batch), when inline editing a data form with children. If the child is on edit mode, when pressing <b>CTRL+S</b> , the system saves both the child entity and the parent entity.
<b>Tab</b>	Move forward through entity records and fields.
<b>Shift + Tab</b>	Move back through entity records and fields.
<b>Alt + Left arrow</b>	<p>Go back.</p> <p>If you want to go back after editing records, either by clicking back or pressing the shortcut, a pop-up will be displayed informing you that changes were made and asking if you want to go back without saving the changes. Clicking <b>Yes</b> will go back without saving the changes . Clicking <b>No</b> will close the pop-up and you'll stay on the page.</p>
<b>Alt + Right arrow</b>	Go forward.

## Shortcuts for date/ date time fields

Press this key	To do this
<b>Alt + Down arrow</b>	Display calendar
<b>Alt + Up arrow</b>	Hide calendar.
<b>Shift + Page Up</b>	Move to the previous month in the calendar.

Press this key	To do this
<b>Shift + Page Down</b>	Move to the next month in the calendar.
<b>Enter</b>	Select the date where the focus is.
<b>Esc (Escape)</b>	Close the date picker without selection.

#### Shortcuts for drop-down fields

Press this key	To do this
<b>Up arrow</b>	Scroll up values in a drop-down without opening the drop-down.
<b>Down arrow</b>	Scroll down values in a drop-down without opening the drop-down.
<b>Alt + Down arrow</b>	Open the drop-down.
<b>Up / Down arrows</b>	Scroll the values of an open drop-down in the direction specified.
<b>Alt + Up arrow</b>	Hide the drop-down.
<b>Enter</b>	Select value.

#### Shortcuts for radio buttons

Press this key	To do this
<b>Left / Right arrow</b>	Deselect or select the radio button if the active option is a radio button.

#### Shortcuts for check boxes

Press this key	To do this
<b>Spacebar</b>	Select or clear the check box if the active option is a check box.

# Automation Blocks

Automation blocks are advanced pre-built features that you can customize and include in your digital journeys. They cover frequent use cases, such as OCR extraction, identity validation, video streaming and collaboration, marketing campaigns management, electronic signatures, digital documents, etc.

Some of the automation blocks, such as [Business Formulas](#) or the [Business Workflows Processor](#) are included in FintechOS Studio.

Others are distributed via SysPacks and must be installed separately. You can use SysPacks to install automation blocks for: OCR extraction, identity validation, Video streaming and collaboration, digital documents management and signing, marketing campaigns management, task management, business decisions, and so on. You can read more about this on the dedicated [Automation Blocks documentation](#).

## Business Workflows

Business workflows help organizations coordinate tasks between people and synchronize data between systems, with the ultimate goal of improving efficiency and responsiveness.

Workflow stages show the status of a record in the workflow and provide processing that must occur for the record to move to the next phase. The tasks, information or documents are passed from one status to another (from one participant to another for action) either manually or triggered by business rules or actions specified for that specific status.

Business rule transitions enforce business logic, such as checking if the user has entered the required data or has proper security to perform a transition.

FintechOS Studio comes with the [Business Workflows Processor](#) pre-installed.

# Business Formulas

Formulas process different inputs from your digital journeys (such as income, age, assets, risk class, etc.) in order to generate desired outputs (such as credit scores, insurance premiums, interests, etc). It is mainly used to define mathematical and logical calculations that support various business needs.

FintechOS [Business Formulas](#) allow users to create advanced computations based on data imported from Excel.

You can implement multi-step calculations, using a variety of data types and built-in functions. An embedded data sets feature allows you to reference predefined value mappings between sets of discriminants (such as age, sex, driving experience, etc) and specific metrics (such as a risk coefficient).

The advantages are:

- Ability to import large Excel data sets such as risk matrices.
- IntelliSense assistance for writing your formulas in the Formula editor.
- Large array of built-in functions.

# Scheduled Jobs

In FintechOS Studio, you can set an automation script to run periodically by using the Scheduling feature. A [scheduled job](#) can be set up to run at a fixed interval of time starting with Start Time (Pool Time) or using a Cron Expression.

**Pool Time** - Refers to the number of threads that can be pooled at once by the underlying Thread Pool Executor the notional number of methods that can be run at the same time.

**Cron Expression** - Cron runs as a daemon process. This means it only needs to be started once and it will keep running in the background. This process makes use of crontab to read the entries of the schedules and kicks off the tasks. Each line in cron is an entry with an expression and a command to run. This entry runs the mentioned script every single minute.

---

## Digital Documents

Digital Documents allow you to automatically generate dynamic, customized documents that integrate real-time business data, such as contracts, agreements, and other essential paperwork. These documents can be used to replace paper-based processes in digital workflows such as customer onboardings, account openings, insurance policy applications, or loan originations.

For more information, see the [dedicated Digital Documents documentation](#).

## Scheduled Jobs

Scheduled jobs allow you to run server automation scripts automatically. To run jobs, assign a user to the JobServer in Configuration Manager.

To schedule a server automation script, follow these steps:

### 1 Add a scheduled job

1. From the FintechOS Studio main menu, click **Automation Blocks > Scheduled Jobs**. The Scheduled Jobs List page appears.
2. Click **Insert**. The Add Scheduled Job page appears.

3. Fill in the mandatory fields (marked with a red asterix). Use the **Schedule Type** field to choose between two types of scheduling:

- **Cron Expression** - Use a Quartz cron expression to schedule the job.
- **Pool Time** - Runs the job repeatedly at a preset time interval. The start date must be set in the future.
  - Pool Time - Time in seconds between job reruns.
  - Repeat Time - When set to -1 will run indefinitely. When set to a positive value (e.g. 5) will run for only 5 times at each pool time.

### IMPORTANT!

The date/time fields refer to the timezone set up on the database (for Azure environments, this is usually UTC), not the date/time settings on your local system.

4. (Optionally) If you want to set up specific days when the job will not execute, in the **Calendar (exclude days)** field , provide the code to do so. The types of calendar exclusion can be: ANNUAL (dd.MM), MONTHLY (dd), WEEKLY (days in week: Monday, Tuesday, .etc), HOLIDAY (dd.MM.yyyy).
5. Click **Save and reload**. The record is saved in the system and the Edit Scheduled Job page appears.
6. Use the **Send Notification On Success** and **Send Notification On Error** checkboxes if you wish to send email notifications when a scheduled job succeeds or fails to a preset email address.
7. Tick the **Enabled** checkbox to enable the scheduled job, otherwise it will not run.

Now you can add schedule services.

innovation studio

Administrator

EN

←

Save and close

Save and reload

Edit Scheduled Job

Run Now

Name

FTOS\_IntegrationProcess

Start Time

17/07/2019 03:00

End Time

17/07/2025 03:00

Schedule Type

Cron Expression

Cron Expression

0 0/1 \* 1/1 \* ? \*

Every minute

Calendar (exclude days)

1

Send Notification On Success

Send Notification On Error

Enabled

Schedule Services

+ Insert

× Delete

Export

Refresh

Execution Order

Name

RunAsyncInstances

Scheduled Job

Export

Refresh

Name

Last Run

Last Successful Run

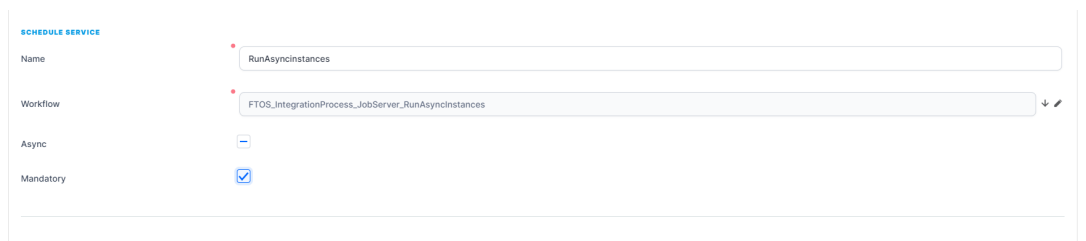
Last Failed Run

TABLE OF CONTENTS

415

## 2 Add schedule services

1. In the Edit Scheduled Job page, scroll-down to the Schedule Services section and click the **Insert** button. The Add Schedule Service page appears.
2. In the **Name** field, provide a name for the service.
3. Click the down arrow next to the Workflow field. A pop-up appears listing all existing server automation scripts.
4. Select the server automation script you want to schedule by double-clicking on it.



**SCHEDULE SERVICE**

Name: RunAsyncInstances

Workflow: FTOS\_IntegrationProcess\_JobServer\_RunAsyncInstances

Async: ☐

Mandatory: ☒

5. Tick the **Async** checkbox if you want the service to run asynchronously.
6. Tick the **Mandatory** checkbox if you wish to prevent any subsequent services of the scheduled job from running if this service fails.
7. Click **Save and close**. Add as schedule services as automation scripts you want to schedule for execution.

If you have more than one scheduled service in the Schedule Services list, you can set their execution order.

## 3 Set the execution order

If you have several automation scripts scheduled and need them to be run in a specific order, drag and drop the records in the Schedule Services list in their execution order (where the first record in the section is the first one to be executed).

If you choose that one of the service is mandatory and it fails, all the following services (scripts) will no longer execute. Also, if you will choose to run script async (when adding /editing a schedule service by selecting the **Async** checkbox), the order of services will be disregarded and all automation scripts will run in parallel.

```
[{
  name: "calAnual",
  type: "ANNUAL",
  excludeDays: ["03.11", "17.01"]
},
{
  name: "calMonthly",
  type: "MONTHLY",
  excludeDays: ["25", "10"]
},
{
  name: "calWeekly",
  type: "WEEKLY",
  excludeDays: ["Saturday", "Sunday"]
},
{
  name: "calHoliday",
  type: "HOLIDAY",
  excludeDays: ["25.12.2018", "01.01.2019"]
}
]
```

#### NOTE

In order for the cron jobs to trigger the automation script execution, the Job Server should be installed on deployment.

The Job Server is aware of jobs changes (time, cron expression, reorder of services).

For all failed jobs, a **Run now** button is displayed which allows running the jobs again. The job data model has includes a parent job id. When running a failed job (by clicking the **Run now** button), a copy of the job is created having the parentJobId = original Job and it will be scheduled to run only once in 1 minute.

## Assign Job Server User

Before attempting to run scheduled jobs, go to [Configuration Manager](#) > qa > jobserver-serverscripts > app-settings > JobServerUserName. Add the user for running the jobs:

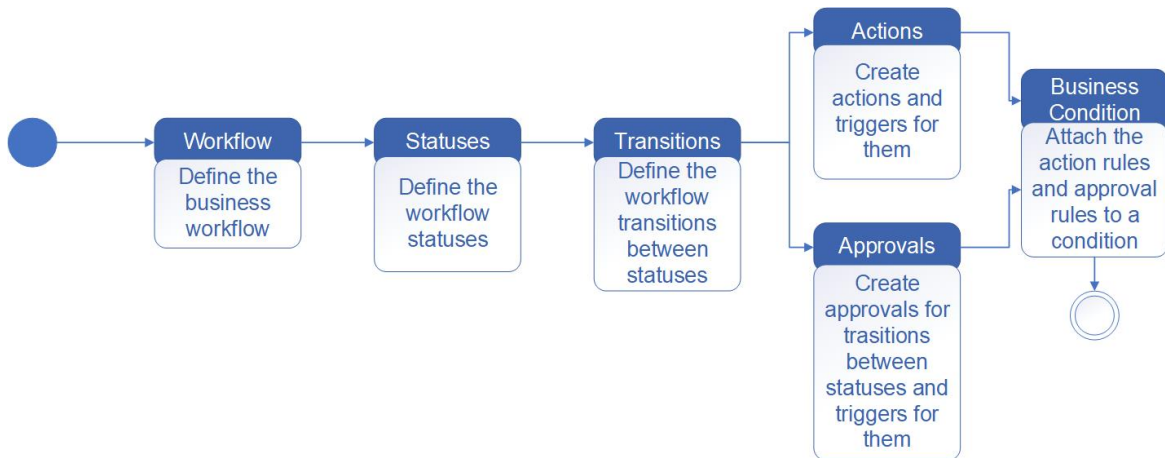
- can be a non-admin [user](#)
- must be active
- have the necessary roles for calling SDK methods in server scripting
- must not be a service account, b2c, or temporary user
- must have the Job Server [security role](#) assigned

## Business Workflows

Business workflows are sets of predefined statuses and rules to transition between those statuses that you can apply to your entity records. Business workflows are reusable, allowing you to associate the same workflow to multiple entities and to centralize the workflows' management.

Business workflows can follow a specific business logic (e.g.: transitioning an account record through statuses like prospect, lead, customer, closed, etc.) and are integrated with features like ["Form Actions" on page 258](#), ["Analytics" on page 449](#), or ["Entity Cloning and Versioning" on page 1261](#) that simplify the development of advanced capabilities.

You can also ["Define Workflow Transition Approvals" on page 445](#) and store audit information related to status transitions to implement certified and documented business processes.



## Set Up Business Workflows

FintechOS Studio enables you to define, execute and automate business processes, passing tasks, information or documents from one participant to another for action, according to a set of rules. Business workflows comprise of statuses, transitions, and business rules (herein referenced as workflow rules as well).

To set up a business workflow, follow these steps:

### Create Business Workflows

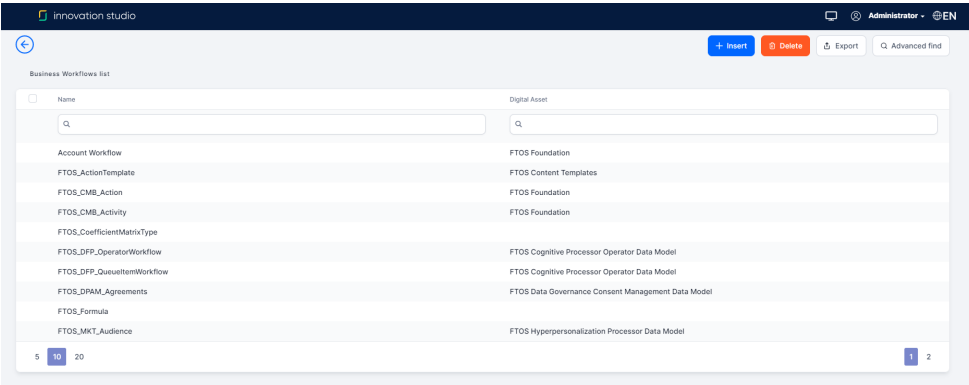
To set up a business workflow with subsequent statuses, transitions and rules, you need to create the business workflow in FintechOS Studio. Then, you need to define the workflow statuses, the transitions between statuses, and the rules.

Once you define all the business workflow transitions, if some business transitions are set to be automatically triggered by specific rules, you need to define the [workflow rules](#).

To do so, follow the steps below:

## Define Business Workflows

1. In FintechOS Studio main menu, click **Automation Blocks > Business Workflow > Business Workflows Designer**. The Business Workflows List page appears.

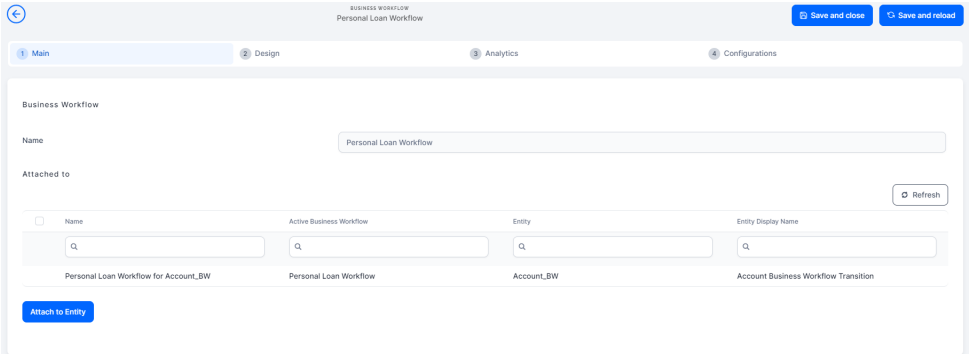


2. Click **Insert**. The **Add Business Workflow** page appears.
3. Enter the **Name** of the business workflow.

Business Workflow

Name

4. Click **Save and Reload**. The **Edit Business Workflow** page appears.
5. On the left side of the grid there is a **Attach to entity** button which opens the list of entities. Select the entity for which the workflow is built.



- Click **Save and reload**.

## Define Workflow Statuses

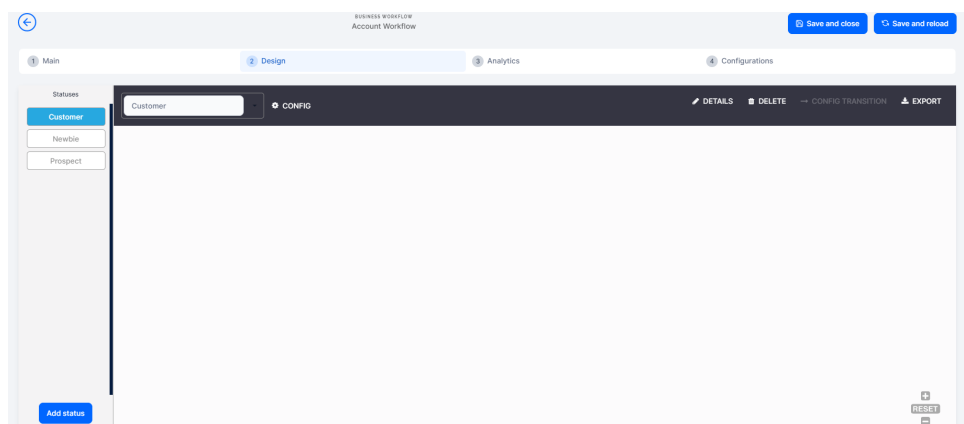
You can define workflow statuses either by using the **Workflow Designer** tab or by the **Configurations** tab. This section describes how to add a business workflow status from both sections.

### HINT

Whether you add a status from the Designer or Configurations tab, they are automatically updated in each tab to reflect the situation properly. Therefore, any modification in each tab is propagated through both places.

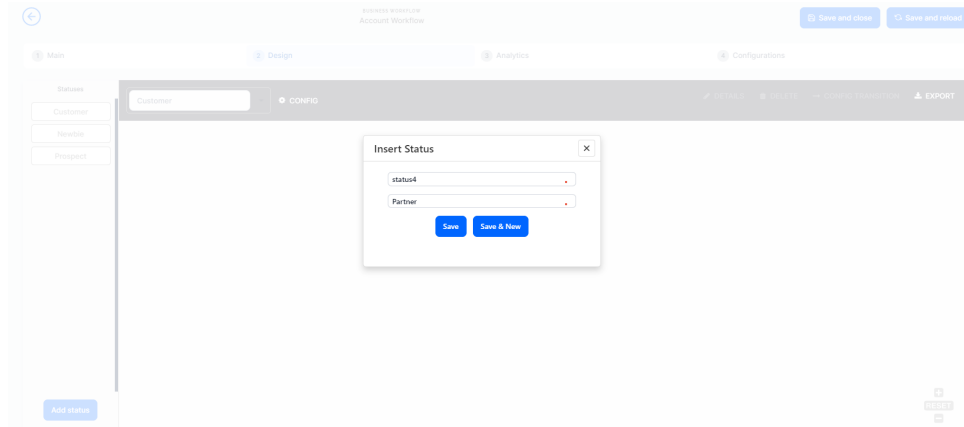
#### Add statuses from the Designer

- After having created the workflow and attaching it to the proper entity, click the **Designer** tab.
- On the left side of the screen, there is a panel which contains the list of status created. To add a status, click the button on the left panel at the bottom which reads **Add status**. It will open a pop-up window.



- In the **Name** field, enter the status name that will be used by the system.

4. In the **Label** field, enter the name of the status that will be displayed in the UI.



5. Click **Save and Reload**.

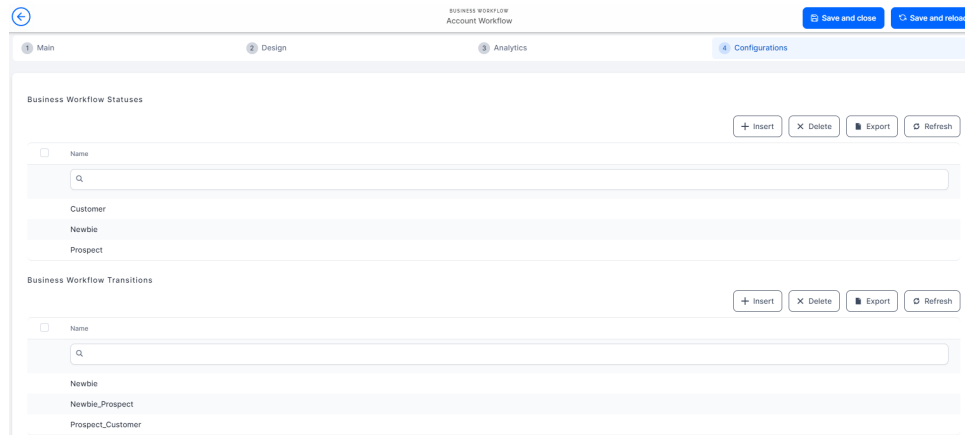
Follow the procedure above as many times as statuses your business workflow has.

After you finish adding the business workflow statuses, you need to define the workflow transitions.

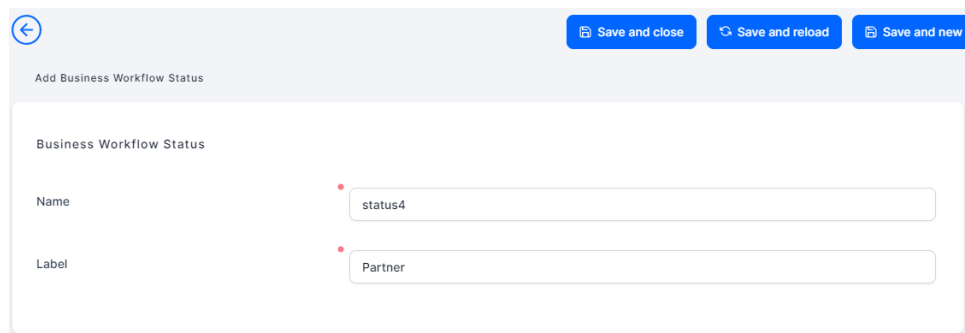
#### Add statuses from the Configurations tab

The second method is available through the Configurations tab. The first grid named **Business Workflow Statuses** shows the statuses list. To add one from here:

1. Click the **Insert** button.



2. Fill in the **name** and **label** for it.



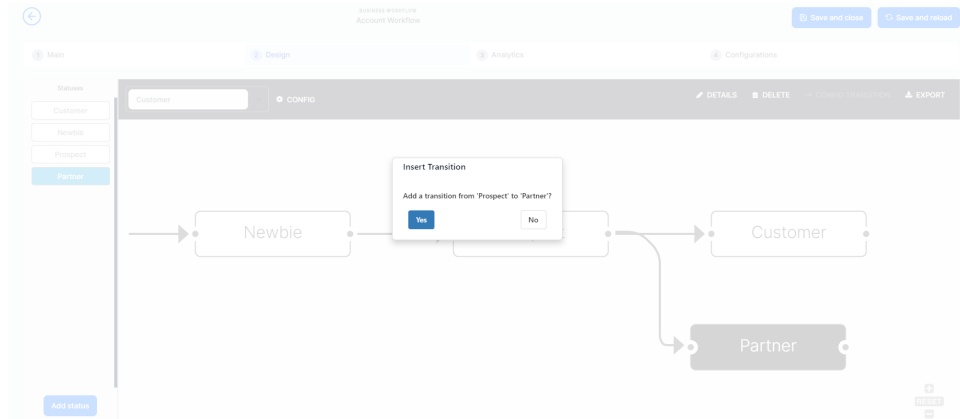
3. Click the **Save and reload** button. Repeat as many times as needed.

## Define Workflow Transitions

You can define workflow transitions either from the **Configurations** tab or by using the **Workflow Designer**.

### Add a transition from the Designer tab

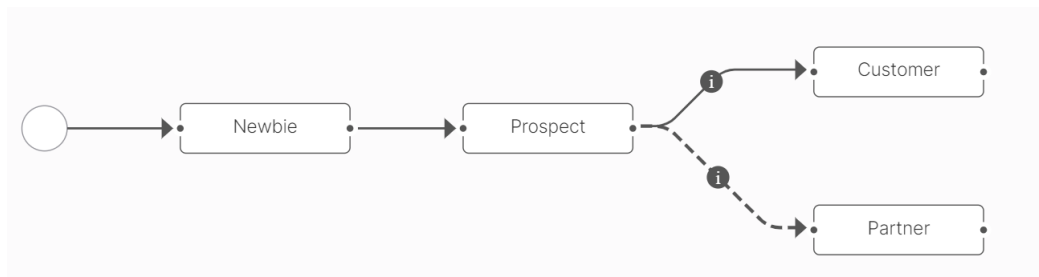
1. Drag and drop the states you wish to create a transition between. Drag them from the left panel to the white panel.



2. Click **Yes** to agree to the creation of the transition.
3. After it has been created, to configure additional features (**action groups and approvals**) select Configuration transition from the right side of the designer.

4. To see how to configure these tabs, access [Edit Workflow Transition Configuration](#).

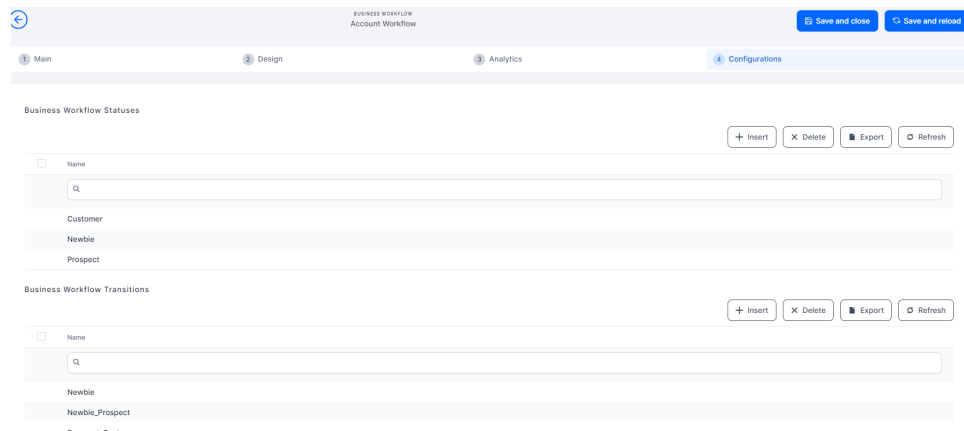
The design of the lines form the diagram marks different configurations.



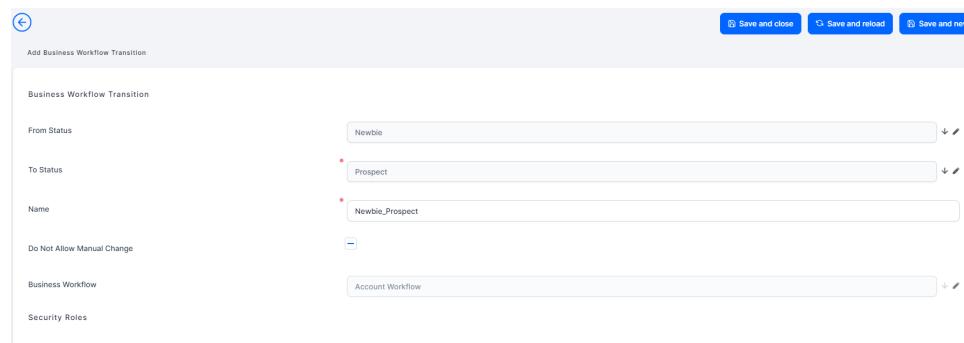
The dash types mark the configurations. The unidirectional arrow with a straight line is a transition with no configurations, e.g. Newbie to Prospect. The unidirectional line with an "i" symbol signifies a transition with an action group, e.g. Prospect to Customer. The dash line signifies a true boolean **do not allow manual transition** between the two statuses. e.g. Prospect to Partner.

#### Add a transition from the Configurations tab

1. In the Edit Business Workflow page, click on the Configurations tab.
2. Click **Insert** for the **BUSINESS WORKFLOW TRANSITIONS** grid. The Add Business Workflow Transition page appears.



3. From the **From Status** field, select the status from which the workflow transitions.



4. From the **To Status** field, select the status to which the workflow transitions.

5. The name is automatically filled in with the concatenation of the names entered in the **From Status** and **To Status** fields. If this is the initial workflow transition it follows the following naming convention: "*\_ToStatus*". However, it can be changed to the desired name for the transition.
6. Tick the **Do Not Allow Manual Change** checkbox if you want the workflow transition to be automatically triggered based on strict business rules.
7. The **Business Workflow** field is non-editable and displays the name of the business workflow for which you define the transitions.
8. If you allowed manual changes (ticked clear the **Do Not Allow Manual Change** checkbox) and want all users to manually perform the workflow transition, skip steps 2 and 3, go to step 4, for example.
9. If you want only users with specific security roles to perform the business status transition, click **Save and Reload**. The **Security Roles** grid appears.

Edit Business Workflow Transition

Business Workflow Transition

From Status

Newbie

To Status

Prospect

Name

Newbie\_Prospect

Do Not Allow Manual Change

☐

Business Workflow

Account Workflow

Security Roles

+ Insert existing

X Remove existing

☐

Name

Q

test

10. To add a specific security role, from the Security Roles section, click the **Insert existing** button and from the pop-up that appears, double-click the security role. Repeat this step to add as many security roles as your business requires.

**NOTE**

If you ticked the **Do Not Allow Manual Change** checkbox, users will not be able to manually change the status transition; it will be automatically triggered based on strict business rules.

11. Click **Save and Close**.

Follow the procedure above as many times as business status transitions you need to meet your business needs. If you want to visually see the workflow statuses and transitions, click **Designer**. The **Workflow Designer** appears.

## Attach Workflow to Entity

FintechOS Studio enables you to provide extensive configuration of the business workflow used on a specific entity, by defining the set of rules to be automatically triggered on workflow status change and also define specific actions to be performed before or after status change.

### Select business workflow on entity

To configure the workflow rules, you first need to select the business workflow you previously created on the entity where you want to use it. To do so, follow these steps:

1. In FintechOS Studio main menu, click **Evolutive Data Model> Data Model Explorer**. The Business Entities List page appears.

2. Double-click on the desired entity. The **Edit Business Entity** page appears.

**Edit Business Entity**

**Details**

Entity Type: Platform Data

Name (only use for add entity): Account

DisplayName: Customer

DisplayCollectionName: Customers

Description:

PrimaryAttributeName (only use for add entity): Name

PrimaryAttributeDisplayName (only use for add entity): Name

Default Entity Status: Draft

3. In the **Business Workflow** field, select the business workflow to be used on the business entity.
4. Click **Save and close**.

A new record is added in the Business Workflows Configurations List page (**Fintech Automation > Business Workflow > Business Workflow Configurations**) with a name that follows this convention: "<the name of the business workflow> for <the name of the entity on which the business workflow was selected>".

## Manage Business Workflows

The Workflow Designer allows you to edit business workflows by editing or deleting both statuses and transitions.

### Edit workflow statuses

To edit a workflow status, in the **Design Area**, click on the status. In the **Details Panel**, click the **Edit** button. The Edit Business Workflow Status page appears. You can edit both the name and label. Make the desired edits, then at the top-right corner of the page click **Save and Close**.

### Edit workflow transitions

To edit a workflow transition, in the **Design Area**, click on the transition. In the **Details Panel**, click **Edit**. The **Edit Business Workflow Transition** page appears. You can change the transition from one status to another, allow manual change on workflow transition or add security roles to restrict users access who manually change the workflow status. Make the desired edits, then at the top-right corner of the page click **Save and Close**.

### How workflow transitions are processed in the database

Workflow transitions triggered from the FintechOS Portal generate two separate database entries on the record, one for saving the form data and one for saving the workflow status information (previous and current status).

When changing the workflow status using the [ftos.businessStatusWorkflow.update](#) function, only the workflow status information is saved in the database entity record.

#### IMPORTANT!

"Event Triggered Automation Scripts" on page 1163 that run on entity updates may trigger either once or twice, depending on whether the workflow status is changed using the [ftos.businessStatusWorkflow.update](#) function or from the FintechOS Portal.

### Delete workflow statuses and transitions

To delete a workflow status, from the Design Area, select the status and in the **Details** panel click **Delete**. A confirmation dialog appears. Click **Yes** to confirm deletion.

The status is deleted and so do all transactions to / from the status.



To delete a workflow transition, select the transition arrow and in the Details Panel click **Delete**. A confirmation dialogue appears. Click **Yes** to confirm deletion. The workflow transition is removed.

### Configure Business Workflows

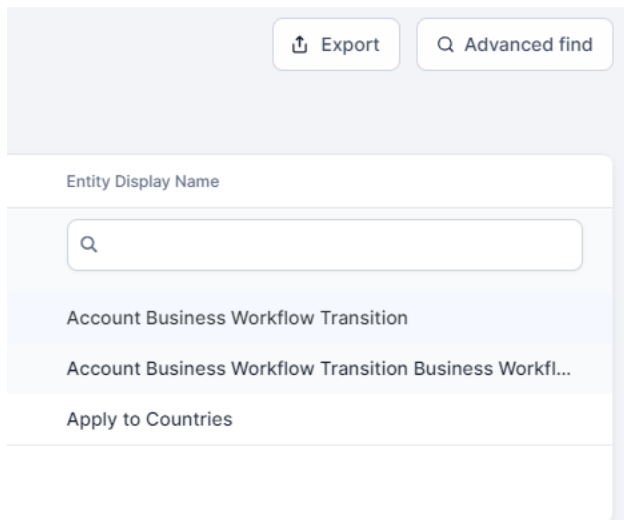
This main menu entry opens the **Business Workflow Configurations List**. From here it is possible to:

- deactivate the workflow
- insert a business condition
- access the transition configurations
- insert data management.

Business Workflow Configurations list				
<input type="checkbox"/>	Name	Active Business Workflow	Entity	Entity Display Name
	<input type="text" value="loan"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input checked="" type="checkbox"/>	Personal Loan Workflow for Acco...	Personal Loan Workflow	Account_BW	Account Business Workflow Tran...
<input type="checkbox"/>	Personal Loan Workflow for Acco...	Personal Loan Workflow	Account_BW_BWA	Account Business Workflow Tran...
<input type="checkbox"/>	Personal Loan Workflow for FTO...	Personal Loan Workflow	FTOS_HolidaysXCountry	Apply to Countries
5	10	20		

**IMPORTANT!**

The configurations from this page cannot be inserted, only exported. They are created and listed as in the picture above when the Business Workflow is created using the menu entry named **Business Workflow Designer**.



By double-clicking on one of the names from the Business Workflow Configurations List, it opens the following tabs and fields. The name is generated by the system when the workflow is created. The name has the structure: Name of the workflow for the name of the entity, e.g. Loan\_Workflow for FTOS\_AB\_Account.

## General tab

1. Double-click on one of the entries from the list.
2. The **General** tab is opened. It holds the following fields:

 A screenshot of the 'General' tab configuration for a Business Workflow. The interface has four tabs: 'General' (selected), 'Business Conditions', 'Transition Configurations', and 'Data Management'. The 'Business Workflow Configuration' section contains the following fields:
 

- Name:** A text field containing 'Personal Loan Workflow for Account\_BW'.
- Allow Only Defined Transitions:** A checkbox that is currently unchecked.
- Business Workflow is active:** A checkbox that is currently checked, with a 'Deactivate' button to its right.

- **Name:** Name of the workflow for the name of the entity, The user can change it to be more suggestive.
- **Allow only defined transitions:**
  - If **true**, then only the defined transitions between statuses will be visible in the entity form, i.e. that if there are four statuses such as Prospect, In progress, Approved and Denied, and two transitions between Prospect\_In Progress and between In Progress and Approved, it will not be possible to add other transitions via code snippets.
  - If **false**, then transitions that have not yet been defined can be via code snippets or other methods.
- **Business Workflow is active:** The workflow is active for the entity it is attached to. To deactivate it, click on the button labelled Deactivate on the right side of the panel.

3. Click **Save and close**.

## Business Condition tab

This tab attaches a condition to the workflow.

Name	Code	Description
Condition1	2345678	

1. Click **Insert**. The **Add Business Condition** page opens. This page can be accessed as well from the main menu as explained in this page [Business Rules](#).

Edit Business Condition

Business Condition

Name

Code

Description

Rule

And +

Income Equals 0

Approval rules

Transition rules

2. Fill in the following fields:

- **Name:** Insert a suggestive name.
- **Code:** Insert a unique code
- **Description:** Insert a suggestive description.
- **Rule:** Add a rule to be executed to trigger a status change or approval, e.g. Savings equals \$400.

3. Click **Save and close**.

4. Two grids open: **Approval Rules** and **Transition Rules**.

## Transition Configuration tab

In this tab, the list of transitions is displayed.

1 General

2 Business Conditions

3 Transition Configurations

4 Data Management

Export

Refresh

Name

Newbie

Newbie\_Prospect

Prospect\_Customer

**IMPORTANT!**

On this page you can only edit a transition.

1. Click on one of the transitions displayed. The **General** tab opens.

The screenshot shows the 'General' tab of the 'Entity Workflow Transition Configuration' interface. At the top, there are three tabs: '1 General' (selected), '2 Action Groups', and '3 Approvals'. Below the tabs, the main content area is titled 'Entity Workflow Transition Configuration'. It contains a 'Name' field with the value 'Newbie\_Prospect' and a 'Requires Approval' checkbox which is currently unchecked.

2. The name cannot be changed. The **Requires Approval** boolean marks whether the transition has to be approved by a back-end officer. If the boolean is true, then the change in status can only be performed if the back-end officer approves it.
3. Click the **Action Groups** tab. Click **Insert** to add a new action group.
4. The **General** tab opens.
5. Insert a **name**, **code** and **description** for the Transition Action Group.
6. Click **Save and reload** to display the following tabs: **Actions Before** and **Actions After**.
7. Click the **Actions before** or the **Actions after** tabs. Click **Insert** to add a new entry. The **General** tab opens.

Entity Workflow Transition Action

Name

Workflow Transition Action\_before

Code

123456

Description

Demo

Action Stage

Before

Apply Always

☒

Action Type

Go to Business Workflow Status

Business Workflow Status

Customer

Message

Congratulations, you are now a customer!

Fields	Description
Name	Insert a suggestive name of the action.
Code	Insert a unique code for the action.
Description	Insert a suggestive description for the action.
Action Stage	This field is read-only. It is automatically filled in by the system with Before or After depending in which grid the user wishes to insert the action and its trigger.
Apply Always	If <b>true</b> , then the action and its trigger (later applied) will always execute.
Action Type	Select from the list: <ul style="list-style-type: none"><li>Go to Business Workflow Status</li><li>Run Custom Script</li><li>Cancel Transition.</li></ul> depending on what is selected from here, then the following fields will be displayed.
Go to Business Workflow Status	

Fields	Description
Business Workflow Status	Select from the list the status to which the action points.
Message	Insert a suggestive message to be displayed to the user in the FintechOS Portal.
Run Custom Script	
Custom Script	Insert the script to run when it is triggered.
Cancel Transition	
Error message	Insert a suggestive message to be displayed to the user in the FintechOS Portal.

1 General
2 Action Triggers

Entity Workflow Transition Action

Name Workflow Transition Action\_before

Code 123456

Description Demo

Action Stage Before

Apply Always ☒

Action Type Go to Business Workflow Status

Business Workflow Status Customer

Message Congratulations, you are now a customer!

8. Click **Save and reload**, the **Action Triggers** tab will be displayed.

9. Click **Insert** . The **Add Action trigger** page will open.

10. Fill in the following fields:

- **Name:** Insert a suggestive name for the trigger.
- **Business Condition:** Select from the list the condition to be applied.
- **Required Evaluation Value:** Select from the list the value to which you wish the condition to be assessed:
  - True
  - False

11. Click **Save and close**.

12. Navigate to the **Approvals** Tab which corresponds to a transition.

13. Click **Insert**. For details on configuring Approvals, see "[Define Workflow Transition Approvals](#)" on page 445.

## Data Management

1. Click the **Insert** button from the grid. The page named **Add Record Protection by Business Workflow Status** opens.
2. In the field **Business Workflow Status**, select from the lookup the status to be protected.

3. In the field **Record Action Type**, select from the lookup the protection level:

- update
- delete.

4. Click **Save and close**.

### Edit Workflow Transition Configuration

You can add specific actions to be performed before or after status change by editing the workflow transition configuration.

It is possible to access the same page following two paths:

- From the main Studio menu, navigate to **Automation Blocks > Business Workflow Configurations**. Next, open the workflow you are working on from the list and click on the third tab named **Transition configurations**. It displays a list of all the transitions created. Select the transition you are interested in. The **ENTITY WORKFLOW TRANSITION CONFIGURATION** page opens.

- From the main Studio menu, navigate to **Fintech Automation > Business Workflows > Business Workflow Designer**. Next, open the workflow you are working on from the list and click the **Designer** tab. Then, select the transition and click the **Config**

**Transition** arrow on the right side of the screen. The **ENTITY WORKFLOW TRANSITION CONFIGURATION** opens.

## Add action group

1. In the **Edit Workflow Transition Configuration** the **General** tab contains the name of the transition and the **Requires Approval** bool.

The screenshot shows the 'Entity Workflow Transition Configuration' form. The 'General' tab is active. The 'Name' field contains 'Newbie\_Prospect'. The 'Requires Approval' checkbox is unchecked. The breadcrumb trail at the top is: ENTITY > Account > BUSINESS WORKFLOW > Account Workflow > ENTITY WORKFLOW CONFIGURATION > Account\_Account Workflow. There are 'Save and close' and 'Save and reload' buttons at the top right.

2. In the Transition Action Groups section, click **Insert**. The **Transition Action Group** page appears.
3. Enter the **Name** of the action group, the **Code** and the **Description** which will be used by the system.
4. Click **Save and reload**. The **Actions before** and the **Actions after** tabs will become available.
5. Add actions to be performed on workflow transition. The procedure for adding actions is the same for both before and after actions.

## Add workflow transaction actions

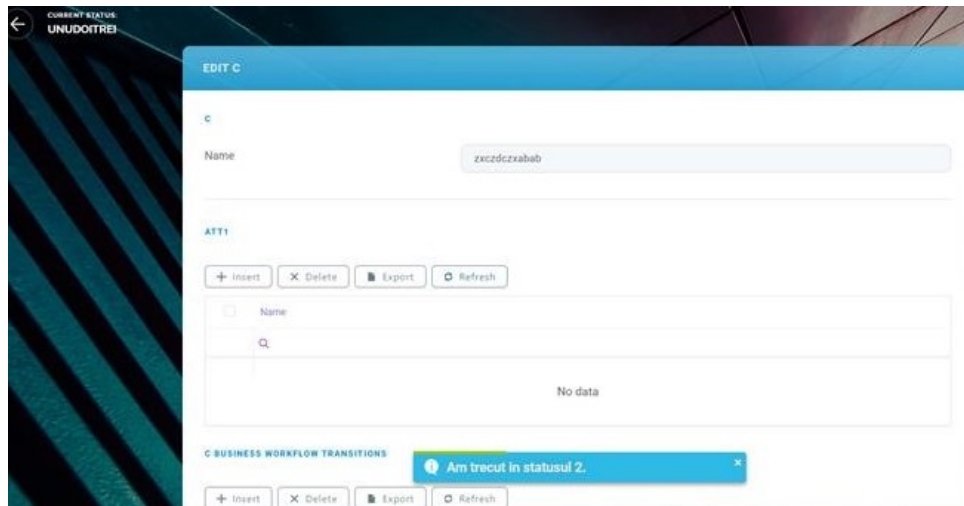
You can add before, after or before and after business workflow status change by adding actions in the Entity Workflow Transaction Actions Before section, respectively Entity Workflow Transaction Actions Before section, or both. The procedure for adding actions is the same:

The screenshot shows the 'Entity Workflow Transition Action' configuration page. It includes the following fields and values:

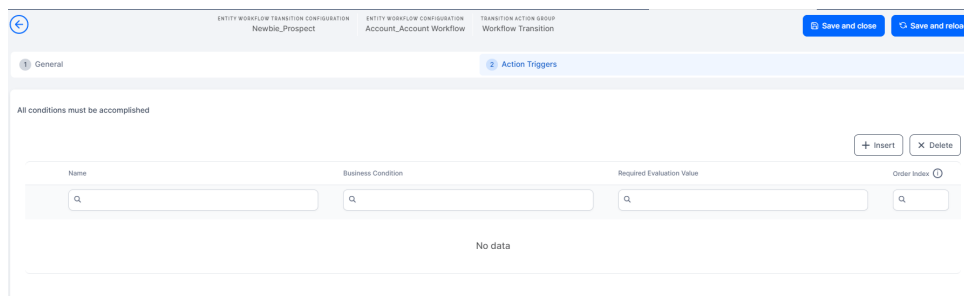
- Name:** changeStatus
- Code:** 65
- Description:** (empty text area)
- Action Stage:** Before
- Apply Always:** ☐
- Action Type:** Go to Business Workflow Status
- Business Workflow Status:** Prospect
- Message:** Your status has been updated. Congrats!

1. Click **Insert** from the section corresponding to when you want the action to be performed (before or after status change). The **Add Entity Workflow Transition Action** page appears.
2. Enter the **Name** of the action to be used by the system.
3. Enter the **Code** of the action. It must be unique.
4. Enter the **description** of the action.
5. The **Action stage** is displayed: before/ after. The field is automatically filled in.
6. Tick the **Apply Always** checkbox to ensure business consistency,
7. Select the **Action Type** from the following available values:
  - **Go to Business Workflow Status.** You have to select the **Business Workflow Status** to which the workflow transitions.
  - **Custom Script.** You have to enter the script **Code** to be executed on workflow transition.
  - **Cancel Transition.** You have to enter the **Error Message** to be displayed if the business condition fails.

8. Enter the **message** for the user to read when the action is triggered. For example, this is how it will be rendered in the FintechOS Portal when the action is triggered and the message is displayed at the bottom of the page.



9. Click **Save and Reload**.
10. Add as many actions as you need, then add the business conditions which if met will trigger specific action execution.
11. The **Action triggers** tab will be available after the action was saved.



## Add action trigger with business condition

1. In the Edit Transition Action Group page, scroll-down to the **Action Business Conditions** section and click **Insert**. The Add Action Business Condition page appears.
2. Enter the **Name** of the action business condition to be used by the system.
3. Select the **Business Condition** from the list of previously added business conditions.
4. Select the **Required Evaluation Value**. If you want the action to be run if the business condition is met, select **True**; otherwise select **False**.
5. Click **Save and Reload**.

## Define User Competences

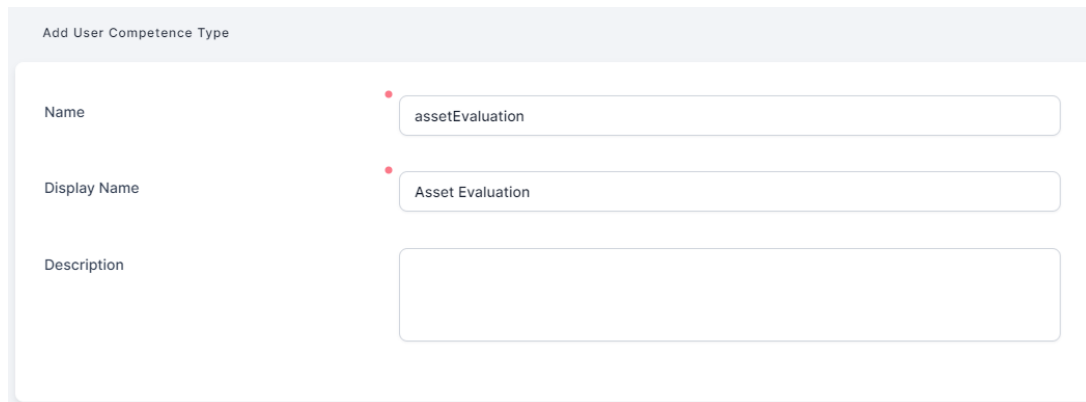
User competences allow their respective users to perform specific tasks in the system. FintechOS Studio allows you to create user competence hierarchies by defining multiple competence levels for the same user competence type.

In the context of workflow transitions approvals, competencies are used to define the users who approve or reject specific workflow transitions.

### 1 Define User Competence Types

1. In the FintechOS Studio main menu, click **Admin > User Competence Settings > User Competence Types**. The **User Competence Types List** page appears.

2. Click **Insert**. The **Add User Competence Type** page appears.



The screenshot shows a web form titled "Add User Competence Type". It contains three input fields:

- Name:** The input field contains the text "assetEvaluation".
- Display Name:** The input field contains the text "Asset Evaluation".
- Description:** The input field is empty.

3. Fill in the **Name**. This name uniquely identifies the user competence type in the system.
4. Fill in the **Display Name**. This is how the user competence type will be displayed in the user interface.
5. Optionally, fill in a **Description** for the user competence type.
6. Click **Save and Reload**.

Repeat the process for any additional user competence types you would like to add.

## **2** Define User Competences

1. In the FintechOS Studio main menu, click **Admin > User Competence Settings > User Competences**. The User Competences List page appears.

2. Click **Insert**. The **Add User Competence** page appears.

The screenshot shows the 'Add User Competence' form. The form is titled 'Add User Competence' in a light blue header. The main content area has a white background. It contains the following fields:

- User Competence Type:** A dropdown menu with 'assetEvaluation' selected. A red dot is above the field.
- Level:** A text input field with '1' entered. A red dot is above the field.
- Name:** A text input field with 'Asset Evaluation\_1' entered. A red dot is above the field.
- Display Name:** A text input field with 'Asset Evaluation\_1' entered.
- Description:** A large, empty text area.
- System Users:** A section at the bottom, currently empty.

3. Select the **User Competence Type** that this competence belongs to.
4. Enter the competence **Level** for the above user competence type. Users with higher competence levels have access to all the lower competence tasks. For instance, a level 2 approver will also have access to all level 1 approval tasks.
5. Fill in the **Name** or leave the default name in place. This name uniquely identifies the user competence in the system.
6. Fill in the **Display Name** or leave the default display name in place. This is how the user competence will be displayed in the user interface.
7. Optionally, fill in a **Description** for the user competence.
8. Click **Save and Reload**.

9. In the **System Users** section, use the **Insert Existing** and **Remove Existing** buttons to assign the competence to the desired users.

The screenshot shows the 'Edit User Competence' form. It includes fields for 'User Competence Type' (set to 'assetEvaluation'), 'Level' (set to '1'), 'Name' (set to 'Asset Evaluation\_1'), and 'Display Name' (set to 'Asset Evaluation\_1'). There is a large text area for 'Description'. Below these fields is the 'System Users' section, which contains two buttons: '+ Insert existing' and 'X Remove existing'. At the bottom, there is a search bar with a magnifying glass icon and a list of users, with 'host' currently selected.

10. Click **Save and Close**.

Repeat the process for any additional user competences you would like to add.

## Define Workflow Transition Approvals

1. Open the **Approvals** tab to fill in the approval's details by clicking **Insert**.

The screenshot shows the 'Add Business Workflow Transition Approval' form. It includes fields for 'Name' (set to 'evaluateStatus'), 'Display Name' (set to 'evaluateStatus'), 'User Competence' (set to 'Loan Approver\_1'), 'Is Advisory' (unchecked), 'Competence Approval Mode' (set to 'Any Of'), 'Rejected Business Workflow Status' (set to 'Newbie'), 'Entity Form' (set to 'accountInsertForm'), 'Rejected Text' (set to 'Customer is not approved'), 'Approval Text' (set to 'Customer is approved'), and 'Triggers Validation Mode' (set to 'Any Of').

- **Name:** This name uniquely identifies the workflow transition approval in the system.
- **Display Name:** This is how the workflow transition approval will be displayed in the user interface.
- **User Competence:** User competence of the person approving the transition. For details, see ["Define User Competences" on page 442](#).
- **Is Advisory:** Check if approval is only advisory. If an approval is advisory, the status change will take place even if the approver rejects the transition, but the rejection will be logged in the business workflow transitions list.
- **Competence Approval Mode:**
  - **Any Of:** If any of the users with the designated competence approves the transition, the transaction passes.
  - **All:** All the users with the designated competence must approve the transition before it passes.
- **Rejected Business Workflow Status:** Alternate workflow status to be applied if the transition is rejected.
- **Entity Form:** Context (entity form or form driven flow) in which the workflow transition triggers the approval process.
- **Rejected Text:** Text message to be displayed on screen if the transition is rejected.
- **Approval Text:** Text message to be displayed on screen if the transition is approved.
- **Triggers Validation Mode:**
  - **Any Of:** The transition requires approval if any of the approval triggers matches.

- **All:** The transition requires approval if all the approval triggers match.

2. Click **Save and Reload**.

3. In the **Approval Triggers** section, add any business conditions that have to be met to trigger the approval. If this section remains empty, the workflow transition will always require approval.

The screenshot shows a form titled "Add Approval Trigger". It contains the following fields and values:

- Name:** approvalTrigger
- Display Name:** approvalTrigger
- Entity Business Workflow Rule:** Condition1 (selected from a dropdown menu)
- Required Evaluation Value:** True (selected from a dropdown menu)

4. Fill in the fields:

- **Name:** Insert a name for the approval trigger.
- **Display Name:** Insert the name that will be displayed in the UI.
- **Entity Business Workflow Rule:** Select the Business Condition.
- **Required Evaluation Value:**
  - True
  - False

5. Click **Save and reload**. Repeat the process for any additional approvals you wish to add.

## Approve Workflow Transitions

When a workflow transition that requires approval is triggered, the workflow is set in a pending state until the approvers with the corresponding competences review the transaction.

If you are a user assigned with workflow approval competences:

1. In FintechOS Portal, access **Main Menu > Approval Tasks > My Approval Tasks** to view your list of approval tasks.

APPROVAL TASKS LIST							
<input type="checkbox"/>	Subject to Approval	Subject Type	From Status	To Status	Assigned User	Competence Type	Competence Lev
	Q	Q	Q	Q	Q	Q	Q
	Pot of Gold	Assets	Evaluation In Progress	Safe		assetEvaluation	
	stuff	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	asset no. 3	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	asset 666	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	asset 666	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	abcd	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	asset 1	Assets	Evaluation In Progress	Safe	host	assetEvaluation	
	asset 1	Assets	Evaluation In Progress	Safe	host	assetEvaluation	

## 2. Double click the desired approval task to review the workflow transition.

EDIT APPROVAL TASK

Task Status  
Active

Required User Competence  
Asset Evaluation\_1

Creation Date  
23/04/2020

Is Advisory ☐

Information

assets: Pot of Gold [Show more details...](#)

Assigned User

Status transition from **evaluationInProgress** to **safe** was requested by **host**

Comment

Reject

Need more info

Approve

RELATED APPROVAL TASKS

<input type="checkbox"/>	Approval Type	Required User Com...	Assigned User	Approval Decision	Comment	Decision Date	Task Status
	Q	Q	Q	Q	Q	Q	Q

No data

3. To view the workflow transition in its original context (the entity form or form driven flow in which it was triggered), click **Show more details...**
4. Optionally, type in a comment in the **Comment** field.
5. Review the workflow transition:
  - Click **Reject** if you don't want to approve the transition. This will change the record's status from Pending to the "Rejected Business Workflow Status: Alternate workflow status to be applied if the transition is rejected." on [page 446](#) defined for approval.
  - Click **Need more info** if you cannot make a decision based on the available data. This will roll back the transition and the record's status will change from Pending to the initial status. Optionally, if you wish another user with corresponding competences to review the workflow transition, select one from the **Assigned User** drop-down box, before clicking Need more info.
  - Click **Approve** to approve the transition and proceed with the record's status change.

## Analytics

This tab offers an advantage by generating a report which exposes information about the entity and the statuses it has passed through. By selecting the dates it is possible to see and pinpoint the exact transition from a status to another took place in a digital journey.

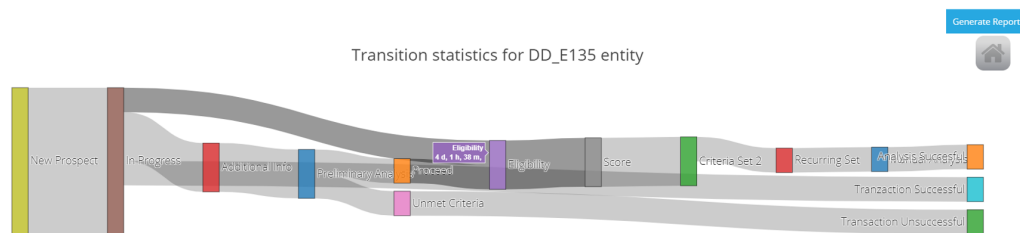
To generate a report follow the steps:

1. Open the FintechOS Studio using the user name and password.
2. Open the main menu, select **Automation Blocks**.
3. Open **Business Workflow Designer** and select the workflow you wish to modify.
4. Click the **Analytics** tab.

5. Fill in the following fields:

- **Entity:** Select the entity on which you wish to see the report.
- **Start date:** Choose the date for the report to begin on.
- **End date:** Choose the date for the report to end.
- **Business unit:** Choose the unit.
- **Include Hierarchy:** Tick the bool if you wish to include the hierarchy from the unit.

6. Click **Generate Report**.



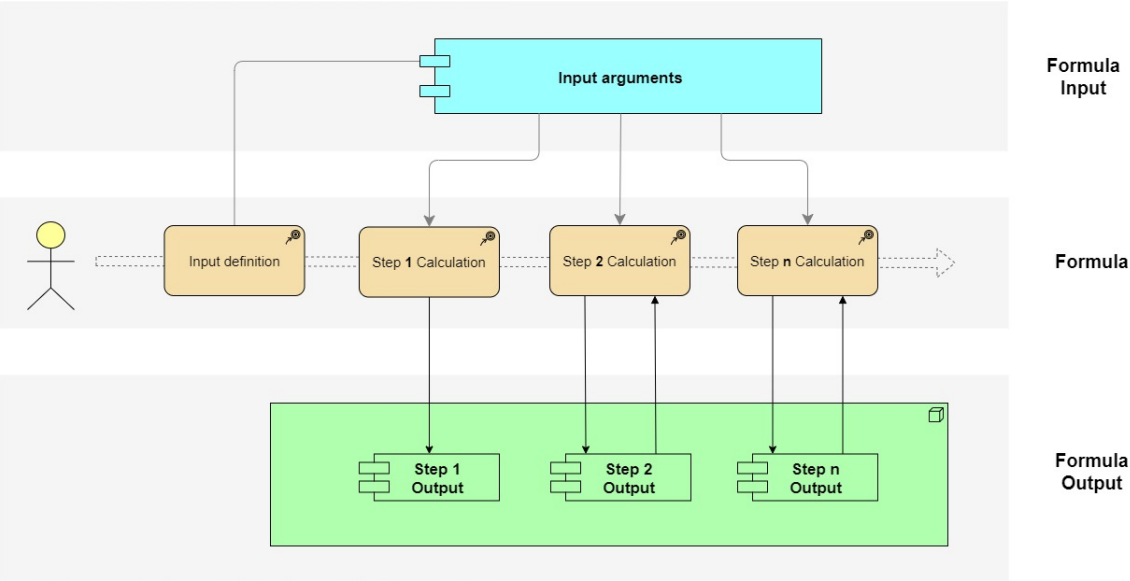
7. If you hover on the statuses you will see more details and if you click and drag them it is possible to move them where you wish.

8. By clicking the **Home** button the statuses will reverse to their initial place.

## Business Formulas

Business formulas help you process different inputs from your digital journeys (such as income, age, assets, risk class, etc.) in order to generate desired outputs (such as credit scores, insurance premiums, interests, etc). They are mainly used to define mathematical and logical calculations that support various business needs.

Formulas use a simple syntax and can incorporate several steps where the result of a step is used in subsequent steps. They can take into account data sets imported in Excel format from third-party systems.



Formula Inputs .....	451
Define Formula Expressions .....	455
Formula Editor .....	469
Data Sets .....	513
Formula Parameter Mapping .....	520
Use Business Formulas in a Digital Journey .....	521
Export a Formula or Data Set .....	523

## Formula Inputs

A formula input is a grouping of parameters that can be used on more than one formula. Each input has a set of arguments can be used by these formulas.

To define the set of arguments that go into a formula:

1. Open the Main Menu in FintechOS Studio and select **Automation Blocks > Business Formulas > Formula Input Parameters**.
2. Click **Insert**.
3. Enter a **Name** for the formula input. This name must be unique. This name will be referenced by formulas that process this specific set of arguments.
4. Optionally enter a **Description** for the formula input.
5. Click **Save and Reload**.

Alternatively, from step 4, it you can delete an input by clicking the "X" sign on the right corner of the screen.

### IMPORTANT!

Once you ["Activate a Formula" on page 463](#), you can no longer delete or update its input, even if you ["Create a New Formula Version Draft" on page 465](#). This will also apply to all Formula Arguments declared in the Formula Input.

## Add Arguments to a Formula Input

A formula argument is a single value (be it simple or collection) which will be used as input for the Business Formulas. Formula arguments are basically the parameters of the formula.

For example, for an insurance product it is possible to have as arguments the construction year of the house, the construction type, the seismic zone, the risks and the insured amount. For a banking product such as a loan, the arguments can be education, age, income, expenses, FICO score etc.

1. In the **Formula input** page, in the **Formula Arguments** section, click the **Insert** button to add an argument.
2. Enter a **Name** for the argument. This is a unique name that identifies the argument in the formula input e.g. Formula Arguments must have a unique name per each Formula Input (you can reuse argument names for different formula inputs).
3. Enter a **Display Name** for the argument. This is how the argument will be displayed in the end-user interface.
4. The **Formula Input** field is automatically populated with your formula input name.
5. Select the **Master Type** of your argument as either Simple Type (a single value) or Collection (a set of multiple values). For example, a **Simple Type - Decimal** can be the amount that the bank customer must pay in a month for a credit they have taken, while a **Collection – Decimal** represents a list containing all the monthly payments installments that the bank customer must pay in order to fully repay the credit they have taken.

6. In the **Formula Argument Sub-Type** field, enter the data type for your argument:

- Whole Number
- Decimal
- Boolean - only for Simple Type master types
- Text - only for Simple Type master types
- Object - only for Simple Type master types. When this data type is selected, an **Object Properties** field will open where you must add the object's name and data types.

For example, you can have a collection master type with a whole number sub-type for the argument moto risks e.g. for each risk, fire, vandalism, hurricane. Each risk has a number representing the level of risk, e.g. 1, 2, 3, 4.

7. Click **Save and Close** .

8. Repeat for any additional arguments you wish to include in your formula. The order for adding the arguments is not relevant.

After you've added all the arguments, click **Save and Close**.

### HINT

To change or delete a Formula Argument, you must make sure that the parent Formula Input is not used in an active formula.

After having set the parameters for the formula, it is time to set the steps for the calculation of each formula.

## Clone Formula Inputs

Cloning a formula input creates a new, independent copy of an existing formula input. This allows you to use an existing input as a template for similar use cases. To clone a formula input:

1. In FintechOS Studio, go to **Main Menu > Automation Blocks > Business Formulas > Formula Input Parameters**.
2. Locate the formula input to be cloned and double-click it to open.
3. Click the **Clone** button in the top right corner of the screen.
4. The cloned formula input is created and opened automatically.
5. Modify the cloned input as needed, then click **Save and Close**.

## Define Formula Expressions

This is where you build business formulas based on your business needs. Using pre-defined arguments, you can structure a formula into steps and test it. The formula expressions support different types of results (simple or collection).

To define a formula expression:

1. Open the Main Menu in FintechOS Studio and select **Automation Blocks > Business Formulas > Formula**
2. Click **Insert** at the top right corner of the page.
3. Enter a **Name** for the formula.
4. Select the **Start Date** when you wish to activate the formula (see ["Formula Versioning" on page 463](#)). By default, it uses the current date and time.
5. In the **Formula Input** field, select the set of arguments that will be processed by the formula (see ["Formula Inputs" on page 451](#) for details).

**NOTE**

Once you ["Activate a Formula"](#) on [page 463](#), you can no longer change the formula input.

6. The **End Date** and **Version** fields are populated automatically based on the formula's versioning. For details, see ["Formula Versioning" on page 463](#).
7. Tick the **Is Audited** checkbox if you wish to track the formula usage. This will save auditing details in the `EbsMetadata_FTOS_CALC_FormulaAudit` table every time the formula is run.

8. Click **Save and Reload**.

CURRENT STATUS: DRAFT

NEXT STATUS: ACTIVE

Save and reload

Business transactions

1 Details

2 History

3 Form Tracking

Formula

Name

docTestFibonacci

Formula Input

docTestFormulaInputs

Start date

23/02/2024 17:12

End date

Version

1

Is Audited

Formula Steps

+ Insert

X Delete

Export menu

Refresh

<input type="checkbox"/>	Name	Exclude From Mapping	Master Type	SubType	CalculationType	Exection Order
<input type="checkbox"/>	fibonacci	(All)	Simple Type	Whole Number	Iteration	1
<input type="checkbox"/>	fn	(All)	Simple Type	Whole Number	Normal	2

Formula Tests

+ Insert

X Delete

Export menu

Refresh

<input type="checkbox"/>	Name	Execution Successfull
<input type="checkbox"/>	test	

Add steps to a formula

Steps allow you to process a formula in successive stages and to use the output from a step as an input argument in subsequent steps.

**HINT**

If your calculation is complex, needs periodic update, or must be simplified for transparency and traceability, we suggest you split it in separate steps, each with its

own expression. This architecture allows calling the result of step N-1 in step N (“result = step 1 + a;”).

For example, you can create a formula that calculates the premium amount in the first step, then use the result as an input in the second step to calculate the monthly payment installments (by dividing the premium amount to the number of months).

1. In the formula screen, in the **Formula Steps** section, click **Insert** to add a step. This will open the Add Formula Step screen.

Save and closeSave and reload

Edit Formula Step

Formula Step

Name

fibonacci

Formula Id

docTestFibonacci

Exclude From Mapping

Output type

Master Type

Simple Type

SubType

Whole Number

Execution type

CalculationType

Iteration

Number of Iterations

n

Formula

1 result[0] = 0;

2 result[1] = 1;

3 result[i] = result[i-1] + result[i-2];

2. Enter a **Name** for the step. The name must be unique in the formula (Two steps can be named the same, but must be in different formulas.) This name can be used in subsequent steps as an input argument.
3. The **Formula Id** field is automatically populated with your formula name.
4. Check the **Exclude From Mapping** checkbox if you want to disable the output parameter mapping for the step (see ["Formula Parameter Mapping" on page 520](#) for details). This is useful, for instance, if you wish to safely test your formula without affecting entries in the database.
5. In the **Master Type** field select the data type for the step's result. This can be either Simple Type (a single value) or Collection (a set of multiple values).  
For collections, you will have to specify a calculation type based on an iteration (see below).
6. In the **Sub-Type** field, enter the data type for the step's result:
  - Whole Number
  - Decimal
  - Boolean - For simple master types only.
  - Text
  - Object - For collection master types only.
  - Date Time

Data type	Example
Whole number	30000 - income in EUR
Decimal	350.78 - rent expenses in GBP
Boolean	true - is married
Text	Construction type e.g. "wood"

Data type	Example
Object	It is a grouping of arguments that are connected. E.g.: "Age": "WholeNumber", "Salary": "Decimal", "Education": "Text", "IsMarried": "Boolean"

7. **Calculation Type** can be set as either Normal or as an Iteration. Iterations allow you to define collections dynamically, based on an iteration counter, using the following format `result[i] = ...`. For details, see ["Dynamic Collections Definitions" on page 472](#).
8. The **Number of Iterations** is applicable only for iteration calculation types and sets the maximum value of the iteration counter. You can use either a formula input argument of type whole number, or the output from a previous formula step.  
If the output of such a formula step is a collection, the counter will be equal to the collection's length. E.g.: If the previous formula step outputs a collection with five items, the number of iterations for the current step will be 5.
9. Enter the expression for the formula step in the **Formula field**. For details, see ["Formula Editor" on page 469](#).
10. Click **Save and Close**.
11. Repeat for any additional steps you wish to include in your formula. The output from a step can be used as an input argument in subsequent steps.

**NOTE**

New formula steps cannot be added on an active formula. For this, you need to create a new version of the formula (see ["Formula Versioning" on page 463](#)).

After you've added all the steps, click **Save and Close**.

**HINT**

The order used for each step is indicated in the Oder Index column on the right in the grid.

## Test Your Formula

You can create various tests for a formula to see how it performs.

To create a formula test:

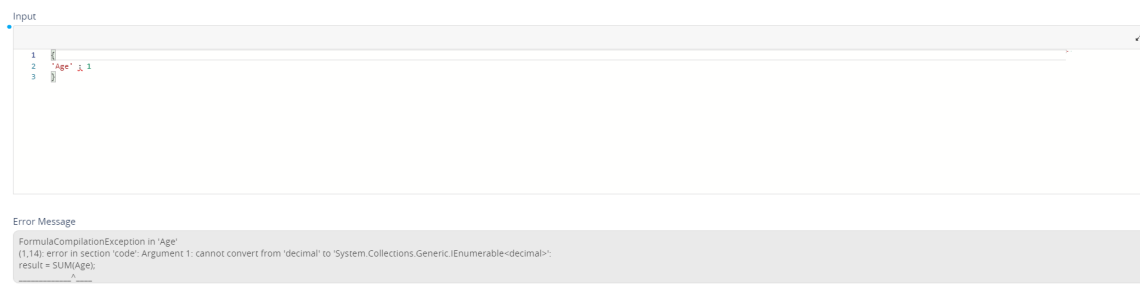
1. In the formula screen, in the **Formula Tests** section, click the **Insert** button to add a test.
2. Enter a **Name** for your test.
3. Enter a **Reference Date** for the test. By default this will be the current date and time. This allows you to test your formula with various data set versions, for instance with a data set that will become active in the future (see "[Data Set Versioning](#)" on page 518).
4. The **Formula** field will be automatically populated with the formula name.
5. Click **Save and Reload**. A new button will appear called "**Test Formula**".

6. Edit the **Input** field with the desired input parameters. This field is filled in automatically with the input from the formula and with default values for each argument from this input. The user can just insert some values, not necessarily arguments.
7. Click the **Test Formula** button at the top right corner of the page.
8. Check the **Execution Successful** checkbox and the **Output** field to investigate the formula execution.

If the result is successful, the output will include the formula result as well as the sequence of inputs and step results that led to it.

```
{
  "formulaResult": 34.0,
  "inputParams": {
    "n": 10,
    "fibonacci": [
      0.0,
      1.0,
      1.0,
      2.0,
      3.0,
      5.0,
      8.0,
      13.0,
      21.0,
      34.0
    ],
    "fn": 34.0,
    "docTestFibonacci": 34.0
  }
}
```

If the result is an error a new text box will appear. In the text box, the system will retrieve the reason for the error.

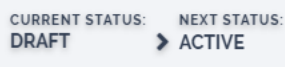


## Formula Versioning

When you first create a formula, it will be in a **Draft** state, meaning that it can be edited and tested, but not used by the system. Users cannot create a new version from this state by clicking the button "Create new version" on the right-side of the screen.

### Activate a Formula

To activate a formula draft, in the formula page, click the **Next Status: Active** field in the top left corner of the screen.



The activation stage implies new conditions for the formula:

- it is possible to create a new version of the formula
- the formula has at least one step
- the form is readonly
- when a versioned formula goes into **Active** status, the parent formula end date will be updated with the versioned formula start date and its status will be updated to **Closed**.
- you can no longer update or delete the formula's input.

### HINT

You can still run a closed version of a formula with the [ftos.formulaEngine.run](#) Server SDK method.

### **IMPORTANT!**

Once activated, you can revert an active formula to the draft status only if:

1. Its start date/time is later than the current date/time.
2. There is no other draft created in a newer formula version (see "[Create a New Formula Version Draft](#)" on the next page).

Otherwise, an active formula can only go into the **Closed** state.

### **Clone a Formula**

After activation, you can create a clone of that formula by clicking the **Clone** button at the top-right corner of the screen.

1 Details
2 History
3 Form Tracking

Formula

Clone

Create New Version

Name

test

Formula Input

Start date

09/03/2022 22:23

End date

Version

1

Is Audited

☐

Formula Steps

+ Insert

X Delete

Export

Refresh

<input type="checkbox"/>	Name	Exclude From Mapping	Master Type	SubType	CalculationType	Exection Order
<input type="checkbox"/>	test	(All)	Simple Type	Whole Number	Normal	1

Formula Tests

+ Insert

X Delete

Export

Refresh

Name	Execution Successfull
	(All)

No data

This allows you create variations based on the same formula as a starting point.

### Create a New Formula Version Draft

A formula that is currently in use cannot be modified. Instead, a **"Create New Version"** button will appear in the top right corner of your formula page. This button will allow anyone to create a new version of the same formula and a user will be able to make modifications for the new version. It will create a new version with start date today or the start date of the formula if it is in the future.

The screenshot shows the 'Details' tab of a formula in Fintechos Studio. At the top, there's a status bar with 'CURRENT STATUS: ACTIVE' and 'NEXT STATUS: DRAFT'. To the right are buttons for 'Save and close', 'Save and reload', and 'Business Transactions'. Below this is a tab bar with 'Details', 'History', and 'Form Tracking'. The 'Details' tab is selected, displaying a form with the following fields: 'Name' (test), 'Formula Input', 'Start date' (09/03/2022 22:23), 'End date', 'Version' (1), and 'Is Audited' (checkbox). On the right side of the form, there are buttons for 'Clone' and 'Create New Version'.

Click the **Create New Version** button to create a new **Draft** version based on the active formula. You can edit the draft version while the active formula is still enabled. It will have the same name as the original one, but the version number will be increased by one, and the start date will be current date, or start date of the previous version plus one day. To find the New Formula Version Draft, open the formula you wish to change and click the **History** tab where all modifications are shown.

### NOTE

The start date of the new version must be later than or equal to the current date and time.

### Activate a Formula Version Draft

Once you finish updating the draft version, change its status from **Draft** to **Active** as shown in the "[Activate a Formula](#)" on page 463 section. The previously active version will be set to a Closed state, and the draft version will become the currently active version.

You can track the formula versions in the **History** tab of the formula page.

←

CURRENT STATUS:  
ACTIVE

→

NEXT STATUS:  
DRAFT

Save and reload

Business Transactions

1 Details

2 History

X Delete

Export

Refresh

<input type="checkbox"/>	Name	Start Date	End Date	Single Value	Business Status	Version	Digital Asset
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	(All)	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Test	25/03/2024 10:37		-	Active	4	
	Test	24/03/2024 22:56	25/03/2024 10:37	-	Closed	1	

## Example

Let's say you have to build a formula to determine the price of the risks for an home insurance. Depending on the construction type and the array of risks, a formula will be written to return the price. The array for the risks are actually coefficients that are whole number e.g. 1, 2, 3, 4, 5.

Firstly, create the input data by creating an argument. In this case we need as arguments the construction type, the structure type and the risks.

Add Formula Argument

Formula Argument

Name

Risks

Display Name

Risks

Formula Input

HouseholdParameters

Master Type

Collection

Formula Argument Sub-Type

Whole Number

Secondly, create the two steps of the formula, where the first step becomes the input argument of the second step. In the first step, we will calculate the array of prices for the risks and in the second step calculate the sum of those prices.

Add Formula Step

Formula Step

Name

retrieve\_risks

Formula Id

risks

↓

Exclude From Mapping

—

Output type

Master Type

Collection

▼

SubType

Decimal

▼

Execution type

CalculationType

Iteration

▼

Number of Iterations

Risk

↓

Formula

1

result[i] = DataSet("004\_RiskPrice", ("struct\_type", ConstrutionType),("risk\_type", Risk));

Add Formula Step

Formula Step

Name

risk\_price

Formula Id

HouseholdPolicyAmount

Exclude From Mapping

Output type

Master Type

Simple Type

SubType

Whole Number

Execution type

CalculationType

Normal

Formula

1 result = SUM(retrieve\_risks);

Thirdly, test the formula by adding a data set and in the formula test grid inserting a structure type, a construction type and an array of risk (the coefficient of those risks that mark a real risk).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1			1	2	3	4	5	6	7	8	9	10	11	12	13
2			Incendiu	tranzit	explicite	caderea ap de dor	furtuna, uragan, vijelie, tornada	grindina	gretate strat zapade	ploaie torentiale	avalansa	inundatii si alunisuri	alunecare teren	furt	vandalism
3	STRUCT_CONCRETE	apartament in bloc/vila sau casa - structura de rezistenta din beton armat	0.0300	0.0040	0.0040	0.0010	0.0050	0.0015	0.0010	0.0013	0.0002	0.0050	0.0100	0.0100	0.002
4	STRUCT_METAL	apartament in bloc/vila sau casa - structura metalica, cizarie (materiale necombustibile)	0.0275	0.0044	0.0044	0.0012	0.0058	0.0017	0.0012	0.0015	0.0002	0.0058	0.0115	0.0115	0.002
5	STRUCT_WOOD	oricare de mai sus (casa sau ap) - structura din lemn	0.0770	0.0140	0.0140	0.0035	0.0065	0.0020	0.0013	0.0017	0.0003	0.0063	0.0125	0.0125	0.002
6															
7															
8															
9															
10															
11															

# Formula Editor

The formula editor is where you write the formula's actual mathematical expressions and computations. You can find the formula editor at the bottom of the “Add/Edit formula step” form.

For example, based on the data a client inserts as their KYC, the formula editor applies a calculation to return data such as the net income of a loan applicant or the age limit for a contract or the insurance premium for a policy. You can also test your formula before activating it. For more information about running tests, see "[Define Formula Expressions](#)" on page 455.

**Add Formula Step**

**Formula Step**

Name

Formula Id

Exclude From Mapping ☐

**Output type**

Master Type

SubType

**Execution type**

CalculationType

**Formula**

```
1 result = formula;
2
```

## Syntax

In the editor, insert your formula expression based on a mathematical syntax comprised of the formula body and a call for calculation (the returned result). IntelliSense is available for quick selection of definitions and to provide additional information (such as properties and attributes) about the selected item.

**IMPORTANT!**

When writing a formula, make sure the calculations match the step you are on, because the result may be used as input in the subsequent steps.

Use the following syntax in the formula editor to define a formula output:

```
result = <formula expression>;
```

You can include multi-line C# code in the formula expression, as long as you assign a result value:

```
var a = 1;
var b = 2;
result = a + b;
```

The usual operator precedence for operations such as addition "+", subtraction "-", multiplication "\*" and division "/" is enforced and you can further control this by using parentheses () or by splitting your computation in multiple steps.

Formulas can be of the following types:

- constant:  $f = \text{sum}(1, n)$
- linear:  $f(x) = 2 * x + 1$ , where  $x = 1, 100$
- 2-dimensional:  $f(x, y) = x * y + 30$ ;
- n-dimensional etc.
- recursive:  $f(x) = f(x-1) + 20$ .

## Formula Arguments

Primary formula arguments are defined in the formula input (see ["Formula Inputs" on page 451](#)). For example, if you defined an argument called *days* in the formula input, you can create a formula step called *years* to convert the number of days into years with the following expression:

```
result = days / 365;
```

You can also use previous steps' outputs as arguments for subsequent step inputs. For example, after the above step, you can create a step that calculates an interest by multiplying the *principal* and *rate* input parameters with the *years* result from the previous step:

```
result = principal * rate * years
```

## Dynamic Collections Definitions

For collection outputs that use an **Iteration** calculation type, you must provide a generic formula that defines the value of a collection item in relation to the iteration counter specified in the Number of Iterations field (for details, see ["Add steps to a formula" on page 457](#)).

```
result[i] = <formula expression>;
```

For example, if based on a collection of item prices called **Values** (either an input argument or an output from a prior step), you wish to generate a collection of objects that pair together item numbers with their corresponding prices, you can use the following step formula:

```
result[i] = new {Name = "Item" + i, Price = Values[i]};
```

### NOTE

To make sure that the number of items matches the number of prices, set the number of iterations to **Values**.

Recursive formula definitions are also supported. For example, to generate a collection with the first x numbers in the Fibonacci sequence, use a whole number input argument called x for the number of iterations (make sure x is greater than or equal to 2) and use the following formula expression:

```
result[0] = 1;
result[1] = 1;
result[i] = result[i-1] + result[i-2];
```

## IntelliSense Support

Pressing Ctrl+Space will launch the **IntelliSense** that can help you learn more about the formula you are editing, keep track of parameters you're typing, add calls to functions and various information with only a few keystrokes.

### HINT

Already pre-defined Steps also appear in IntelliSense since they can be called in any subsequent step.

## Built-in Functions

You can include the following built-in functions in your formula expressions:

**For Simple types**

### IIF

Returns one of two values, depending on whether the Boolean Condition evaluates to true or false. Output can either be a number or a string. String based output cannot currently be used together with collection based functions like SELECT.

**Syntax**

```
/**
 * @param booleanCondition - condition that has to return
 true/false
 * @param trueValue - the value that is returned if
 booleanCondition is evaluated as true
 * @param falseValue - the value that is returned if
 booleanCondition is evaluated as false
 * @returns - return trueValue or falseValue
 */
IIF(booleanCondition, trueValue, falseValue): boolean
```

**Example**

```
//Example Formula Test
/*
```

```

Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 250
*/

//Exemple Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
400};
*/

//Example Result
//result = IIF(COUNT(simpleCollection) > 0, "trueResult",
"falseResult");

result = IIF(COUNT(simpleCollection) > 0, SUM
(simpleCollection)/COUNT(simpleCollection) , 0);

```

## AVERAGE

Returns the average of the numbers received as parameters.

### Syntax

```

/**
 * @param {number} number1 - required, the first number for
which you want the average
 * @param {number} number2, ... - optional, additional
numbers for which you want the average
 * @returns {number} - the average of the numbers
 */
AVERAGE(number1, [number2], ...): number

```

### Example

```
//Example Formula Test
/*
Input:

Output:
    result = 250
*/

//Example Formula Step
result = AVERAGE(100, 200, 300, 400);
```

## ABS

Returns the absolute value of a number.

### Syntax

```
/**
 * @param {number} number - the number of which you want the
 * absolute value
 * @returns {number} - the absolute value
 */
ABS(number): number
```

### Example

```
//Example Formula Test
/*
Input:
    num = -100
Output:
    result = 100
*/

//Example Formula Step
result = ABS(num);
```

## POWER

Raises a number to a power.

### Syntax

```
/**
 * @param {number} number - the base number
 * @param {number} exponent - the exponent to which the base
 * number is raised
 * @returns {number} -
 */
POWER(number, exponent): number
```

### Example

```
//Example Formula Test
/*
Input:
    num = 10
    exp = 2
Output:
    result = 100
*/

//Example Formula Step
result = POWER(num, exp);
```

## ODD

Returns true if the integer number is odd, otherwise false.

### Syntax

```
/**
 * @param {number} number - the number the needs to be
 * verified
 * @returns {boolean} -
 */
ODD(number): boolean
```

### Example

```
//Example Formula Test
/*
Input:
    num = 3
```

```

Output:
    result = true
*/

/*
Input:
    num = 2
Output:
    result = false
*/

//Example Formula Step
result = ODD(num);

```

## EVEN

Returns true if the integer number is even, otherwise false.

### Syntax

```

/**
 * @param {number} number - the number the needs to be
 verified
 * @returns {boolean} -
 */
EVEN(number): boolean

```

### Example

```

//Example Formula Test
/*
Input:
    num = 3
Output:
    result = false
*/

/*
Input:
    num = 2
Output:
    result = true

```

```
*/

//Example Formula Step
result = EVEN(num);
```

## TRUNC(Number)

Calculates the integral part of a specified decimal number.

### Syntax

```
/**
 * @param {number} number - the number the needs to be
truncated
 * @returns {number} - the integral part of the number
 */
TRUNC(number): number
```

### Example

```
//Example Formula Test
/*
Input:
    num = 17.53M
Output:
    result = 17
*/

//Example Formula Step
result = TRUNC(num);
```

## ROUND(Number, [Precision])

Rounds a decimal value to a specified number of fractional digits.

### Syntax

```
/**
 * @param {number} num - the number the needs to be rounded
 * @param {number} precision - optional, number of decimal
places in the return value. The default value is 0
```

```
* @returns {number} - The number nearest to num that
contains a number of fractional digits equal to precision.
*/
ROUND(num, [precision]): number
```

### Example

```
//Example Formula Test
/*
Input:
    num = -17.51M
Output:
    result = -18
*/

/*
Input:
    num = 17.51M
    precision = 1
Output:
    result = 17.5
*/

//Example Formula Step
result = ROUND(num, precision);
```

## ROUNDUP

Returns the smallest integral value that is greater than or equal to the specified decimal number.

### Syntax

```
/**
* @param {number} num - the number the needs to be rounded
up
* @returns {number} - the smallest integral value that is
greater than or equal to the specified decimal number.
*/
ROUNDUP(num): number
```

**Example**

```
//Example Formula Test
/*
Input:
    num = -17.51M
Output:
    result = -17
*/

/*
Input:
    num = 17.51M
Output:
    result = 18
*/

//Example Formula Step
result = ROUNDUP(num);
```

## ROUNDDOWN

Returns the largest integer less than or equal to the specified decimal number.

**Syntax**

```
/**
 * @param {number} num - the number the needs to be rounded
down
 * @returns {number} - the largest integer less than or
equal to the specified decimal number.
 */
ROUNDDOWN(num): number
```

**Example**

```
//Example Formula Test
/*
Input:
    num = -17.51M
```

```

Output:
    result = -18
*/

/*
Input:
    num = 17.51M
Output:
    result = 17
*/

//Example Formula Step
result = ROUNDDOWN(num);

```

## FLOOR

Returns the largest integer less than or equal to the specified decimal number.

### Syntax

```

/**
 * @param {number} num - the number
 * @returns {number} - the largest integer less than or
 * equal to the specified decimal number.
 */
FLOOR(num): number

```

### Example

```

//Example Formula Test
/*
Input:
    num = -17.51M
Output:
    result = -18
*/

/*
Input:
    num = 17.51M
Output:

```

```

    result = 17
*/

//Example Formula Step
result = FLOOR(num);

```

### For Collection Types

## SELECT

Applies a function to each element of the collection and returns a new collection with the results of the function invocation.

```

//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 }]

Output:
    result = [100, 200]
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT = 100};
    var prod2 = new {Name = "Product2" , priceWithVAT = 200};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2);
*/
result = FROM(productCollection).SELECT(x=>x.priceWithVAT);

//Example Formula Test
/*
Input:

```

```

        simpleCollection = [100, 200, 300, 400]

Output:
    result = [200, 400, 600, 800]
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
400};
*/
result = FROM(simpleCollection).SELECT(x=>x * 2);

```

## WHERE

Applies a function to each element of the collection and returns a new collection with the filtered elements that respect the condition.

```

//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 }]

Output:
    result = [200]
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT =
100};

```

```

    var prod2 = new {Name = "Product2" , priceWithVAT = 200};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2);
*/
result = FROM(productCollection).WHERE
(x=>x.priceWithVAT>150).SELECT(x=>x.priceWithVAT);

```

## FIRSTORDEFAULT

Returns first element of the collection or the default value (0 for numeric elements). It goes well when used with WHERE and you are sure only one record is returned.

```

//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 }]

Output:
    result = 200
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT = 100};
    var prod2 = new {name = "Product2" , priceWithVAT = 200};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2);
*/
result = FROM(productCollection).WHERE
(x=>x.priceWithVAT>150).SELECT
(x=>x.priceWithVAT).FIRSTORDEFAULT();

```

## GROUPBY

Can only be used with aggregate function SUM.

Returns a new collection grouped by a property of the object.

```
//Example Formula Test
/*
Input:
    productCollection = [
        {category: "Cat1", name:"Product1", priceWithVAT:
100, quantity: 1 },
        {category: "Cat2", name:"Product2", priceWithVAT:
200, quantity: 2 },
        {category: "Cat1", name:"Product3", priceWithVAT:
300, quantity: 3 }]

Output:
    result = [
        {
            "category": "Cat1",
            "priceWithVAT": 400.0
        },
        {
            "category": "Cat2",
            "priceWithVAT": 200.0
        }
    ]
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {category: "Cat1", name = "Product1" ,
priceWithVAT = 100, quantity: 1};
    var prod2 = new {category: "Cat2", name = "Product2" ,
priceWithVAT = 200, quantity: 2};
    var prod3 = new {category: "Cat1", name = "Product3" ,
priceWithVAT = 300, quantity: 3};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2,
prod3);
*/
result = FROM(productCollection).GROUPBY("category").SUM
("priceWithVAT");
```

## EXTENDELEMENTS

Returns a new collection with a new property added to all elements in the collection. Also, for each element it assigns a value for the new added property.

```
//Example Formula Test
/*
Input:
    productCollection = [
        {category: "Cat1", name:"Product1", priceWithVAT:
100, quantity: 1 },
        {category: "Cat2", name:"Product2", priceWithVAT:
200, quantity: 2 },
        {category: "Cat1", name:"Product3", priceWithVAT:
300, quantity: 3 }]

Output:
    result = [
        {
            "category": "Cat1",
            "name": "Product1",
            "priceWithVAT": 100.0,
            "quantity": 1.0,
            "totalPriceWithVAT": 100.0
        },
        {
            "category": "Cat2",
            "name": "Product2",
            "priceWithVAT": 200.0,
            "quantity": 2.0,
            "totalPriceWithVAT": 400.0
        },
        {
            "category": "Cat1",
            "name": "Product3",
            "priceWithVAT": 300.0,
            "quantity": 3.0,
            "totalPriceWithVAT": 900.0
        }
    ]
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
```

```

3. Created in Formula Step like :
    var prod1 = new {category: "Cat1", name = "Product1" ,
    priceWithVAT = 100, quantity: 1};
    var prod2 = new {category: "Cat2", name = "Product2" ,
    priceWithVAT = 200, quantity: 2};
    var prod3 = new {category: "Cat1", name = "Product3" ,
    priceWithVAT = 300, quantity: 3};
    var productCollection = new
    EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2,
    prod3);
    */
    result = FROM(productCollection).EXTENDELEMENTS
    ("totalPriceWithVAT", x=>x.priceWithVAT * x.quantity);

```

To add multiple properties to a collection, use the EXTENDELEMENTS function repeatedly:

```

var newColl = FROM(CollArg).EXTENDELEMENTS("elem1",
x=>x.Price*2);
result = FROM(newColl).EXTENDELEMENTS("elem2",
x=>x.Price*10);

```

## COUNT

Count is a property of collections, it returns the number of elements from the collection.

```

//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 }]

Output:
    result = 2
    */

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object

```

```

2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT =
100};
    var prod2 = new {name = "Product2" , priceWithVAT =
200};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2);
*/
result = FROM(productCollection).WHERE
(x=>x.priceWithVAT>=100).Count;

```

## SUM

Returns the sum of a property from the array of objects.

```

//Example Formula Test
/*
Input:
    productCollection = [
        {category: "Cat1", name:"Product1", priceWithVAT:
100, quantity: 1 },
        {category: "Cat2", name:"Product2", priceWithVAT:
200, quantity: 2 },
        {category: "Cat1", name:"Product3", priceWithVAT:
300, quantity: 3 }]

Output:
    result = 600
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {category: "Cat1", name = "Product1" ,
priceWithVAT = 100, quantity: 1};
    var prod2 = new {category: "Cat2", name = "Product2" ,
priceWithVAT = 200, quantity: 2};
    var prod3 = new {category: "Cat1", name = "Product3" ,
priceWithVAT = 300, quantity: 3};
    var productCollection = new
EBS.Core.Formulas.ObjectVector<dynamic>(prod1, prod2,
prod3);

```

```
*/
result = FROM(productCollection).SUM("priceWithVAT");
```

## MIN & MAX

Return the items from the collection with the min/max value for a property of the object.

Examples:

```
//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 },
        {"name":"Product3", "priceWithVAT": 100 }]

Output:
    result = [{"name":"Product2", "priceWithVAT": 200}]
*/

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT =
100};
    var prod2 = new {name = "Product2" , priceWithVAT =
200};

*/
result = FROM(productCollection).MAX(x=>x.priceWithVAT);
```

```
//Example Formula Test
/*
Input:
    productCollection = [
        {"name":"Product1", "priceWithVAT": 100 },
        {"name":"Product2", "priceWithVAT": 200 },
        {"name":"Product3", "priceWithVAT": 100 }]

Output:
```

```

        result = [{"name":"Product1", "priceWithVAT": 100 },
                  {"name":"Product3", "priceWithVAT": 100 }]
    */

//Example Formula Step
/* productCollection can be:
1. An argument of type Collection of Object
2. The result of a Previous Step
3. Created in Formula Step like :
    var prod1 = new {name = "Product1" , priceWithVAT =
100};
    var prod2 = new {name = "Product2" , priceWithVAT =
200};

*/
result = FROM(productCollection).MIN(x=>x.priceWithVAT);

```

#### For Simple Collections Types

## RANGE with SKIP and/or TAKE

Generates a sequence of numbers within a specified range.

#### Syntax

```

/**
 * @param {number[]} simpleCollection - simple collection of
decimals
 * @param rangeOperators - range operators of type SKIP and
TAKE
 * @returns {number[]} - the sequence of numbers within a
specified range.
 */
RANGE(simpleCollection, rangeOperators...): number[]

```

#### Example

```

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

```

```

Output:
    result = [100, 200]
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
400};
*/
result = RANGE(simpleCollection, TAKE(2));

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = [200, 300]
*/

//Example Formula Step
result = RANGE(simpleCollection, SKIP(1), TAKE(2));

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = [200, 300, 400]
*/

//Example Formula Step
result = RANGE(simpleCollection, SKIP(1));

```

## MIN / MAX

Returns the min/max from the collection.

## Syntax

```
/**
 * @param simpleCollection - simple collection of decimals
 * @returns {number} -
 */
MIN(simpleCollection): number
MAX(simpleCollection): number
```

## Example

```
//Example Formula Test
/*
Input:
    simpleCollection = [200, 300, 400]

Output:
    result = 100
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
    EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
    400};
*/
result = MIN(simpleCollection);

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 400
*/

//Example Formula Step
result = MAX(simpleCollection);
```

## AVERAGE

Returns the average of the collection of numbers received as parameter.

### Syntax

```
/**
 * @param {number[]} simpleCollection - simple collection of
 * decimals
 *
 * @returns {number} - the average of the numbers
 */
AVERAGE(simpleCollection): number
```

### Example

```
//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 250
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
    EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
    400};
*/
result = AVERAGE(simpleCollection);
```

## SUM / COUNT

Returns the sum/count of the elements from the collection.

### Syntax

```

/**
 * @param {number[]} simpleCollection - simple collection of
 decimals
 * @returns {number} -
 */
SUM(simpleCollection): number
COUNT(simpleCollection): number

```

### Example

```

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 1000
*/

//Example Formula Step
result = SUM(simpleCollection);

//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 4
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
400};
*/
result = COUNT(simpleCollection);

```

## SUMIF / COUNTIF

Returns the sum/count of the elements from the collection that respect the condition.

### Syntax

```
/**
 * @param {number[]} simpleCollection - simple collection of
 * decimals
 * @param filter - the condition for filtering
 * @returns {number} -
 */
SUMIF(simpleCollection, filter): number
COUNTIF(simpleCollection, filter): number
```

### Example

The 'it' from the filter is just a convention for naming an item of the collection.

```
//Example Formula Test
/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 700
*/

//Example Formula Step
/* simpleCollection can be:
1. An argument of type Collection of Whole Number or Decimal
2. The result of a Previous Step
3. Created in Formula Step like :
    var simpleCollection = new
EBS.Core.Formulas.DecimalVector(100, 200, 300, 400);
    OR
    var simpleCollection = new decimal[] {100, 200, 300,
400};
*/
result = SUMIF(simpleCollection, it > 250);

//Example Formula Test
```

```

/*
Input:
    simpleCollection = [100, 200, 300, 400]

Output:
    result = 2
*/

//Example Formula Step
result = COUNTIF(simpleCollection, it > 250);

```

## Formula Built-in Excel Functions

# RATE

Returns the interest rate per period of an annuity. RATE is calculated by iteration and can have zero or more solutions.

## Syntax

```

/**
 * @param {number} nPer - Required. The total number of
 * payment periods in an annuity.
 * @param {number} pmt - Required. The payment made each
 * period and cannot change over the life of the annuity.
 * @param {number} pv - Required. The present value – the
 * total amount that a series of future payments is worth now.
 * @param {number} fv - Optional. The future value, or a
 * cash balance you want to attain after the last payment is
 * made. Default value 0.
 * @param {number} dueDate - Optional. 0 or omitted - At the
 * end of the period / 1 - At the beginning of the period
 * @param {number} guess - Optional. Your guess for what the
 * rate will be. Default is 10%
 * @returns {number} - Returns the interest rate per period
 * of an annuity
 */
RATE(nPer, pmt, pv, fv, dueDate, guess): number

```

## Example

```
//Example Formula Test
/*
Input:
    years: 4,
    monthlyPayment: -200,
    loanAmount: 8000

Output:
    result = 0.00770147248821008M

*/

//Example Formula Step
result = RATE(years * 12, monthlyPayment, loanAmount);
```

## PMT

Calculates the payment for a loan based on constant payments and a constant interest rate.

### Syntax

```
/**
 * @param {number} rate - Required. The interest rate for the
 * loan.
 * @param {number} nPer - Required. The total number of
 * payments for the loan.
 * @param {number} pv - Required. The present value – the
 * total amount that a series of future payments is worth now.
 * @param {number} fv - Optional. The future value, or a
 * cash balance you want to attain after the last payment is
 * made. Default value 0.
 * @param {number} dueDate - Optional. 0 or omitted - At the
 * end of the period / 1 - At the beginning of the period
 * @returns {number} - Returns the payment for a loan based
 * on constant payments and a constant interest rate.
 */
PMT(rate, nPer, pv, fv, dueDate)): number
```

### Example

```
//Example Formula Test
```

```

/*
Input:
    anualRate: 0.08,
    monthsPayments: 10,
    loanAmount: 10000

Output:
    result = -1037.03208935916

*/

//Example Formula Step
result = PMT(anualRate/12, monthsPayments, loanAmount);

```

## NPER

Returns the number of periods for an investment based on periodic, constant payments and a constant interest rate.

### Syntax

```

/**
 * @param {number} rate - Required. The interest rate for the
 loan.
 * @param {number} pmt - Required. The payment made each
 period and cannot change over the life of the annuity.
 * @param {number} pv - Required. The present value – the
 total amount that a series of future payments is worth now.
 * @param {number} fv - Optional. The future value, or a
 cash balance you want to attain after the last payment is
 made. Default value 0.
 * @param {number} dueDate - Optional. 0 or omitted - At the
 end of the period / 1 - At the beginning of the period
 * @returns {number} - Returns the number of periods for an
 investment based on periodic, constant payments and a
 constant interest rate.
 */
NPER(rate, pmt, pv, fv, dueDate): number

```

### Example

```

//Example Formula Test
/*

```

```

Input:
    annulInterestRate: 0.12,
    monthlyPayment: -100,
    presentValue: -1000,
    futureValue: 10000

Output:
    result = 59.6738657

*/

//Example Formula Step
result = NPER(annulInterestRate/12, monthlyPayment,
presentValue, futureValue, 1);

```

## NPV

Calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).

### Syntax

```

/**
 * @param {number} rate - Required. The interest rate per
 * period.
 * @param {number} values - Required. The cash flow values.
 * It must contain at least one negative value (a payment) and
 * one positive value (a receipt).
 * @returns {number} - Calculates the net present value of an
 * investment by using a discount rate and a series of future
 * payments (negative values) and income (positive values).
 */
NPV(rate, values): number

```

### Example

```

//Example Formula Test
/*
Input:
    annulDiscountRate: 0.1,
    cashFlowValues: [-10000, 3000, 4200, 6800]

```

```

Output:
    result = 1,188.44

*/

//Example Formula Step
result = NPV(annulDiscountRate, cashFlowValues);

```

### Formula Built-in DateTime Functions

## GETDATEPART

Returns the datePart of the date. The datePart refers to DAY, MONTH Or YEAR. Use DATEPART.DAY/MONTH/YEAR to specify the desired format.

### Syntax

```

/**
 * @param dateTimeArgument {Date} - the date argument
 * @param datePart - refers to DAY, MONTH Or YEAR
 * @returns {number} - the datePart of the date
 */
GETDATEPART(dateTimeArgument, datePart): number

```

### Example

```

//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-01T00:00:00.000Z";
    datePart = DATEPART.DAY;

Output:
    result = 1;
*/

//Example Formula Step
result = GETDATEPART(dateTimeArgument, datePart);

```

## GETDATE

Returns the date without the time component.

#### Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @returns {Date} - the date without the time component
 */
GETDATE(dateTimeArgument): Date
```

#### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-01T10:05:00.000Z";
Output:
    result = "2021-05-01T00:00:00Z";
*/

//Example Formula Step
result = GETDATE(dateTimeArgument);
```

## GETDAY

Returns the day of the month.

#### Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @returns {number} - the day of the month
 */
GETDAY(dateTimeArgument): number
```

#### Example

```
//Example Formula Test
/*
Input:
```

```

        dateTimeArgument = "2021-05-15T00:00:00.000Z";
Output:
        result = 15;
*/

//Example Formula Step
result = GETDAY(dateTimeArgument);

```

## GETMONTH

Returns the month component of the date.

Syntax

```

/**
 * @param dateTimeArgument {Date} - the date argument
 * @returns {number} - the month component of the date
 */
GETMONTH(dateTimeArgument): number

```

Example

```

//Example Formula Test
/*
Input:
        dateTimeArgument = "2021-05-15T00:00:00.000Z";
Output:
        result = 5;
*/

//Example Formula Step
result = GETMONTH(dateTimeArgument);

```

## GETYEAR

Returns the year component of the date.

Syntax

```

/**

```

```

* @param dateTimeArgument {Date} - the date argument
* @returns {number} - the day of the month
*/
GETYEAR(dateTimeArgument): number

```

### Example

```

//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-15T00:00:00.000Z";
Output:
    result = 2021;
*/

//Example Formula Step
result = GETYEAR(dateTimeArgument);

```

## ADDDATEPART

Returns a new date that adds the specified number of datePart. The datePart refers to DAY, MONTH or YEAR. Use DATEPART.DAY/MONTH/YEAR to specify the desired format.

### Syntax

```

/**
* @param dateTimeArgument {Date} - the date argument
* @param value {number} - the value to be added to the given date
* @param datePart - refers to DAY, MONTH Or YEAR
* @returns {Date} - a new date that adds the specified number of datePart
*/
ADDDATEPART(dateTimeArgument, value, datePart): Date

```

### Example

```

//Example Formula Test
/*

```

```

Input:
    dateTimeArgument = "2021-05-15T00:00:00.000Z";
    value = 5;
    datePart = DATEPART.MONTH;
Output:
    result = "2021-10-15T00:00:00.000Z";
*/

//Example Formula Step
result = ADDDATEPART(dateTimeArgument, 5, DATEPART.MONTH);

```

## ADDDAYS

Returns a new date that adds the specified number of days.

### Syntax

```

/**
 * @param dateTimeArgument {Date} - the date argument
 * @param value {number} - the value to be added to the given
 * date
 * @returns {Date} - a new date that adds the specified
 * number of days
 */
ADDDAYS(dateTimeArgument, value): Date

```

### Example

```

//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-15T00:00:00.000Z";
    value = 15;
Output:
    result = "2021-05-30T00:00:00.000Z";
*/

//Example Formula Step
result = ADDDAYS(dateTimeArgument, value);

```

## ADDMONTHS

Returns a new date that adds the specified number of months.

### Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @param value {number} - the value to be added to the given
 * date
 * @returns {Date} - a new date that adds the specified
 * number of months
 */
ADDMONTHS(dateTimeArgument, value): Date
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-15T00:00:00.000Z";
    value = 2;
Output:
    result = "2021-07-15T00:00:00.000Z";
*/

//Example Formula Step
result = ADDMONTHS(dateTimeArgument, value);
```

## ADDYEARS

Returns a new date that adds the specified number of years.

### Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @param value {number} - the value to be added to the given
 * date
 * @returns {Date} - a new date that adds the specified
 * number of years
 */
ADDYEARS(dateTimeArgument, value): Date
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument = "2021-05-15T00:00:00.000Z";
    value = 5;
Output:
    result = "2026-05-15T00:00:00.000Z";
*/

//Example Formula Step
result = ADDYEARS(dateTimeArgument, value);
```

## DATECOMPARE

Returns '-1' if date1 is earlier than date2, '0' if date1 is the same as date2, and it returns '1' if date1 is later than date2.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - '-1' if dateTimeArgument1 is earlier
   than dateTimeArgument2, '0' if dateTimeArgument1 is the same
   as dateTimeArgument2 and it returns '1' if dateTimeArgument1
   is later than dateTimeArgument2
 */
DATECOMPARE(dateTimeArgument1, dateTimeArgument2): number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2021-05-15T00:00:00.000Z";
    dateTimeArgument2 = "2021-06-15T00:00:00.000Z";
Output:
    result = -1;
*/
```

```
//Example Formula Step
result = DATECOMPARE(dateTimeArgument1, dateTimeArgument2);
```

## DATEDIFF

Returns the difference between two dates. The difference is expressed in the specified datePart. The datePart refers to DAY, MONTH or YEAR. Use DATEPART.DAY/MONTH/YEAR to specify the desired format.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @param datePart - refers to DAY, MONTH Or YEAR
 * @returns {number} - the difference between
 * dateTimeArgument1 and dateTimeArgument2 expressed in the
 * specified datePart
 */
DATEDIFF(dateTimeArgument1, dateTimeArgument2, datePart):
number
```

### Example

```
//Example Formula Test
/*
Input:
  dateTimeArgument1 = "2021-05-25T00:00:00.000Z";
  dateTimeArgument2 = "2021-05-05T00:00:00.000Z";
  datePart = DATEPART.DAY;
Output:
  result = 20;
*/

//Example Formula Step
result = DATEDIFF(dateTimeArgument1, dateTimeArgument2,
datePart);
```

## DIFFDAYS

Returns the difference of days between two dates.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - the difference of days between two
 * dates
 */
DIFFDAYS(dateTimeArgument1, dateTimeArgument2): number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2021-05-01T00:00:00.000Z";
    dateTimeArgument2 = "2021-05-11T00:00:00.000Z";
Output:
    result = -10;
*/

//Example Formula Step
result = DIFFDAYS(dateTimeArgument1, dateTimeArgument2);
```

## DIFFMONTHS

Returns the difference of months between two dates.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - the difference of months between two
 * dates
 */
DIFFMONTHS(dateTimeArgument1, dateTimeArgument2): number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2021-08-01T00:00:00.000Z";
    dateTimeArgument2 = "2021-07-11T00:00:00.000Z";
Output:
    result = 1;
*/

//Example Formula Step
result = DIFFMONTHS(dateTimeArgument1, dateTimeArgument2);
```

## DIFFMONTHSEXACT

Returns the difference of months between two dates.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - the difference of months between two
 * dates
 */
DIFFMONTHSEXACT(dateTimeArgument1, dateTimeArgument2):
number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2021-08-01T00:00:00.000Z";
    dateTimeArgument2 = "2021-07-11T00:00:00.000Z";
Output:
    result = 0;
*/

//Example Formula Step
result = DIFFMONTHSEXACT(dateTimeArgument1,
dateTimeArgument2);
```

## DIFFYEARS

Returns the difference of years between two dates.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - the difference of years between two
 * dates
 */
DIFFYEARS(dateTimeArgument1, dateTimeArgument2): number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2022-08-01T00:00:00.000Z";
    dateTimeArgument2 = "2021-11-11T00:00:00.000Z";
Output:
    result = 1;
*/

//Example Formula Step
result = DIFFYEARS(dateTimeArgument1, dateTimeArgument2);
```

## DIFFYEARSEXACT

Returns the difference of years between two dates.

### Syntax

```
/**
 * @param dateTimeArgument1 {Date} - the first date argument
 * @param dateTimeArgument2 {Date} - the second date argument
 * @returns {number} - the difference of years between two
 * dates
 */
DIFFYEARSEXACT(dateTimeArgument1, dateTimeArgument2): number
```

### Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument1 = "2022-08-01T00:00:00.000Z";
    dateTimeArgument2 = "2021-11-11T00:00:00.000Z";
Output:
    result = 0;
*/

//Example Formula Step
result = DIFFYEARSEXACT(dateTimeArgument1,
dateTimeArgument2);
```

## FIRSTDAYOFMONTH

Return the first day of month.

Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @returns {Date} - the first day of month
 */
FIRSTDAYOFMONTH(dateTimeArgument): Date
```

Example

```
//Example Formula Test
/*
Input:
    dateTimeArgument = "2022-08-21T00:00:00.000Z";
Output:
    result = "2022-08-01T00:00:00";
*/

//Example Formula Step
result = FIRSTDAYOFMONTH(dateTimeArgument);
```

## LASTDAYOFMONTH

Return the last day of month.

## Syntax

```
/**
 * @param dateTimeArgument {Date} - the date argument
 * @returns {Date} - the last day of month
 */
LASTDAYOFMONNTH(dateTimeArgument): Date
```

## Example

```
//Example Formula Test
/*
Input:
  dateTimeArgument = "2022-08-21T00:00:00.000Z";
Output:
  result = "2022-08-31T00:00:00";
*/

//Example Formula Step
result = LASTDAYOFMONNTH(dateTimeArgument);
```

## Data Set Calls

To extract value mappings from data sets (see ["Data Sets" on the next page](#) for details), use the following syntax:

```
DataSet("<data set name>", ("<discriminant 1 name>", <discriminant 1 value>), ("<discriminant 2 name>", <discriminant 2 value>) ... )
```

If all the discriminants are given as a parameter, the result will be a value. However, if one of the discriminants is not sent as a parameter, the result will be an array.

### IMPORTANT!

You can only call data sets that are active. For details, see ["Data Set Versioning" on page 518](#).

## Example

In the example below, we return a risk coefficient from a data set called *RiskPrice* with two discriminants (*structure\_type* and *risk\_type*). We retrieve the value corresponding to the structure and risk stored in the *myStructure* and *myRisk* input parameters.

### IMPORTANT!

The sub-types that are text are written using quotation marks.

```
result = DataSet("RiskPrice", ("structure_type",
myStructure), ("risk_type", myRisk));
```

## Data Sets

Data sets are mappings that associate values for a set of discriminants (such as age, sex, driving experience, etc.) to values for a specific metric (such as a risk coefficient). You can call data sets from your functions, allowing you to reference the predefined metric value that matches a set of arguments (for details, see ["Data Set Calls" on the previous page](#)).

Data set values are populated by importing mapping values from Excel files and/or by manual entries.

For example, you can import a file with coefficients for the customer's age.

	A	B
1	Age	Value
2	[;17]	0
3	[18;25]	50
4	[26;35]	100
5	[36;50]	150
6	[51;]	75

## Create a Data Set

Data set discriminants is the grid where a user can teach the system the two dimensional data in rows and columns.

The value of an argument is dependent to a discriminant contained in the Excel file imported. For example, a column or row could be the customer's age, their income, status, the item insured, type of car, risk zones etc. For each of these text items you can set coefficients in Excel.

**IMPORTANT!**


If the Excel file has more than one sheet, those other sheets will not be imported. Please, import one sheet at a time.

1. Open the Main Menu in FintechOS Studio and elect **Automation Blocks > Business Formulas > Business Data Sets**.
2. Click **Insert**.
3. Enter a **Name** for the data set. This is a unique name used to identify the data set in the system.
4. Optionally enter a **Description** for the data set.
5. In the **Value Types** field, select if the data set returns **Numeric** or **Text** values.
6. Check the **Single Value** box if the data set includes a mapping between a single discriminant key and a single metric value.
7. Select the **Start Date** at which the data set becomes active. Select the time as well. This makes it possible to activate and later on deactivate the formula for minute to minute.
8. The **End Date** and **Version** fields are populated automatically based on the data set's versioning (see "[Data Set Versioning](#)" on page 518 for details).
9. The **Has Column Description** and **Has Row Description** checkboxes indicate that the Excel files used to import value mappings include field descriptions in the second row and/or second column (key values are set in the first row and first column).
10. Click **Save and Reload**.

## Define Data Set Discriminants (non Single Value data sets)



1. In the data set screen, in the **Data Set Discriminants** section, click the **Insert** button to add a discriminant.
2. Enter a **Name** for the discriminant.
3. The **Data Set** field is automatically populated with your data set name.
4. Optionally enter a **Description** for the discriminant.
5. In the **Values Type** field, select the data type for the discriminant. This can be either **Text**, **Numeric**, or **Option Set**. In the case of option sets, an **Option Set** field will be displayed, allowing you to select the option set that the discriminant values must belong to (see ["Add Option Set Attributes" on page 89](#) for details).
6. Select **Is Interval** if the discriminant values are in the form of a value range.
  - Intervals must be entered using the following syntax: [*<maximum value>*], [*<minimum value>*; *<maximum value>*], or [*<minimum value>*; ].
  - If the minimum value is not specified, the interval covers all values smaller than or equal to the maximum value. If the maximum value is not specified, the interval covers all values greater than or equal to the minimum value.
  - You can use closed intervals [...], open intervals (...), or half-open intervals [...)(...).
7. Select **Is Row Key** if discriminant values are represented in the first column of the imported Excel file used to add data set values (see ["Add Data Set Values \(non Single Value data sets\)" on the next page](#)). You can have a only one row key discriminant per data set.
8. Select **Is Column Key** if discriminant values are represented in the first row of the imported Excel file used to add data set values (see ["Add Data Set Values \(non Single Value data sets\)" on the next page](#)).

Value data sets)" on the next page). You can have only one column key discriminant per data set.

9. Click the **Save and Close** button () at the top right corner of the page.
10. Repeat for any additional discriminants you wish to include in your formula.

## Add Data Set Values (single value data sets)

For this kind of data set, no Excel file is needed. The values are inserted in the form and the "**Start Import**" button is missing.

1. In the data set screen, in the **Data Set Values** section, select **Insert**.
2. The **Data Set** field is automatically populated with your data set name.
3. Enter a **Name** for the value mapping.
4. Optionally enter a **Description** for the value mapping.
5. Click the **Save and Reload** () button at the top right corner of the page.
6. At the bottom of the page:
  - i. Enter the discriminant value in the left column.
  - ii. Enter the metric value in the right column.
7. Click the **Save and Close** button () at the top right corner of the page.

## Add Data Set Values (non Single Value data sets)



Data set value mappings are imported via Excel files for discriminants that are column/row keys and entered manually for discriminants that are not.


### HINT

If your data set includes more than two discriminants, set the discriminants with the top two highest cardinalities as row key and column key. This way, you will need fewer sets of values.

For instance, if you have 3 discriminants, such as age (4 age brackets), education (5 education levels), and sex (2 sexes), you can set age and education as row and column key values. This will allow you to populate the data set with only two value

sets: one Excel file with age/education mappings for men and another with age/education mappings for women.

1. In the data set screen, in the **Data Set Values** section, click the **Insert** button to add a set of values.
2. The **Data Set** field is automatically populated with your data set name.
3. Enter a **Name** for the set of values.
4. Optionally enter a **Description** for the set of values.
5. Insert in the **Import file** the Excel file you wish to use.
6. Click the "**Start Import**" button on the right side of the corner.
7. Click the **Save and Reload** () button at the top right corner of the page.
8. Enter keys for any non-row/column key discriminants:
  - i. Click the **Insert** button in the Data Set Discriminant Values section.
  - ii. Enter a **Name** for the discriminant.
  - iii. The **Data Set Value** is populated automatically.
  - iv. Select the corresponding **Data Set Discriminant** from the list.
  - v. Enter the discriminant key value in the **Discriminant Value Text** field.
  - vi. Click the **Save and Close** () button at the top right corner of the page.
  - vii. Repeat for any remaining non-row/column key discriminants.
9. Click the **Add file** button to select the Excel import file for the row/column key discriminants. The Excel file must match the discriminants' data type settings and formatting (description row/description column).
10. Click the **Start Import** button at the top right corner of the page. After the import finishes, the imported data will be displayed at the bottom of the page.

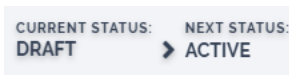
11. Click the **Save and Close** button () at the top right corner of the page.
12. Repeat for any additional value sets you wish to include in your data set.

## Data Set Versioning

When you first create a data set, it will be in a **Draft** state, meaning that it can be edited, but not used by the system.

### Activate a Data Set

To activate a data set draft, in the data set page, click the **Next Status: Active** field in the top left corner of the screen.



Once activated, a data set can be used in formulas (see ["Data Set Calls"](#) on page 512 for details).

### IMPORTANT!

Once activated, you can revert an active data set to the draft status only if:

1. Its start date/time is later than the current date/time.
2. There is no other draft created in a newer data set version (see ["Create a New Data Set Version Draft"](#) below).

### Create a New Data Set Version Draft

A data set that is currently in use cannot be modified. Instead, a **Create New Version** button will appear in the top right corner of your data set page.

Click the **Create New Version** button to create a new draft version based on the active data set. You can edit the draft version while the active data set is still enabled. The new version will replicate the start date of the original data set if that date is in the future. Otherwise, it will have the current date as start date (the date and time when it was created e.g. 15:53), which will have to be changed as it creates a *Date is prior to current time* error.

←

CURRENT STATUS:  
DRAFT

→

NEXT STATUS:  
ACTIVE

Save and reload

Business Transactions

Name

Test

Display Name

Test

Description

Values Type

Numeric

Single Value

☐

Return Default Value

☐

Ignore Empty Values

☐

Start Date

10/01/2023 15:53

End Date

Date is prior to current time

Formula/Business Data Set ... 10:25

Activate a Data Set Version Draft

Once you finish updating the draft version, change its status from Draft to Active as shown above. The previously active version will be set to a Closed state, and the draft version will become the currently active version.

You can track the data set versions in the **History** tab of the data set page.

←

CURRENT STATUS:  
ACTIVE

→

NEXT STATUS:  
DRAFT

Save and reload

Business Transactions

1 Details

2 History

× Delete

Export

Refresh

<input type="checkbox"/>	Name	Start Date	End Date	Single Value	Business Status	Version	Digital Asset
<input type="checkbox"/>	Test	25/03/2024 10:37		<input type="checkbox"/>	Active	4	
<input type="checkbox"/>	Test	24/03/2024 22:56	25/03/2024 10:37	<input type="checkbox"/>	Closed	1	

Clone Data Sets

Cloning a data set creates a new, independent copy of an existing data set. This allows you to use an existing data set as a template for similar use cases. To clone a data set:

1. In FintechOS Studio, go to **Main Menu > Automation Blocks > Business Formulas > Business Data Sets**.
2. Locate the data set to be cloned and double-click it to open.
3. Click the **Clone** button in the top right corner of the screen.
4. The cloned data set is created and opened automatically.
5. Modify the cloned data set as needed, then click **Save and Close**.

## Formula Parameter Mapping

After defining a formula, as by the configurations in "[Define Formula Expressions](#)" on [page 455](#), create the mapping needed for the use of the formula. To do so:

1. Open FintechOS Studio, open the main menu, select **Business Formulas**, select **Formula Parameters Mapping**.
2. Select **Insert** to add a new one, or **Delete** to erase a mapping.
3. To create a new one, fill in the following:

The screenshot shows the 'Formula parameter mapping' form. It has three tabs: 'Definition' (selected), 'Input', and 'Output'. The 'Definition' tab contains the following fields:

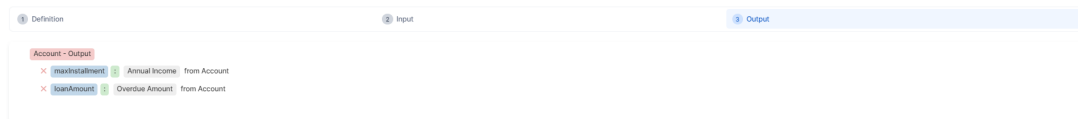
- Data Mapping Type:** A dropdown menu with 'Formula' selected.
- Master Entity:** A dropdown menu with 'Account' selected.
- Operation Name:** A dropdown menu with 'MaxAmountOverMaxTensor' selected.
- Name:** A text field containing 'MaxAmountOverMaxTensor\_formula\_Account'.

- **Data Mapping Type:**
  - formula
  - insurance type.
- **Master Entity:** Choose the corresponding entity from where the user wishes to get the data.

**NOTE**

Be sure to select the same entity as the entity for the formula created before.

- **Operation Name:** Choose the formula from the list.
  - **Name:** Insert a name for the mapping.
4. Click **Save and reload**.
  5. Click **Input** to map the input to the fields from the entity.
  6. Click **Save and reload**.
  7. Select the **Output** tab to map the results from the formula to the fields from the account.



8. Click **Save and Reload**.

## Use Business Formulas in a Digital Journey

After you configured a data set, a formula expression in the editor, and activated the formula, you can use the formula in a journey or entity form.

### Call Formulas in a form-driven flow (no code)

1. Open FintechOS Studio, open the main menu and select **Digital Experience > Customer Journeys > Form Driven Flows**.
2. Open the form driven flow you wish to work on.
3. Select the **Actions** tab, then click **Insert**.

- Click **Insert** and select the call formula with data mapping and select the formula you wish to execute and the mapping you have done.

Form Action

Action name

execute +

Call Formula with Data Mapping Select... Select...

- Click "**Save**" and then click "**Save and reload**".
- Continue to add as any as you have or need.
- Open each step and add the action by clicking the '**Flow control**'.

Entity Customer Form Demo1 Step Step1

Save and close Save and reload Open in code

1 General 2 UI 3 Flow Control 4 Advanced 5 Security Roles 6 Actions

Default Navigation Rule

☐ Close Flow ☒ Navigate to another Step ☐ Navigate to another Flow

Select step  
leave empty for default behaviour

Step3

Actions to be Performed

Call formula engine

☒ Select All

☒ Call formula engine

- In the section "Actions to be performed" add the action that calls the formula.
- Click **Save and Reload**.

### Track form driven flows that are using a specific formula

After you configure a form driven flow to call a formula, the form will be listed in the Form Tracking tab of the formula's edit form.

1 Details 2 History 3 Form Tracking

Entity	Name
Loan1	Cust

## Call formulas using server side scripts

Formulas can be called from server side scripts using the following method:

```
/**
 * Call formula by name
 * @param formulaName is the name of the formula
 * @param input is the input that must be provided in order to
compute the formula
 * @param options are the formula runtime options of type
IFtosRunFormulaOption. A property that can be set here is
referenceDate if you need to call a past version of the formula
 * @return the object with the calculated values
 */
server.formulas.runFormula(formulaName: string, input: any,
options: any): any
```

## Example

```
var input = {
  age: 20,
  region: "test"
}
server.formulas.runFormula("formulaName", input, {});
```

For more information, see the [Server SDK Reference Guide](#).

## Export a Formula or Data Set

To transfer a business formula on a different environment, use "[Digital Solutions Packages](#)" on [page 1086](#):

Deployment Package Component

Type

<input type="checkbox"/>	Name	Start date	End date	Business Status	Version	Digital Asset
<input type="checkbox"/>	test	09/03/2022 22:23		Active	1	
<input type="checkbox"/>	risks	03/03/2022 22:44		Draft	1	

When exporting/importing a business formula from one environment to another via a digital solution package:

- Only the latest version of the formula is exported.
- Before exporting, each formula can be configured to be either activated (default) or imported as draft on the destination environment.
- On import, the system compares the formula names, input parameters, and output steps on the destination environment with the ones in the package
  - If all are identical with those on the import package, the system concludes the formula to be imported is an update to the one found on the environment, and the formula to be deployed becomes the latest version of the existing formula.

### IMPORTANT!

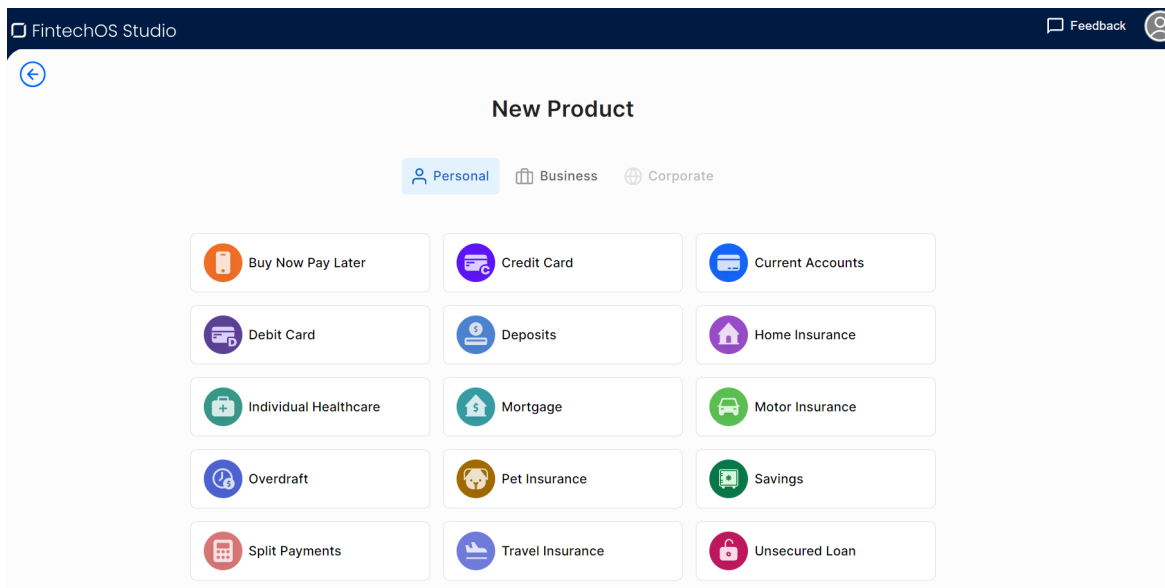
If the formula on the deployment environment is in a **draft state**, the system displays an error message on importing the package. The user must then remove or delete the draft version in order to deploy/import the formula from the package

- If the formula name matches, but the input and output parameters are different from the package formula, the formula is imported as a new one with the name changed to `name_importdate`.
- If the name differs, a new formula is created with version no. 1.
- The start date of the imported formulas/data sets are set to the import date.
- The **VersionStamp ID** is updated when reexporting a package that received a change. For example, a package with a formula is exported from one environment with v22.1.10 and imported on another environment with v22.1.9.3. The package has a certain VersionStamp ID. When a change is made on v22.1.10 and the package is

reimported on v22.1.9.3, the VersionStamp ID changes. This way, you can see that the package received a change before being reimported on the destination environment.

# Products

Product Factory is a central hub for managing your banking and insurance product portfolio. Its intuitive, sentence-based product designer lets you create products by customizing templates tailored to your business needs. Once a product design is finalized, you can approve it for production, track its performance using analytics, and enhance it through iterative updates and successive versions.



## Product Templates

The core of Product Factory is the Product Designer, FintechOS' solution for building banking and insurance products. The products are built using templates that allow you to customize key features like amounts, durations, fees, interests, etc., all streamlined in a unified user experience that is highly configurable and easy to use. You can either:

- Use ["Built-In Product Templates"](#) on the next page that are available out-of-the box for a variety of scenarios.
- Create your own ["Custom Product Templates"](#) on page 835 tailored to your specific business requirements.

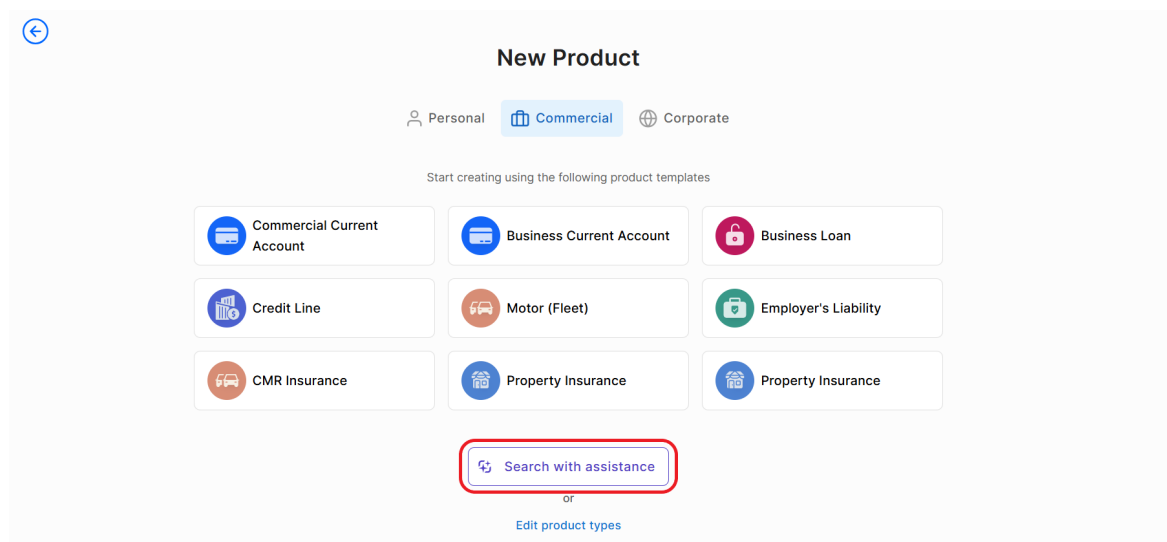
## Select a Product Template

To select a template for your product:

1. Go to **Main Menu > Product**.
2. Click **+ New Product**.
3. Browse the available templates and select one. If you need help, you can ["Use Dex to Select a Product Template"](#) below.

## Use Dex to Select a Product Template

["Dex"](#) on page 45 can help you select a template for your product that matches your requirements. Either click the Dex assistant button in the top right corner of FintechOS Studio, or, at the bottom of the **New Product** screen, click the **Search with Assistance** button.



1. Interact with Dex to provide details about the product you want to create.
2. After collecting enough information, Dex generates a list of relevant product templates. Click the **Open list** button to view the suggestions.
3. Hover over a proposed template and click either **Use** to pre-populate the product data based on the template or **Preview** to examine the product template further before making a decision.

## Built-In Product Templates

Product Factory comes with a variety of product templates that are available out-of-the-box for a variety of use cases (loans, insurance plans, deposits, mortgages, etc).

---

### Personal Buy Now Pay Later

Buy now, pay later (BNPL) is a type of short-term financing that allows consumers to make purchases and pay for them over time, usually with no interest. Use the Product Factory in the FintechOS Platform to create a BNPL loan product.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

#### Main Info

This section determines the amount, term, disbursement, **codebtors**<sup>1</sup>, **refinancing**<sup>2</sup>, and **restructuring**<sup>3</sup> characteristics of the loan.

---

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

<sup>2</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>3</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>
Codebtors	<ul style="list-style-type: none"> <li>• <b>Allows unlimited</b> - There is no limit on the number of codebtors that can guarantee the repayment.</li> <li>• <b>Allows up to</b> - There is a specified maximum number of codebtors allowed.</li> <li>• <b>Does not allow</b> - No codebtors are allowed.</li> </ul> <p><i>E.g.: Allows up to 2 codebtors.</i></p>

Parameter	Description
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>

### Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

☒ used in eligibility

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Buy Now Pay Later product. For details on versions, see ["Product Life Cycle" on page 848](#).

Below you can check a real-life example of how a Buy Now Pay Later product is built in Product Designer.

## Buy Now Pay Later Loan Use Case

1. For our use case, in the **Main Info** section, we've set the loan amount from 1,000 to 10,000 EUR, with a term of 1 to 6 months, and the auto-disbursement feature is enabled. Unlimited co-debtors, refinancing, and restructuring are not allowed, and there are no insurances required. The product is available from 08/30/2023 to 08/30/2028.

### Personal BNPL

BNPL857

Available from 08/30/2023 to 08/30/2028 ×

Main Info

Loan amount from 1000 € to 10000 € with a term of 1 months to 6 months

Does not allow codebtors

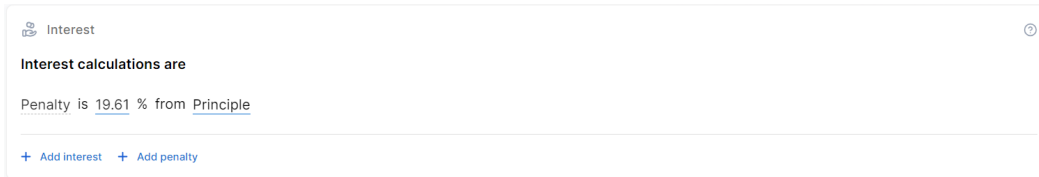
Does not allow refinancing

Does not allow restructuring

Already set up

Is autodisbursed

2. While there is no **Interest** applied for the duration of the loan, a fixed **penalty** interest of 19.61% from the loan principal is configured to apply, should the customers miss their payments on time.



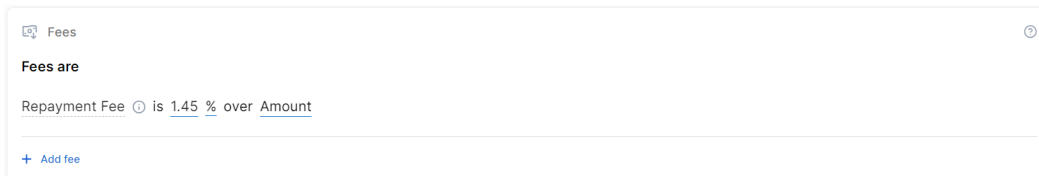
Interest

Interest calculations are

Penalty is 19.61 % from Principle

+ Add interest + Add penalty

3. In the **Fees** section, we've added a one-time non-returnable up-front fee of 1.45% applied to the loan's amount.



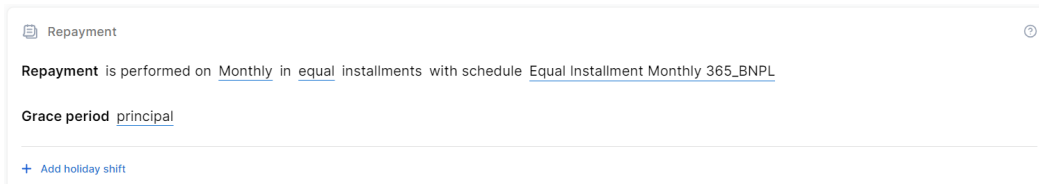
Fees

Fees are

Repayment Fee is 1.45 % over Amount

+ Add fee

4. The **Repayment** requirements are configured for monthly equal payments, with a pre-configured schedule of Equal Installment Monthly 365\_BNPL. There's also a Grace period added to the principal, so customers can safely delay their monthly payments.



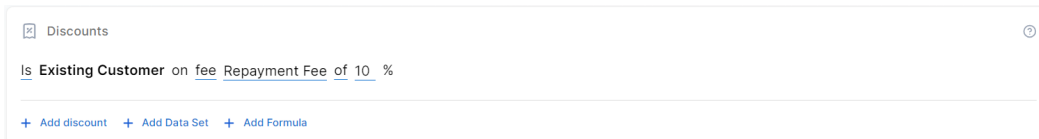
Repayment

Repayment is performed on Monthly in equal installments with schedule Equal Installment Monthly 365\_BNPL

Grace period principal

+ Add holiday shift

5. For **Discounts**, there is a 10% discount on the Repayment Fee, *IF* the customer applying for the loan is an Existing Customer.



Discounts

Is Existing Customer on fee Repayment Fee of 10 %

+ Add discount + Add Data Set + Add Formula

6. There are two **Underwriting** rules for the customer to be eligible for this type of product:

- Age must be over 21
- Length of Employment must be over 1 year

Underwriting

Age Has over 21 Years Old ...

Length of Employment is over 1 ...

+ Add rule + Add Data Set + Add Formula

## 7. The **Documents** section is configured as follows:

The Required documents are ID card and Income statement, which are mandatory for the customer;

The Provided documents are Terms and Conditions, mandatory to be signed also.

Documents

**Required Documents**

BI/Ci mandatory for debtor ...

Income Statement mandatory for debtor ...

+ Add document

**Provided Documents**

TermsAndConditions is static requires signature and

BNPL Terms&Conditions.docx ✓

+ Add document

## Personal Current Account

A personal current account is a bank account where customers can store and withdraw money for most day-to-day banking operations, like salary deposits, paying mobile phone bills, utility bills, online shopping, or direct debits.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

This section determines the deposit's currency, amount, and term. It also allows you to select options for withdrawal, **auto-renew**<sup>1</sup>, and top-ups.

Parameter	Description
Deposit amount and term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be deposited in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be deposited in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the deposit's term in either days, weeks, months, or years.</li> </ul> <p><i>E.g.: Deposit amount up to 100000 \$ with a term of 1 to 5 years.</i></p>
Withdrawal	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can take out money from the deposited funds before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot take out money from the deposited funds before maturity.</li> </ul>
Auto-renew	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The deposit can be auto-renewed.</li> <li>• <b>Does not allow</b> - The deposit cannot be auto-renewed.</li> </ul>
Top-ups	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can add additional funds to the deposit before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot add additional funds to the deposit before maturity.</li> </ul>

## Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes

---

<sup>1</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.
- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

E.g.: *Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

## Fees

To set up a **fee**<sup>2</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the the conditions under which the fee is applied, how

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

<sup>2</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).

3. Enter the amount of the fee:

- **value** - a fixed value in the specified currency.
- **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
- **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

You can set up multiple fees that will be charged independently, based on their ["Fee Types" on page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary (**+Add discount > Create New**).
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).

3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

**IMPORTANT!**

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

**Underwriting**

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

**HINT**

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a

manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "Underwriting Data Set Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "Multi-Dimensional Data Sets" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

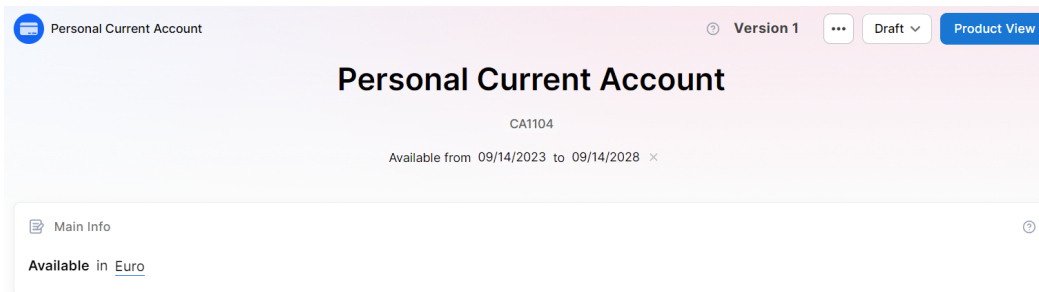
Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Current Account product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how a Personal Current Account product is built in Product Designer.

## Personal Current Account Use Case

1. For our use case, in the **Main Info** section, we've set the currency for the current account to Euro (EUR), and the product is available from 09/14/2023 to 09/14/2028.



Personal Current Account

Version 1 Draft Product View

### Personal Current Account

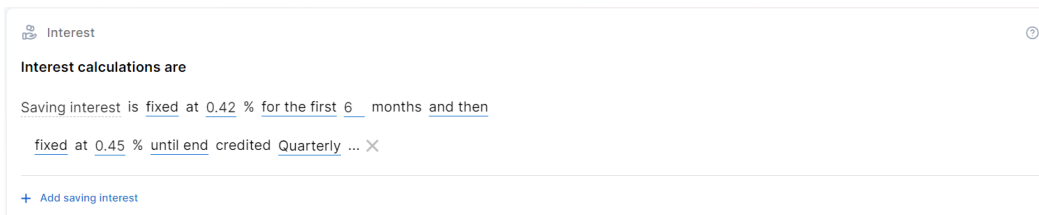
CA1104

Available from 09/14/2023 to 09/14/2028

Main Info

Available in Euro

2. In the **Interest** section, the current account is set to collect a fixed savings interest of 0.42% for the first 6 months (/deposits?), followed by a fixed savings interest of 0.45% until the end (account closure). The collected interest is set to be credited quarterly.



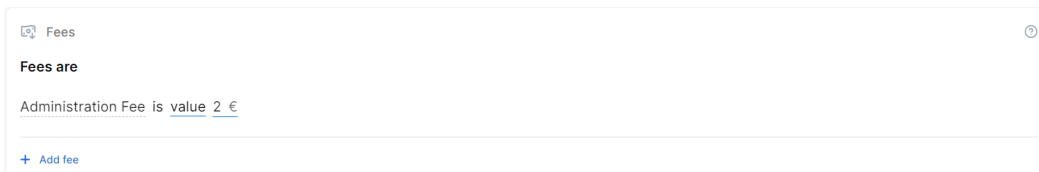
Interest

Interest calculations are

Saving interest is fixed at 0.42 % for the first 6 months and then fixed at 0.45 % until end credited Quarterly ...

+ Add saving interest

3. To provide predictability of the current account costs, in the **Fees** section, we've implemented a single monthly Administration Fee of 2 EUR.



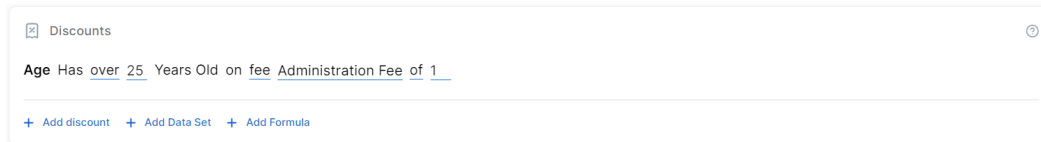
Fees

Fees are

Administration Fee is value 2 €

+ Add fee

4. The **Discounts** section is configured to apply a discount of 1 EUR on the Administration Fee configured earlier, only for customers who are over 25 years old.

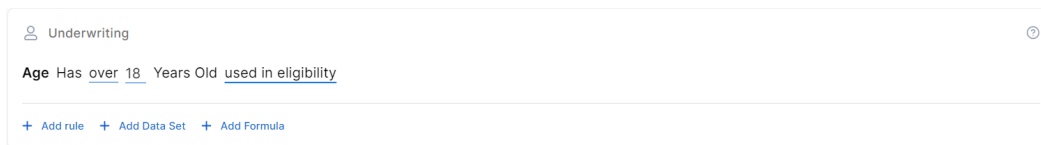


Discounts

Age Has over 25 Years Old on fee Administration Fee of 1

+ Add discount + Add Data Set + Add Formula

5. There is one **Underwriting** rule based on Age, used in eligibility, so that only persons over 18 years old are allowed to open a current account.

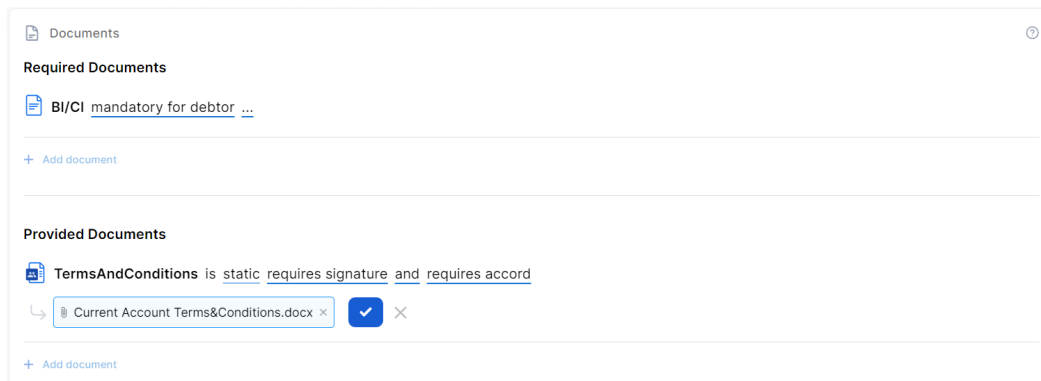


Underwriting

Age Has over 18 Years Old used in eligibility

+ Add rule + Add Data Set + Add Formula

6. Lastly, the **Documents** section is set to require a mandatory ID from the customer, as well as a signature and accord of the Terms & Conditions document provided.



Documents

**Required Documents**

BI/CI mandatory for debtor ...

+ Add document

**Provided Documents**

TermsAndConditions is static requires signature and requires accord

Current Account Terms&Conditions.docx ✓ ×

+ Add document

## Personal Debit Card

A debit card is a payment card linked to a customer's checking or savings account that can be used in place of cash to make purchases.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

This section determines the deposit's currency, amount, and term. It also allows you to select options for withdrawal, **auto-renew**<sup>1</sup>, and top-ups.

Parameter	Description
Deposit amount and term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be deposited in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be deposited in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the deposit's term in either days, weeks, months, or years.</li> </ul> <p><i>E.g.: Deposit amount up to 100000 \$ with a term of 1 to 5 years.</i></p>
Withdrawal	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can take out money from the deposited funds before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot take out money from the deposited funds before maturity.</li> </ul>
Auto-renew	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The deposit can be auto-renewed.</li> <li>• <b>Does not allow</b> - The deposit cannot be auto-renewed.</li> </ul>
Top-ups	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can add additional funds to the deposit before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot add additional funds to the deposit before maturity.</li> </ul>

## Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes

---

<sup>1</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.
- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

E.g.: *Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

## Fees

To set up a **fee**<sup>2</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the the conditions under which the fee is applied, how

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

<sup>2</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).

3. Enter the amount of the fee:

- **value** - a fixed value in the specified currency.
- **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
- **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

You can set up multiple fees that will be charged independently, based on their ["Fee Types" on page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary (**+Add discount > Create New**).
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).

3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

**IMPORTANT!**

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

**Underwriting**

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

**HINT**

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a

manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "Underwriting Data Set Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "Multi-Dimensional Data Sets" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Debit Card product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Personal Credit Card

A credit card is a payment card that allows the customer to make purchases by borrowing funds from the issuing bank up to a predetermined credit limit. The borrowed amount must be paid back at a later date, allowing the users to make purchases even if they don't have the funds immediately available. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the amount, term, disbursement, **refinancing**<sup>1</sup>, and **restructuring**<sup>2</sup> characteristics of the loan.

---

<sup>1</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>2</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description	Default
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>	
Disbursement	<ul style="list-style-type: none"> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>	
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>	Allows refinancing.
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>	Allows restructuring.

## Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+</b> <b>Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "Fee Types" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Credit Card product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how a Personal Credit Card product is built in Product Designer.

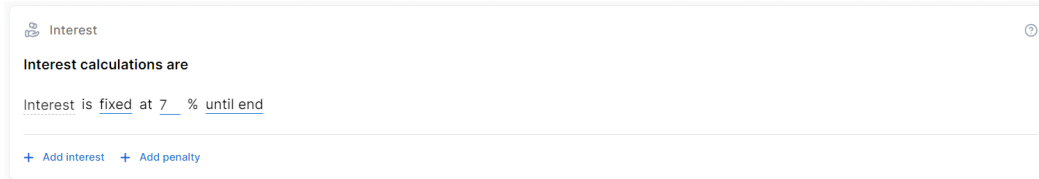
## Personal Credit Card Use Case

1. For our use case, in the **Main Info** section, we've set the loan amount for 500 to 5,000 EUR, with a term of 12 to 36 months. The product is not revolving or auto-disbursed, it doesn't allow refinancing or restructuring, and is available from 09/29/2023 to 09/29/2028.

The screenshot shows the 'Personal Credit Card' configuration page. At the top, there's a header with a menu icon, the title 'Personal Credit Card', and buttons for 'Version 1', 'Draft', and 'Product View'. Below the header, the title 'Personal Credit Card' is centered, followed by the ID 'CC1597' and the availability period 'Available from 09/29/2023 to 09/29/2028'. The main content area is titled 'Main Info' and contains the following configuration details:

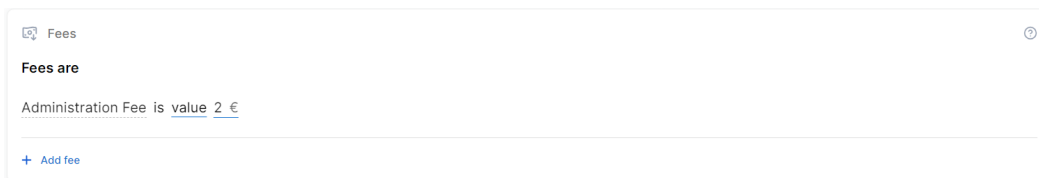
- Loan amount:** from 500 € to 5000 € with a term of 12 months to 36 months
- Is not revolving**
- Is not autodisbursed**
- Does not allow refinancing**
- Does not allow restructuring**

2. To provide predictability of the credit card costs, we've implemented a fixed type **Interest**, with a value of 7%, and no penalties.



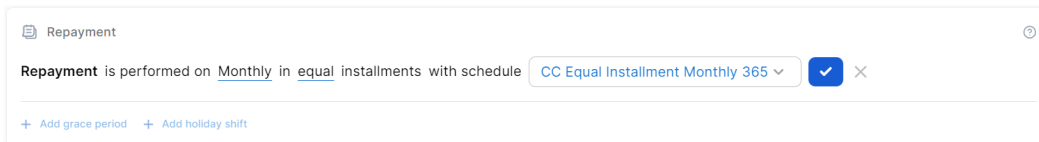
The screenshot shows the 'Interest' configuration panel. It has a title 'Interest' with a help icon. Below the title, it says 'Interest calculations are'. The main configuration line reads 'Interest is fixed at 7 % until end', where '7' is in a blue input field. At the bottom, there are two links: '+ Add interest' and '+ Add penalty'.

3. In the **Fees** section, we've implemented a monthly Administration Fee of 2 EUR.



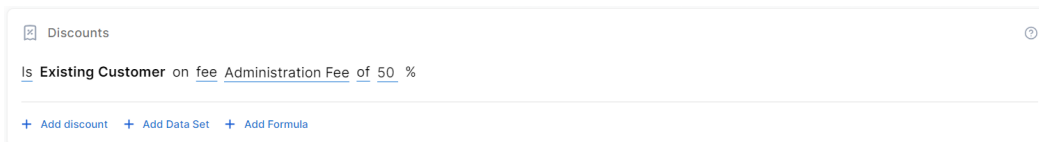
The screenshot shows the 'Fees' configuration panel. It has a title 'Fees' with a help icon. Below the title, it says 'Fees are'. The main configuration line reads 'Administration Fee is value 2 €', where '2' is in a blue input field. At the bottom, there is a link: '+ Add fee'.

4. The **Repayment** section is configured for monthly payments with the CC Equal Installment Monthly 365 schedule, and no grace period.



The screenshot shows the 'Repayment' configuration panel. It has a title 'Repayment' with a help icon. Below the title, it says 'Repayment is performed on Monthly in equal installments with schedule'. The schedule is set to 'CC Equal Installment Monthly 365' in a dropdown menu. To the right of the dropdown is a blue checkmark icon and an 'x' icon. At the bottom, there are two links: '+ Add grace period' and '+ Add holiday shift'.

5. For **Discounts**, there is a 50% discount on the Administration Fee, IF the customer applying for the credit card is an Existing Customer.



The screenshot shows the 'Discounts' configuration panel. It has a title 'Discounts' with a help icon. Below the title, it says 'Is Existing Customer on fee Administration Fee of 50 %', where '50' is in a blue input field. At the bottom, there are three links: '+ Add discount', '+ Add Data Set', and '+ Add Formula'.

6. In the **Underwriting** section, we've created two rules to filter out customers with County Court sentences or customers who have declared bankruptcy.

<add pic. (n/a yet)>

7. Lastly, the **Documents** section is set to require a mandatory ID from the customer, as well as a signature and accord of the Terms & Conditions document provided.

Documents

Required Documents

BI/Ci mandatory for debtor ...

+ Add document

Provided Documents

TermsAndConditions is static requires signature and requires accord

CreditCard Terms&Conditions.docx ✓

+ Add document

## Personal Deposits

Deposits are bank accounts in which customers can deposit and withdraw money. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the deposit's currency, amount, and term. It also allows you to select options for withdrawal, **auto-renew**<sup>1</sup>, and top-ups.

Parameter	Description
Deposit amount and term	<ul style="list-style-type: none"> <li><b>up to</b> - Sets the maximum amount that can be deposited in the specified currency.</li> <li><b>from ... to</b> - Sets a range for the amount that can be deposited in the specified currency.</li> <li><b>with a term of ... to</b> - Sets a time range for the deposit's term in either days, weeks, months, or years.</li> </ul> <p><i>E.g.: Deposit amount up to 100000 \$ with a term of 1 to 5 years.</i></p>

<sup>1</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

Parameter	Description
Withdrawal	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can take out money from the deposited funds before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot take out money from the deposited funds before maturity.</li> </ul>
Auto-renew	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The deposit can be auto-renewed.</li> <li>• <b>Does not allow</b> - The deposit cannot be auto-renewed.</li> </ul>
Top-ups	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can add additional funds to the deposit before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot add additional funds to the deposit before maturity.</li> </ul>

## Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

*E.g.: Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see "[Fee Types](#)" on page 900.
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on "[Product Formulas](#)" on page 865.

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on [page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Payment

The Payment section determines the capitalization for the deposit's interest, and how the payments are impacted by **holiday shifts**<sup>1</sup>.

Parameter	Description
Payment	<p>Sets up the payment capitalization options.</p> <ul style="list-style-type: none"> <li>• <b>Interest Actual/Actual No Capitalization</b> - Interest payments are not added to the deposit's principal amount.</li> <li>• <b>Interest Actual/Actual With Capitalization</b> - Interest payments are added to the deposit's principal amount.</li> </ul>

---

<sup>1</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

Parameter	Description
Holiday Shift	<p>Determines what happens if a payment date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul> <p><i>E.g.: Holiday shift forward following calendar for United States of America and United Kingdom defer due date.</i></p>

## Bonuses

In the **Bonuses** section, you can define bonuses to any of the already configured interest items, increasing the total payment value.

There are three ways to create bonuses:

1. Follow the sentence-based interface to configure a bonus based on a dictionary attribute (e.g., Age >18), for which you define the bonus. Note that you can also create new attributes, extending your dictionary (**+Add discount > New Attribute**).
2. Add a **Dataset** based on one, two, or more attributes for an interest item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define bonuses with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each interest item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary (**+Add discount > Create New**).
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a "Lexicon Term" on page 897 (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use "Product Formulas" on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use "Product Data Sets" on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "Underwriting Data Set Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set ?

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☐ 1st column is Miles per year ▼
 ☐ 1st row is Driving Experience ▼

used in eligibility ☒ ✕

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on [page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

## Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p>E.g.: <i>Income statement mandatory for debtor and mandatory for co-debtor.</i></p>

---

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p>E.g.: <i>Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Deposits product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Personal Secured Loan

A secured loan is a type of loan where the borrower pledges a **collateral**<sup>1</sup> (consisting of one or more **guarantees**<sup>2</sup>) in exchange for borrowing money. The collateral serves to secure the loan until its repayment. If the borrower defaults on the loan, the lender has the legal right to seize the guarantees to recover the outstanding debt.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the amount, term, **down payment**<sup>3</sup>, **disbursement**<sup>4</sup>, **codebtors**<sup>5</sup>, **refinancing**<sup>6</sup>, and **restructuring**<sup>7</sup> characteristics of the loan.

---

<sup>1</sup>An asset or property that the borrower pledges to the lender as security for the repayment of a loan, giving the lender the right to take ownership of the pledged asset in the event of default.

<sup>2</sup>Specific asset used to secure a loan as part of the collateral coverage.

<sup>3</sup>Down payment is an initial up-front partial payment for the purchase of items/services, such as a car or a house.

<sup>4</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

<sup>5</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

<sup>6</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>7</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description	Default
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>	
Down Payment	<ul style="list-style-type: none"> <li>• <b>Percentage</b> - Sets the up-front partial payment required for the loan.</li> </ul>	

Parameter	Description	Default
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>	
Codebtors	<ul style="list-style-type: none"> <li>• <b>Allows unlimited</b> - There is no limit on the number of codebtors that can guarantee the repayment.</li> <li>• <b>Allows up to</b> - There is a specified maximum number of codebtors allowed.</li> <li>• <b>Does not allow</b> - No codebtors are allowed.</li> </ul> <p><i>E.g.: Allows up to 2 codebtors.</i></p>	Allows unlimited codebtors.

Parameter	Description	Default
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>	Allows refinancing.
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>	Allows restructuring.

### Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Collateral

You can determine the **collateral**<sup>1</sup> cover used to secure a loan as a percentage of the loaned value. For example, a collateral cover of 25% indicates that the applicant must provide collateral in the amount of at least 25% of the loan amount.

1. Add the desired loan percentage in the **Collateral cover** field. Select the **allows partial release** option if you wish to allow the **partial release**<sup>2</sup> of collateral once certain conditions are met.

Once the collateral cover is set, the **+Add guarantee** button is enabled.

---

<sup>1</sup>An asset or property that the borrower pledges to the lender as security for the repayment of a loan, giving the lender the right to take ownership of the pledged asset in the event of default.

<sup>2</sup>A partial release of collateral means that if the borrower has met specific requirements or paid off a portion of the loan, the lender may release a part of the collateral while still holding the remaining part as security.

2. To configure the **guarantees**<sup>1</sup> that are part of the collateral, click **+Add guarante**. Select one of the available guarantees, and add the maximum accepted coverage as a percentage.
3. Add multiple guarantees, as needed.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on "[Product Data Sets](#)" on page 877).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a "[Lexicon Term](#)" on page 897 (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use "[Product Formulas](#)" on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use "[Product Data Sets](#)" on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "[Underwriting Data Set](#)").

---

<sup>1</sup>Specific asset used to secure a loan as part of the collateral coverage.

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Mortgage product. For details on versions, see ["Product Life Cycle" on page 848](#).

## Personal Bike Insurance

A personal bike insurance is an insurance policy designed to provide coverage for individuals who own bicycles. It protects against various risks and losses associated with the ownership and use of a bicycle, such as theft, accidental damage, third-party liability, vandalism, or loss during transit.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the <a href="#">"Insured Objects"</a> on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in <a href="#">"Product Formulas"</a> on page 865 e.g. for underwriting or pricing). Each attribute is based on a <a href="#">"Lexicon Term"</a> on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4edda; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas"](#) on page 865, allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖
Natural Earthquake, Floods, Hail, La...	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖
Crime Vandalism, Burglary/Robbery	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✖ Optional	<input checked="" type="checkbox"/> ✖ Optional
Liability coverage	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✖
Liability Third Party Liability			<input checked="" type="checkbox"/> ✖

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the `kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans` [Configuration Manger](#) key.

Override the Default Claim Rules

To customize the claim rules (such as the **deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three main plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has several coverage options listed below it, including 'Man-Made coverage', 'Intentional', 'Accidental', 'Natural coverage', 'Water Damage', and 'Others'. Each coverage option has a toggle switch and a pencil icon. A tooltip is visible over the '1' in the 'Natural coverage' row, indicating the deductible value: 15,000 €, indemnity limit: 1,000,000 €, waiting period: not set, and event limit: 1.

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan configuration panel in the FintechOS Studio. The panel displays the following fields: 'Deductible' (15000 €), 'Indemnity limit' (2000000 €), 'Waiting period' (months), 'Event limit' (1000000 €), and 'Event limit occurrence' (2). A 'Save' button is located at the bottom of the panel.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

**Add Dynamic Dimensions to an Insurance Plan**

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

### Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to activate your product and enable you to include it in ["Offers" on page 852](#). For details on versions, see ["Product Life Cycle" on page 848](#).

## Home Insurance

Home insurance is an insurance policy that provides coverage specific to homes, belongings, and the other structures or assets on a property.

To create a home insurance product, in FintechOS Studio, go to **Main Menu > Products > +New Product > Home Insurance**. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

---

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has a set of coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Man-Made coverage	Intentional	Accidental	Natural coverage	Water Damage	Others
Nature's Wrath	Off	On	On	On	On	On
The Evil that Men Do	On	On	On	On	On	On
Full	On	On	On	On	On	On

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan details and a side-panel for editing the 'Natural coverage' row. The side-panel contains fields for 'Deductible', 'Indemnity limit', 'Waiting period', 'Event limit', and 'Event limit occurrence', each with a text input and a 'Save' button.

Plan	Man-Made coverage	Intentional	Accidental	Natural coverage	Water Damage	Others
Nature's Wrath	Off	On	On	On	On	On
The Evil that Men Do	On	On	On	On	On	On
Full	On	On	On	On	On	On

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

#### Add Dynamic Dimensions to an Insurance Plan

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the ["Plan Dimension"](#) on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

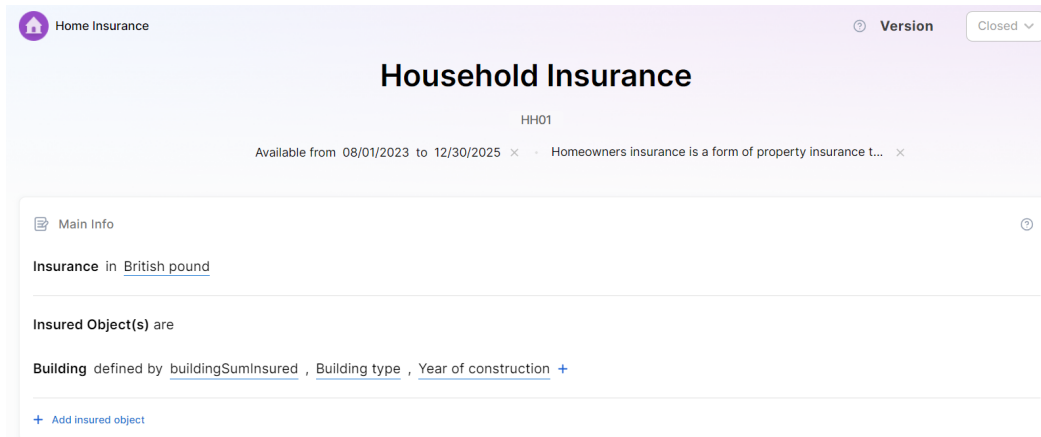
- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Home Insurance product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how a Home Insurance product is built in Product Designer.

## Household Insurance Use Case

1. For our use case, in the **Main Info** section, we've set the currency to British pound (GBP), the **Insured Object** is **Household**, defined by buildingSumInsured, Building type, and Year of construction. The product is available from 08/01/2023 to 12/30/2025.



2. In the **Coverages** section, we've added the following:

**Household** coverage with multiple sub-coverages:

- **FLEXA** sub-coverage: Aircraft Fall, Explosion, Fire, Lightning
- **Nat-Cat** sub-coverage: Earthquake, Floods, Land Slide

**Content** coverage divided in: Other Natural Disasters sub-coverage of Hail, Heavy Rain, Snow & Ice Pressure / Snow Avalanches, and Windstorm.

- **Water Damage** sub-coverage: Break of pipes, Water Damage caused by carelessness, Water Damage due to a neighbor.
- **Malicious people** sub-coverage: Political Risks and Vandalism
- **Theft** sub-coverage: Burglary and Robbery

**Home Assistance** coverage with indemnity limit of 5000 GBP and Assistance Services

**Third party liability** coverage with indemnity limit of 10000 GBP and with waiting period of 1 day, with a Liability sub-coverage and a Third Party Liability.

3. The **Underwriting** rules that determine the applicant's eligibility are:

- **Age** (must be over 18)
- **Household coverage**: must have an insuredValue between 50000 and 1000000 GBP
- **Content coverage**: must have an insuredValue between 1000 and 20000 GBP

Underwriting

Building rules are +

age is over 18 ...

Household coverage rules are +

insuredValue is between 50000 and 1000000 ...

Content coverage rules are +

insuredValue is between 1000 and 20000 ...

4. In the **Pricing** section, we've added a **Household coverage** with a FLEX formula based off insuredValue / 100, where insuredValue is defined from input.

Pricing

Building [add formula](#)

Household coverage [add formula](#)

FLEXA formula ⓘ

$\text{insuredValue} / 100$

All 1 attributes have been defined

Nat-Cat → Content coverage

5. Lastly, the **Documents** section is configured to require signature of the Terms and Conditions for Household coverage.

Documents

Required Documents

Terms and conditions for coverage Household

## Individual Healthcare

An individual health insurance plan is a policy that covers a person under the policy against medical bills and hospitalization costs. Can cover only the insurance holder or additional beneficiaries such as spouse, children, or other relatives..

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

---

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has a set of coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Man-Made coverage	Intentional	Accidental	Natural coverage	Water Damage	Others
Nature's Wrath	Off	On	On	On	On	On
The Evil that Men Do	On	On	On	On	On	On
Full	On	On	On	On	On	On

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' side-panel. It contains the following fields:

- Deductible:** 15000 €
- Indemnity limit:** 2000000 €
- Waiting period:** months
- Event limit:** 1000000 €
- Event limit occurrence:** 2

A 'Save' button is located at the bottom of the side-panel.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

**Add Dynamic Dimensions to an Insurance Plan**

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Individual Healthcare product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Personal Mortgage

A mortgage is a type of loan used to purchase or maintain a home, land, or other types of real estate. The borrower agrees to pay the lender over time, typically in a series of regular payments that are divided into principal and interest. The property serves as collateral to secure the loan until its repayment.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

This section determines the amount, term, **down payment**<sup>1</sup>, **disbursement**<sup>2</sup>, **codebtors**<sup>3</sup>, **refinancing**<sup>4</sup>, and **restructuring**<sup>5</sup> characteristics of the loan.

Parameter	Description	Default
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>	
Down Payment	<ul style="list-style-type: none"> <li>• <b>Percentage</b> - Sets the up-front partial payment required for the loan.</li> </ul>	

---

<sup>1</sup>Down payment is an initial up-front partial payment for the purchase of items/services, such as a car or a house.

<sup>2</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

<sup>3</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

<sup>4</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>5</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description	Default
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>	
Codebtors	<ul style="list-style-type: none"> <li>• <b>Allows unlimited</b> - There is no limit on the number of codebtors that can guarantee the repayment.</li> <li>• <b>Allows up to</b> - There is a specified maximum number of codebtors allowed.</li> <li>• <b>Does not allow</b> - No codebtors are allowed.</li> </ul> <p><i>E.g.: Allows up to 2 codebtors.</i></p>	Allows unlimited codebtors.

Parameter	Description	Default
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>	Allows refinancing.
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>	Allows restructuring.

### Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Collateral

You can determine the **collateral**<sup>1</sup> cover used to secure a loan as a percentage of the loaned value. For example, a collateral cover of 25% indicates that the applicant must provide collateral in the amount of at least 25% of the loan amount.

1. Add the desired loan percentage in the **Collateral cover** field. Select the **allows partial release** option if you wish to allow the **partial release**<sup>2</sup> of collateral once certain conditions are met.

Once the collateral cover is set, the **+Add guarantee** button is enabled.

---

<sup>1</sup>An asset or property that the borrower pledges to the lender as security for the repayment of a loan, giving the lender the right to take ownership of the pledged asset in the event of default.

<sup>2</sup>A partial release of collateral means that if the borrower has met specific requirements or paid off a portion of the loan, the lender may release a part of the collateral while still holding the remaining part as security.

2. To configure the **guarantees**<sup>1</sup> that are part of the collateral, click **+Add guarante**. Select one of the available guarantees, and add the maximum accepted coverage as a percentage.
3. Add multiple guarantees, as needed.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on "[Product Data Sets](#)" on page 877).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a "[Lexicon Term](#)" on page 897 (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use "[Product Formulas](#)" on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use "[Product Data Sets](#)" on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "[Underwriting Data Set](#)").

---

<sup>1</sup>Specific asset used to secure a loan as part of the collateral coverage.

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

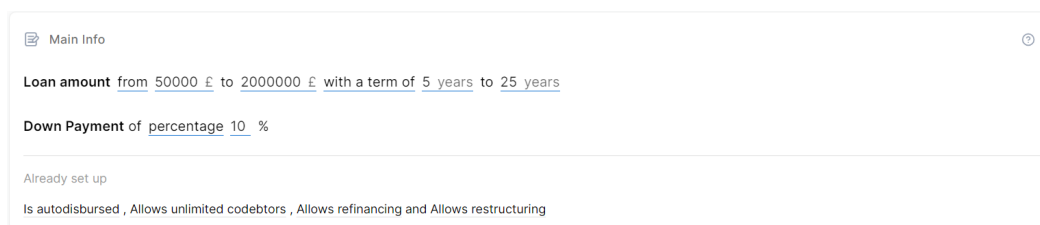
- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Mortgage product. For details on versions, see ["Product Life Cycle" on page 848](#).

Below you can check a real-life example of how a Mortgage product is built in Product Designer.

## Mortgage Use Case

1. For our use case, in the **Main Info** section, we've set the loan amount for 50,000 to 2,000,000 GBP, with a term of 5 to 25 years, and a down payment of 10%. Also, auto-disbursement, unlimited co-debtors, refinancing, and restructuring are allowed. The product's availability period was set from 06/01/2023 to 12/31/2024.



Main Info

Loan amount from 50000 £ to 2000000 £ with a term of 5 years to 25 years

Down Payment of percentage 10 %


Already set up

Is autodisbursed , Allows unlimited codebtors , Allows refinancing and Allows restructuring

2. The **Interest** calculations are based on variable rates, as follows:

The **Interest Variable** is a variable type indexed to BEBR (Bank of England Base Rate) with a margin of 1.22% for the first 24 installments, then:

- Variable indexed to BEBR with a margin of 1.25% for 60 installments, and
- Variable indexed to BEBR with a margin of 3.49% until end.

 Interest

**Interest calculations are**


Interest Variable is variable indexed to BEBR with a margin of 1.22 % for the first 24 installments and then

variable indexed to BEBR with a margin of 1.25 % for 60 installments and then

variable indexed to BEBR with a margin of 3.49 % until end

[+ Add interest](#)
[+ Add penalty](#)

3. We've set two **Commissions** for the mortgage loan: an **Administration Fee** of 0.28% over the Remaining Value, and a **Fee applied** of 0.26% over the Financed Value.

 Commissions

**Commissions are**


Administration Fee CZRML is 0.28 % over Remaining Value

Fee applied is 0.26 % over Financed Value

[+ Add comission](#)

4. For **Insurances** we've configured 3 Life Insurance policies based on the Age attribute, as follows:

- 10 GBP when age is between 18 and 35 years old.
- 10 GBP when age is between 18 and 35 years old.
- 20 GBP when age is over 51 years old.

 Insurances

**Insurances are**

Life Insurance of 10 £ when age is between 18 and 35 years old

Life Insurance of 15 £ when age is between 36 and 50 years old

Life Insurance of 20 £ when age is over 51 years old

[+ Add insurance](#)

5. The **Repayment** fee is performed on Months in equal installments, with the **Equal Installment Monthly 365\_TLU** schedule. The grace period applies to both the loan principal and the interest, with a holiday shift set to Forward.

Repayment

Repayment is performed on Months in equal installments with schedule Equal Installment Monthly 365\_TLU

Grace period both

Holiday shift forward

6. The only discount in the **Discounts** section is configured for Current Account on Interest, with a Interest Variable of 0.5%.

Discounts

Has Current Account on interest Interest Variable of 0.5 %

[+ Add discount](#)
[+ Add Dataset](#)
[+ Add Formula](#)

7. In the **Collaterals** section, we have a Collateral cover of 95% and a Residential Real Estate maximum accepted coverage of 95%.

Collaterals

Collateral cover of 95 %

Residential Real Estate max accepted coverage of 95 %

[+ Add guarantee](#)

8. The **Underwriting** section is configured as follows:

- Rule 1: Age is between 18 and 65 years old
- Rule 2: Is in BlackList
- Rule 3: Is First time buyer

Formula:  $\text{CurrentDTI} < \text{MaxDTI}$ , where  $\text{CurrentDTI} = (\text{ExistingInstallmentsAmount} + \text{LoanInstallmentAmount}) / \text{Income}$

The screenshot shows the 'Underwriting' configuration panel. It includes a title bar with a user icon and a help icon. Below the title, there are three rules defined for eligibility: 'Age Has between 18 and 65 Years Old used in eligibility', 'Is In BlackList used in eligibility', and 'Is First time buyer used in eligibility'. A 'Formula' section contains a text input field with the formula 'CurrentDTI < MaxDTI'. At the bottom, a status message says 'All 2 attributes have been defined ...'. Navigation buttons at the bottom are '+ Add rule', '+ Add Dataset', and '+ Add Formula'.

9. Lastly, for **Documents**, we have the ID that is mandatory for debtor and co-debtor in the **Required Documents** section, and a generated Consent file that requires signature and accord in the **Provided Documents** section.

The screenshot shows the 'Documents' configuration panel. It has a title bar with a document icon and a help icon. The panel is divided into two sections: 'Required Documents' and 'Provided Documents'. In the 'Required Documents' section, there is one entry: 'ID mandatory for debtor ...'. In the 'Provided Documents' section, there is one entry: 'Consent is generated requires signature ...'. Below this entry, there is a link 'with consent.docx' and a 'Details' button.

## Motor Insurance

Motor Insurance provides financial compensation to cover any injuries caused to people or their property. You have to have motor insurance before you can drive your vehicle in a public place. It protects you, your vehicle, and other motorists against liability in case there is an accident.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

---

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three main plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has several sub-coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, indicating fields for deductible, indemnity limit, waiting period, and event limit.

Plan	Sub-Coverage	Status	Details
Nature's Wrath	Man-Made coverage	Off	
	Intentional	On	1.000 €, 200.000 €, 3 m, 2 / 100.000 €
	Accidental	On	100.000 €, 1
The Evil that Men Do	Natural coverage	On	15.000 €, 1.000.000 €, 1
	Water Damage	On	
Full	Natural coverage	On	15.000 €, 2.000.000 €, 2 / 1.000.000 €
	Water Damage	On	

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' sub-coverage settings. The side-panel is open, displaying fields for Deductible, Indemnity limit, Waiting period, Event limit, and Event limit occurrence. The 'Save' button is visible at the bottom of the side-panel.

Field	Value
Deductible	15000 €
Indemnity limit	2000000 €
Waiting period	months
Event limit	1000000 €
Event limit occurrence	2

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

### Add Dynamic Dimensions to an Insurance Plan

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to activate your product and enable you to include it in ["Offers" on page 852](#). For details on versions, see ["Product Life Cycle" on page 848](#).

## Personal Overdraft

An overdraft is an extension of credit from a lending institution that is granted when an account reaches zero. The overdraft allows the account holder to continue withdrawing money even when the account has no funds in it or has insufficient funds to cover the amount of the withdrawal.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the overdraft limit amount and term:

- **up to** - Sets the maximum amount that can be loaned in the specified currency.
- **with a term up to** - Sets a maximum term for the overdraft repayment in either days, months, or years.

*E.g.: Overdraft limit amount up to 1000 EUR with a term up to 3 months.*

### Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.
- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

*E.g.: Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

The overdraft interest is charged on the account's negative balance (when the account holder withdraws funds in excess of the money available in the account).

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

Click **+ Add overdraft interest** to set up an interest for the overdraft amount based on the desired interest rate(s).

- **fixed** - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.
- **variable** - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.
- **based on formula** - Allows you to set interest rates based on ["Product Formulas" on page 865](#).
- **for the first / for** - Applies the interest rate for a specified number of **installments**<sup>1</sup>.
- **until end** - Applies the interest rate until the end of the loan term.

*E.g.: Overdraft Interest is fixed at 5% for the first 3 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.*

## Fees

To set up a **fee**<sup>2</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

### 3. Enter the amount of the fee:

- **value** - a fixed value in the specified currency.
- **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
- **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

You can set up multiple fees that will be charged independently, based on their ["Fee Types" on page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary (**+Add discount > Create New**).
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one

for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set](#)

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Overdraft product. For details on versions, see ["Product Life Cycle" on page 848](#).

## Pet Insurance

Pet insurance is a health plan the owner pays on a monthly or annual basis in exchange for reimbursement of eligible veterinary expenses. When a pet is seen by a veterinarian for a covered condition, the owner pays the vet in full and then submits a claim to their insurance company for reimbursement, which is dictated by the terms of the policy.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p>E.g.: <i>Household defined by Building Class, Building Materials, Property Condition.</i></p> <p>E.g.: <i>Car defined by Car Condition, Engine Type, Mileage.</i></p> <p>E.g.: <i>Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

### Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three main plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has several coverage options listed below it. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, indicating the deductible value.

Plan	Coverage	Details
Nature's Wrath	Man-Made coverage	Off
	Intentional	1.000 €, 200.000 €, 3 m, 2 / 100.000 €
	Accidental	100.000 €, 1
The Evil that Men Do	Natural coverage	15.000 €, 1.000.000 €, 1
	Water Damage	On
Full	Natural coverage	15.000 €, 2.000.000 €, 2 / 1.000.000 €
	Water Damage	On

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan configuration panel. It includes fields for 'Deductible', 'Indemnity limit', 'Waiting period', 'Event limit', and 'Event limit occurrence'. The 'Deductible' field is set to 15000, 'Indemnity limit' is 2000000, 'Waiting period' is 1 month, 'Event limit' is 1000000, and 'Event limit occurrence' is 2. A 'Save' button is at the bottom.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

**Add Dynamic Dimensions to an Insurance Plan**

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Home Insurance product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Personal Savings

A savings account is an interest-bearing deposit account held at a bank or other financial institution. It is a good option for parking cash for short-term needs.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the deposit's currency, amount, and term. It also allows you to select options for withdrawal, **auto-renew**<sup>1</sup>, and top-ups.

---

<sup>1</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

Parameter	Description
Deposit amount and term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be deposited in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be deposited in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the deposit's term in either days, weeks, months, or years.</li> </ul> <p><i>E.g.: Deposit amount up to 100000 \$ with a term of 1 to 5 years.</i></p>
Withdrawal	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can take out money from the deposited funds before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot take out money from the deposited funds before maturity.</li> </ul>
Auto-renew	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The deposit can be auto-renewed.</li> <li>• <b>Does not allow</b> - The deposit cannot be auto-renewed.</li> </ul>
Top-ups	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can add additional funds to the deposit before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot add additional funds to the deposit before maturity.</li> </ul>

### Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.
- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

E.g.: *Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see "[Fee Types](#)" on page 900.

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

### 3. Enter the amount of the fee:

- **value** - a fixed value in the specified currency.
- **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
- **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

You can set up multiple fees that will be charged independently, based on their ["Fee Types" on page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary **(+Add discount > Create New)**.
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one

for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set](#)

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Savings product. For details on versions, see ["Product Life Cycle" on page 848](#).

## Personal Split Payments

A split payment involves using multiple payment sources to settle the whole cost of a single transaction. For example, a person using two different credit cards to pay for an item.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the amount, **down payment**<sup>1</sup>, **disbursement**<sup>2</sup>, and **restructuring**<sup>3</sup> characteristics of the product.

---

<sup>1</sup>Down payment is an initial up-front partial payment for the purchase of items/services, such as a car or a house.

<sup>2</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

<sup>3</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description	Default
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>	
Down Payment	<ul style="list-style-type: none"> <li>• <b>Percentage</b> - Sets the up-front partial payment required for the loan.</li> </ul>	

Parameter	Description	Default
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>	
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>	Allows restructuring.

## Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on <a href="#">"Loan Periodicity" on page 892</a>.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the <a href="#">"Payment Schedule Type" on page 881</a>.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

☒ used in eligibility

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Split Payments product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Travel Insurance

Travel insurances provide financial protection against unexpected events that may occur while traveling, such as trip cancellations, interruptions, delays, medical emergencies, lost or delayed baggage, and other unforeseen circumstances.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the <a href="#">"Insured Objects"</a> on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in <a href="#">"Product Formulas"</a> on page 865 e.g. for underwriting or pricing). Each attribute is based on a <a href="#">"Lexicon Term"</a> on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4edda; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the `kv/<environment name>/mkexp-bff/appSettings/numberOfAllowedPlans` [Configuration Manger](#) key.

Override the Default Claim Rules

To customize the claim rules (such as the **deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has a set of coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Man-Made coverage	Intentional	Accidental	Natural coverage	Water Damage	Others
Nature's Wrath	Off	On	On	On	On	On
The Evil that Men Do	On	On	On	On	On	On
Full	On	On	On	On	On	On

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' side-panel. It contains the following fields:

- Deductible: 15000 €
- Indemnity limit: 2000000 €
- Waiting period: months
- Event limit: 1000000 €
- Event limit occurrence: 2

A 'Save' button is located at the bottom of the side-panel.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

**Add Dynamic Dimensions to an Insurance Plan**

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to activate your product and enable you to include it in ["Offers" on page 852](#). For details on versions, see ["Product Life Cycle" on page 848](#).

## Personal Unsecured Loan

An unsecured loan is a loan that doesn't require any type of **collateral**<sup>1</sup>. Instead of relying on the borrower's assets as security, lenders approve unsecured loans based on the borrower's creditworthiness. Examples of unsecured loans include personal loans or student loans.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

---

<sup>1</sup>An asset or property that the borrower pledges to the lender as security for the repayment of a loan, giving the lender the right to take ownership of the pledged asset in the event of default.

## Main Info

This section determines the amount, term, disbursement, **codebtors**<sup>1</sup>, **refinancing**<sup>2</sup>, and **restructuring**<sup>3</sup> characteristics of the loan.

Parameter	Description
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>

---

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

<sup>2</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>3</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>
Codebtors	<ul style="list-style-type: none"> <li>• <b>Allows unlimited</b> - There is no limit on the number of codebtors that can guarantee the repayment.</li> <li>• <b>Allows up to</b> - There is a specified maximum number of codebtors allowed.</li> <li>• <b>Does not allow</b> - No codebtors are allowed.</li> </ul> <p><i>E.g.: Allows up to 2 codebtors.</i></p>
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>

## Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

### Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Personal Unsecured Loan product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how an Unsecured Personal Loan product is built in Product Designer.

## Unsecured Personal Loan Use Case

1. For our use case, in the **Main Info** section, we've set the loan amount for 500 to 5,000 EUR, with a duration of 12 to 60 months. Also, auto-disbursement, unlimited co-debtors, refinancing, and restructuring are allowed. The product is available from 08/30/2023 to 08/30/2028.

### Personal Unsecured Loan

TLU866

Available from 08/30/2023 to 08/30/2028 ×

Main Info

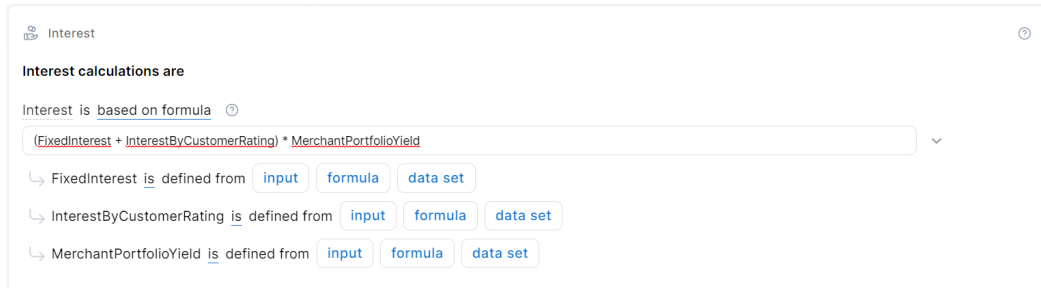
**Loan amount** from 500 € to 5000 € with a term of 12 months to 60 months

Already set up

Is autodisbursed , Allows unlimited codebtors , Allows refinancing and Allows restructuring

2. The **Interest** rate is based on a formula. Say the interest rate should be calculated using a fixed interest and increasing it by the customer's rating, while also considering the historical yield your financial institution had with a certain type of broker or merchant who sold the product to customers:

### Interest = (FixedInterest + InterestByCustomerRating) \* MerchantPortfolioYield



**Interest**

Interest calculations are

Interest is based on formula

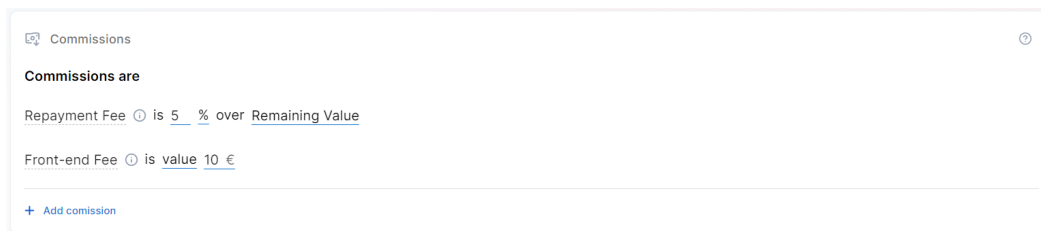
$(\text{FixedInterest} + \text{InterestByCustomerRating}) * \text{MerchantPortfolioYield}$

FixedInterest is defined from

InterestByCustomerRating is defined from

MerchantPortfolioYield is defined from

3. We've set two **Commissions** for the loan: a **Repayment Fee** of 5% over the Remaining Value, and a **Front-End Fee** of 10 EUR.



**Commissions**

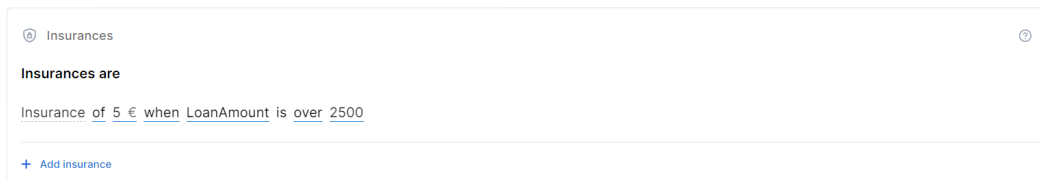
Commissions are

Repayment Fee is 5 % over Remaining Value

Front-end Fee is value 10 €

[+ Add commission](#)

4. The **Insurance** section, a **Monthly Insurance** is configured for 5 EUR if the loan amount exceeds 2,500 EUR.



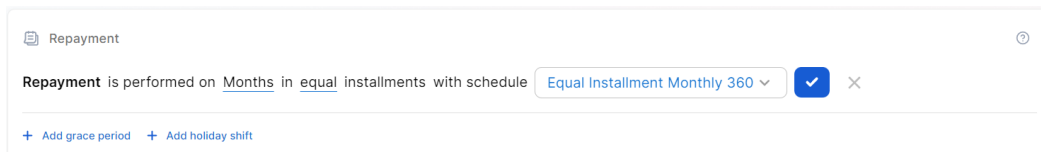
**Insurances**

Insurances are

Insurance of 5 € when LoanAmount is over 2500

[+ Add insurance](#)

5. The **Repayment** fee is set for equal installments, with a monthly 360 schedule.



**Repayment**

Repayment is performed on Months in equal installments with schedule Equal Installment Monthly 360 ☒ ×

[+ Add grace period](#) [+ Add holiday shift](#)

6. In the **Discounts** section, we've set a 10% discount for Interest, if the applicant's age is between 18 and 25 years old.

Discounts

Age Has between 18 and 25 Years Old on interest Variable Libor 1M06BFEF9CFAD of 10 %

+ Add discount

+ Add Dataset

+ Add Formula

7. The **Underwriting** section is configured as follows:

- Rule 1: Age higher than 18 years old
- Rule 2: Is not BlackListed
- Rule 3: Not in litigation
- Rule 4: CompanyState has following values bankruptcy, liquidation, active
- Formula:  $\text{CurrentDTI} < \text{MaxDTI}$ , where  $\text{CurrentDTI} = (\text{ExistingInstallmentsAmount} + \text{LoanInstallmentAmount}) / \text{Income}$

Underwriting

Building rules are +

age is over 18 ...

Household coverage rules are +

insuredValue is between 50000 and 1000000 ...

Content coverage rules are +

insuredValue is between 1000 and 20000 ...

- Dataset: FicoScore vs. Score

Dataset 

	ApplicationScoreDecision			
FicoScoreDecision		Rejected	Manual	Approved
	Rejected	Rejected	Rejected	Rejected
	Manual	Rejected	Manual	Manual
	Approved	Rejected	Manual	Approved
FicoScoreDecision				
	FicoScore	<520	520;720	>=720
		Rejected	Manual	Approved
ApplicationScoreDecision	ApplicationScore			
		<90	90;120	>=120
		Rejected	Manual	Approved

☐ 1st column is Selection 
☐ 1st row is Selection 

ApplicationScore =  
 AgeScore+MaritalStatusScore+EmploymentStatusScore+TimeWithCurrentEmployerScore+HighestDPDScore+NoActiveLoans12MonthsScore

## Personal Accidents

An personal accidents insurance plan is a policy that provides financial protection in the event of injuries, disabilities, or death resulting from an accident. It is focused on covering unforeseen events that cause bodily harm, typically due to external, violent, and accidental means.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

---

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three main plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has several sub-coverages listed below it. The 'Natural coverage' sub-coverage is selected, and a tooltip is shown over the pencil icon next to its deductible value of 15,000 €.

Plan	Sub-Coverage	Status	Deductible	Indemnity Limit	Waiting Period	Event Limit	Event Limit Occurrence
Nature's Wrath	Man-Made coverage	Off					
	Intentional	On	1,000 €	200,000 €	3 m	2 / 100,000 €	
	Accidental	On	100,000 €	1			
The Evil that Men Do	Natural coverage	On	15,000 €	1,000,000 €	1		
	Water Damage	On					
Full	Natural coverage	On	15,000 €	2,000,000 €	2	100,000 €	2
	Water Damage	On					

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' sub-coverage settings. The side panel is open, displaying fields for Deductible, Indemnity limit, Waiting period, Event limit, and Event limit occurrence. The 'Save' button is visible at the bottom of the side panel.

Field	Value
Deductible	15000 €
Indemnity limit	2000000 €
Waiting period	months
Event limit	1000000 €
Event limit occurrence	2

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

### Add Dynamic Dimensions to an Insurance Plan

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

### Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Individual Healthcare product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Commercial Business Current Account

A Business Current Account is a bank account that is used for business transactions. It can be opened by a business, allowing incoming and outgoing payments in the business's name.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the deposit's currency, amount, and term. It also allows you to select options for withdrawal, **auto-renew**<sup>1</sup>, and top-ups.

---

<sup>1</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

Parameter	Description
Deposit amount and term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be deposited in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be deposited in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the deposit's term in either days, weeks, months, or years.</li> </ul> <p><i>E.g.: Deposit amount up to 100000 \$ with a term of 1 to 5 years.</i></p>
Withdrawal	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can take out money from the deposited funds before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot take out money from the deposited funds before maturity.</li> </ul>
Auto-renew	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The deposit can be auto-renewed.</li> <li>• <b>Does not allow</b> - The deposit cannot be auto-renewed.</li> </ul>
Top-ups	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The depositor can add additional funds to the deposit before maturity.</li> <li>• <b>Doesn't allow</b> - The depositor cannot add additional funds to the deposit before maturity.</li> </ul>

## Interest

Interest on a savings or current account is the amount of money a bank or financial institution pays a depositor for storing their money with the bank. The financial institution pays the depositor a percentage of their account balance, and makes regular interest payments across the **statement**<sup>1</sup> cycle.

Click **+ Add saving interest** or **+ Add sight interest** to set up a long term or overnight interest rate respectively

---

<sup>1</sup>A bank statement summarizes all the account's monthly transactions and is typically sent by the bank to the account holder every month.

- **fixed** - Sets an interest rate that is a fixed percentage of the statement's amount.
- **for the first / and then** - Applies different interest rates over defined periods of time, in months.
- **until end** - Applies the interest rate until account closure.
- **based on formula** - Allows you to set savings interest rates based on "[Product Formulas](#)" on page 865.
- **credited** - Occurrence of interest being credited to the customer (Daily, Maturity, Monthly, Quarterly, Yearly).
- **capitalized** - Determines if the interest is calculated and credited to the main account, based on the total amount accumulated at each statement.

E.g.: *Savings Interest is fixed at 0.42% for the first 6 months and then fixed at 0.45% until end, credited Monthly, capitalized.*

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see "[Fee Types](#)" on page 900.

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

### 3. Enter the amount of the fee:

- **value** - a fixed value in the specified currency.
- **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
- **based on formula** - Allows you to set fees based on ["Product Formulas"](#) on [page 865](#).

You can set up multiple fees that will be charged independently, based on their ["Fee Types"](#) on [page 900](#). E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create new attributes, extending your dictionary (**+Add discount > Create New**).
2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets"](#) on [page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one

for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set](#)

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

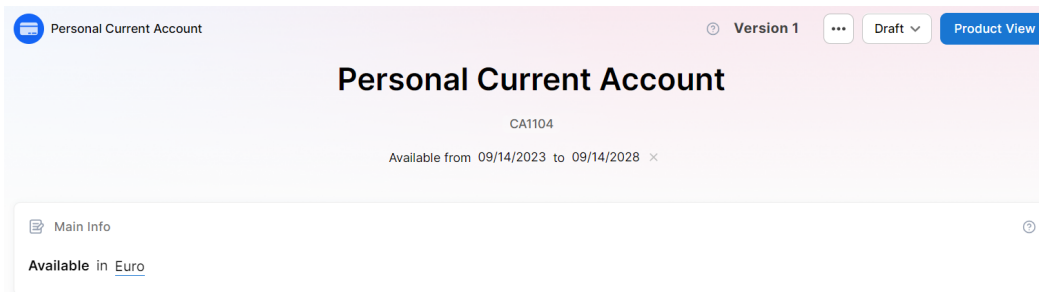
Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your SME Business Current Account product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how a SME Business Current Account product is built in Product Designer.

## SME Business Current Account Use Case

1. For our use case, in the **Main Info** section, we've set the currency for the current account to Euro (EUR), and the product is available from 09/14/2023 to 09/14/2028.



Personal Current Account

Version 1 Draft Product View

### Personal Current Account

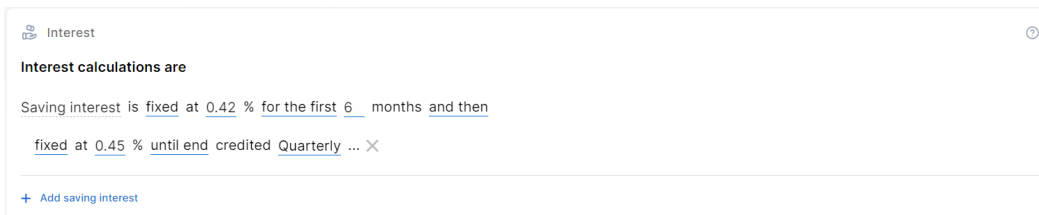
CA1104

Available from 09/14/2023 to 09/14/2028

Main Info

Available in Euro

2. In the **Interest** section, the current account is set to collect a fixed savings interest of 0.42% for the first 6 months (/deposits?), followed by a fixed savings interest of 0.45% until the end (account closure). The collected interest is set to be credited quarterly.



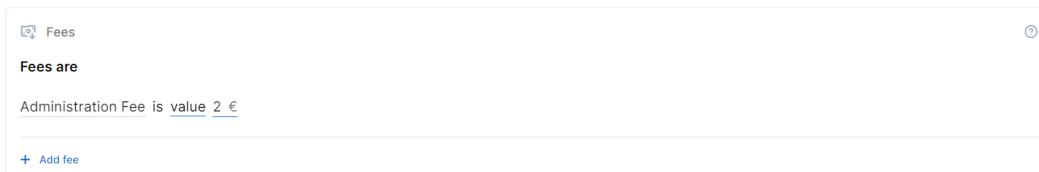
Interest

Interest calculations are

Saving interest is fixed at 0.42 % for the first 6 months and then fixed at 0.45 % until end credited Quarterly ...

+ Add saving interest

3. To provide predictability of the current account costs, in the **Fees** section, we've implemented a single monthly Administration Fee of 2 EUR.



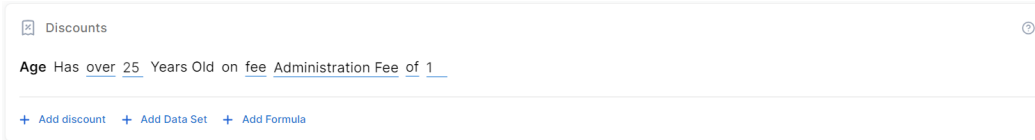
Fees

Fees are

Administration Fee is value 2 €

+ Add fee

4. The **Discounts** section is configured to apply a discount of 1 EUR on the Administration Fee configured earlier, only for customers who are over 25 years old.

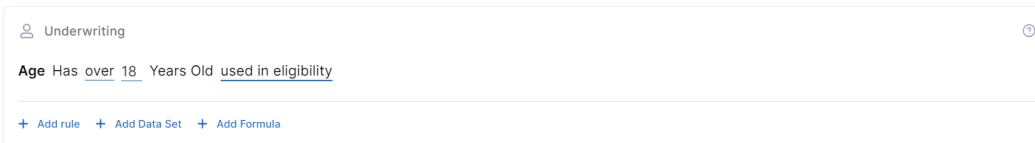


Discounts

Age Has over 25 Years Old on fee Administration Fee of 1

+ Add discount + Add Data Set + Add Formula

5. There is one **Underwriting** rule based on Age, used in eligibility, so that only persons over 18 years old are allowed to open a current account.

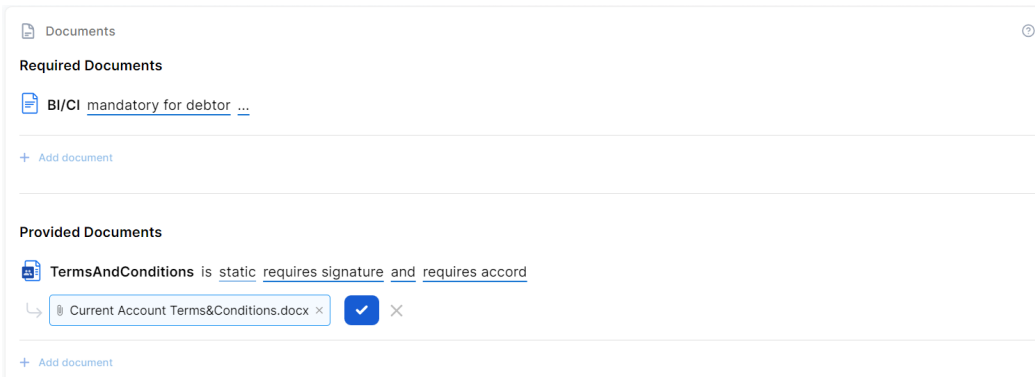


Underwriting

Age Has over 18 Years Old used in eligibility

+ Add rule + Add Data Set + Add Formula

6. Lastly, the **Documents** section is set to require a mandatory ID from the customer, as well as a signature and accord of the Terms & Conditions document provided.



Documents

Required Documents

BI/CI mandatory for debtor ...

+ Add document

Provided Documents

TermsAndConditions is static requires signature and requires accord

Current Account Terms&Conditions.docx ✓ ×

+ Add document

## Commercial Business Loan

A Business Loan is an unsecured loan used to finance business operations, expansion, equipment purchases, inventory, or working capital needs.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

This section determines the amount, term, disbursement, **codebtors**<sup>1</sup>, **refinancing**<sup>2</sup>, and **restructuring**<sup>3</sup> characteristics of the loan.

Parameter	Description
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.</i></p>

---

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

<sup>2</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>3</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> <li>• <b>Is autodisbursed</b> - Once the loan is approved, the loan amount is transferred to the borrower's bank account automatically.</li> <li>• <b>Is not autodisbursed</b> - Once the loan is approved, the borrower must request the transfer of the loan amount.</li> </ul>
Codebtors	<ul style="list-style-type: none"> <li>• <b>Allows unlimited</b> - There is no limit on the number of codebtors that can guarantee the repayment.</li> <li>• <b>Allows up to</b> - There is a specified maximum number of codebtors allowed.</li> <li>• <b>Does not allow</b> - No codebtors are allowed.</li> </ul> <p><i>E.g.: Allows up to 2 codebtors.</i></p>
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>

## Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on ["Product Data Sets" on page 877](#)).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

☒ used in eligibility

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your SME Business Loan product. For details on versions, see ["Product Life Cycle" on page 848](#).

## Commercial Credit Line

A Credit Line is a type of loan that allows a business to borrow money for funding working capital needs and ongoing operations. A credit line is especially helpful during times of revenue fluctuations since bills and unexpected expenses can be paid by drawing from the loan.

In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

This section determines the amount, term, revolving, disbursement, **refinancing**<sup>1</sup>, and **restructuring**<sup>2</sup> characteristics of the loan.

---

<sup>1</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>2</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

Parameter	Description	Default
Loan amount and loan term	<ul style="list-style-type: none"> <li>• <b>up to</b> - Sets the maximum amount that can be loaned in the specified currency.</li> <li>• <b>from ... to</b> - Sets a range for the amount that can be loaned in the specified currency.</li> <li>• <b>with a term of ... to</b> - Sets a time range for the loan repayment in either months or years.</li> </ul> <p><i>E.g.: Loan amount from 10,000 to 100,000 Euro with a term of 1 to 5 years.</i></p>	
Revolving (Is /Is not)	Allows a business to borrow money as needed for funding working capital needs and continuing operations such as meeting payroll.	Is
Disbursement	<ul style="list-style-type: none"> <li>• <b>Allows only one disbursement</b> - The borrower receives the loan amount in full in a single installment.</li> <li>• <b>Allows up to ... disbursements</b> - The borrower receives the loan amount in multiple installments, up to a fixed limit.</li> <li>• <b>Allows unlimited disbursements</b> - The borrower receives the funds as needed, without a set limit on the number of disbursements, up to the total loan amount.</li> </ul>	Allows only one disbursement.
Refinancing	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be refinanced.</li> <li>• <b>Does not allow</b> - The loan cannot be refinanced.</li> </ul>	Allows refinancing.

Parameter	Description	Default
Restructuring	<ul style="list-style-type: none"> <li>• <b>Allows</b> - The loan can be restructured.</li> <li>• <b>Does not allow</b> - The loan cannot be restructured.</li> </ul>	Allows restructuring.

### Interest

The interest sets up the amount of money (which is distinct from the principal repayments) that the borrower pays as a cost for borrowing money when repaying the loan. You can set up both regular and **penalty interests**<sup>1</sup>, with **interest rates**<sup>2</sup> that are fixed, variable or based on a formula.

---

<sup>1</sup>Additional interest charged when a borrower fails to make timely payments on a loan (misses a payment deadline or makes a payment that is less than the agreed-upon amount).

<sup>2</sup>The interest rate is the amount a lender charges a borrower and is a percentage of the principal (i.e., the amount loaned).

Parameter	Description
Interest	<p>Click <b>+ Add interest</b> to set up an interest for the loan based on the desired interest rate(s).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the principal amount, regardless of changes in the market interest rates.</li> <li>• <b>variable</b> - Sets an interest rate based on an underlying benchmark interest rate plus a specified percentage margin.</li> <li>• <b>based on formula</b> - Allows you to set interest rates based on <a href="#">"Product Formulas"</a> on page 865.</li> <li>• <b>for the first / for</b> - Applies the interest rate for a specified number of <b>installments</b><sup>1</sup>.</li> <li>• <b>until end</b> - Applies the interest rate until the end of the loan term.</li> </ul> <p><i>E.g.: Interest is fixed at 5% for the first 6 installments and then variable indexed to EURIBOR 3M with a margin of 1% until end.</i></p> <div> <p><b>NOTE</b></p> <p>If you define multiple interests, the user will have the option to select one of them during the application.</p> </div>

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

Parameter	Description
Penalty	<p>Click + <b>Add penalty</b> to set up a penalty interest for the loan and the reference value it is applied to (principal, interest, front end fee, etc.).</p> <ul style="list-style-type: none"> <li>• <b>fixed</b> - Sets an interest rate that is a fixed percentage of the reference value.</li> <li>• <b>based on formula</b> - Set an interest rate based on "<a href="#">Product Formulas</a>" on page 865 that is applied to the reference value.</li> </ul> <p>E.g.: <i>Penalty is 1.5% from Loan Interest.</i></p> <div> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• If you define multiple penalty interests, all will be applied. For instance, if you set up a 1.5% penalty from interest and a 1% penalty from principal, the borrower will pay both when missing a payment.</li> <li>• You can select multiple reference values in the same penalty interest, e.g.: <i>Penalty is 2% from Loan Principal, Loan Interest.</i> This will apply a 2% interest to both the loan principal and loan interest.</li> <li>• Once you set up a penalty interest based on a reference value, you cannot reuse that reference value in another penalty interest. E.g.: If you set up a penalty interest as a percentage of the principal, you cannot create another penalty interest that is also based on the principal.</li> <li>• Selecting the <b>General</b> reference value cumulates all the other available reference values, therefore it is mutually exclusive with them. If you create a penalty</li> </ul> </div>

Parameter	Description
	<p>interest based on the General reference value, you cannot create any other penalty interests. Likewise, if you create a penalty interest based on another reference value, you can no longer create a penalty interest based on the General reference value.</p>

## Fees

To set up a **fee**<sup>1</sup> for your product:

1. Click **+Add fee**.
2. Select a predefined type of fee from the list, or click **Create new** to define a new fee type (you can also rename an existing fee by clicking the fee name). The type of fee determines parameters such as the conditions under which the fee is applied, how often the fee is charged, whether the fee is refundable or not, etc. For more information, see ["Fee Types" on page 900](#).
3. Enter the amount of the fee:
  - **value** - a fixed value in the specified currency.
  - **percentage** - a specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set fees based on ["Product Formulas" on page 865](#).

---

<sup>1</sup>Fees are charges and costs that customers have to pay for specific services and transactions, such as processing a loan application, ongoing support, early repayment, or account maintenance. Fees differ from the main costs of a loan or insurance (the loan's principal/interest or the insurance's premium) in that they typically cover operational expenses and are not related to the loaned/insured amount.

You can set up multiple fees that will be charged independently, based on their "[Fee Types](#)" on page 900. E.g.:

- *Front-end fee is 25 \$.*
- *Repayment fee is 4% over remaining value.*

This will always charge the borrower a \$25 fee on loan application. If, during the loan, the borrower decides to repay the loan in advance, a 4% fee is charged over the loan's remaining value.

## Insurances

Insurances are required for borrowers that meet certain risk criteria, in order to cover the potential losses if they default on the loan.

To set up an insurance for your product:

1. Click **+Add insurance**.
2. Select the "[Bancassurance Class](#)" on page 891 (e.g.: **Credit Insurance**, **Life Insurance**, **Home Insurance** or **Other Insurance**).
3. Select the periodicity for the insurance premium, e.g.: **30 Days**, **Once**, **Monthly**, **Weekly**, **Trimestrial**, or **Annual**.
4. Select the amount to insure:
  - **value** - A fixed amount in the specified currency.
  - **percentage** - A specified percentage of either the remaining value, financed value, paid value, unused amount, used amount, overdraft limit amount, or amount.
  - **based on formula** - Allows you to set the insurance amount based on "[Product Formulas](#)" on page 865.
5. Select **when** the insurance should be issued based on the value of a specified "[Lexicon Term](#)" on page 897.

You can set up multiple insurances that will be issued independently, depending on whether they meet the issuance condition. E.g.:

- *Life insurance paid monthly of 100% over remaining value when Age is over 60 Years Old.*
- *Other Insurance paid once of 50% over financed value when In BlackList is In BlackList.*

This will issue a monthly insurance over the remaining loan value if the applicant is older than 60.

If the applicant is marked as *In BlackList*, which is a boolean lexicon term, an insurance of 50% over the financed value is issued when extending the loan.

#### NOTE

Remember to go to the **Provided Documents** section and add a document template to be signed by the customer for each insurance you've configured.

### Repayment

The Repayment section determines the periodicity and repayment schedule for the loan's **installments**<sup>1</sup>, and how the repayments are impacted by **holiday shifts**<sup>2</sup> and **grace periods**<sup>3</sup>.

---

<sup>1</sup>Regular payment that a borrower is required to make to the lender to repay a loan over time.

<sup>2</sup>Adjustment of installment due dates that coincide with public holidays by moving the due date to the last working day prior or first working day after the holiday.

<sup>3</sup>A specified period of time during which a borrower is not required to make regular loan payments without incurring late fees or penalties. Typically granted immediately after the loan disbursement or before the start of the regular repayment schedule.

Parameter	Description
Repayment	<p>Sets up the periodicity and schedule type of the repayment.</p> <ul style="list-style-type: none"> <li>• <b>performed on</b> - Determines the frequency of installments based on "Loan Periodicity" on page 892.</li> <li>• <b>equal/decreasing</b> - Determines if the installments are equal (by adjusting the principal to keep the sum between principal and interest constant on each installment) or decreasing (by repaying the same principal on every installment and having the interest decrease over time along with the outstanding principal).</li> <li>• <b>with schedule</b> - Determines the repayment schedule, based on the "Payment Schedule Type" on page 881.</li> </ul> <p><i>E.g.: Repayment is performed on monthly in equal installments with schedule Equal Installment Monthly 365_TLU.</i></p> <div> <p><b>NOTE</b></p> <p>When generating a repayment schedule, the default calculation precision for determining the remaining balance of each installment is set to 10 decimal places (rounded to 2 decimals in the final payment schedule). You can customize this precision between 2 and 28 decimal places by configuring the <b>PricingRoundingDecimals</b> key within the <code>kv/&lt;environment name&gt;/mkexp-pfai/app-settings</code> directory in the <a href="#">Configuration Manager</a>.</p> </div> <div> <p><b>NOTE</b></p> <p>You can add multiple repayment schedules that the applicant can choose from by clicking <b>+Add repayment</b> repeatedly.</p> </div>

Parameter	Description
Holiday Shift	<p>Determines what happens if an installment's due date overlaps a public holiday.</p> <ul style="list-style-type: none"> <li>• <b>forward</b> - Change the due date to the next working day after the holiday.</li> <li>• <b>backward</b> - Change the due date to the last working day before the holiday.</li> <li>• <b>country</b> - Select the country or countries used as reference for the public holidays calendar.</li> <li>• <b>defer due date</b> - The payment amount is calculated based on the normal due date, even if the actual due date is shifted.</li> </ul>
Grace Period	<p>Determines what expenses are exempted during the grace period and how long the grace period is.</p> <ul style="list-style-type: none"> <li>• <b>interest</b> - Borrower is exempt from repaying the interest.</li> <li>• <b>principal</b> - Borrower is exempt from repaying the principal.</li> <li>• <b>both</b> - Borrower is exempt from paying both the principal and the interest.</li> <li>• <b>installments</b> - The number of installments that the grace period covers.</li> </ul> <p><i>E.g.: Grace period interest 2 installments.</i></p>

## Discounts

In the **Discounts** section, you can define discounts on any of the already configured interest items, commission items, or on all pricing elements.

There are three ways to create discounts:

1. Follow the sentence-based interface to configure a condition based on a dictionary attribute (e.g., Age >18), for which you define a discount. Note that you can also create

new attributes, extending your dictionary (**+Add discount > Create New**).

2. Add a **Dataset** based on one, two, or more attributes for a pricing item. You can add more data sets, one for each pricing item. For more details on how to create Data Sets, see ["Product Data Sets" on page 877](#).
3. Add a formula to define discounts with more advanced conditions based on mathematical expressions, data sets, and other inputs. You can add more formulas, one for each pricing item. For more details on how to create formulas in Product Designer, see ["Product Formulas" on page 865](#).

### IMPORTANT!

A discount does NOT override the previous value of a pricing item, but is applied to it, decreasing the pricing item's final value. For example:

- A discount of 10% applied to an existing 10€ commission results in a new value of 9€ for the commission;
- A discount of 2€ applied to an existing 10€ commission results in a new value of 8€ for the commission.

## Collateral

You can determine the **collateral**<sup>1</sup> cover used to secure a loan as a percentage of the loaned value. For example, a collateral cover of 25% indicates that the applicant must provide collateral in the amount of at least 25% of the loan amount.

1. Add the desired loan percentage in the **Collateral cover** field. Select the **allows partial release** option if you wish to allow the **partial release**<sup>2</sup> of collateral once certain conditions are met.

Once the collateral cover is set, the +Add guarantee button is enabled.

---

<sup>1</sup>An asset or property that the borrower pledges to the lender as security for the repayment of a loan, giving the lender the right to take ownership of the pledged asset in the event of default.

<sup>2</sup>A partial release of collateral means that if the borrower has met specific requirements or paid off a portion of the loan, the lender may release a part of the collateral while still holding the remaining part as security.

2. To configure the **guarantees**<sup>1</sup> that are part of the collateral, click **+Add guarante**. Select one of the available guarantees, and add the maximum accepted coverage as a percentage.
3. Add multiple guarantees, as needed.

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for underwriting rules based on "[Product Data Sets](#)" on page 877).

There are three ways to add Underwriting rules:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a "[Lexicon Term](#)" on page 897 (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use "[Product Formulas](#)" on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use "[Product Data Sets](#)" on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the "[Underwriting Data Set](#)").

---

<sup>1</sup>Specific asset used to secure a loan as part of the collateral coverage.

Values" on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using "[Multi-Dimensional Data Sets](#)" on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

#### Documents

Specify the document types required from the applicants (and/or others involved in the origination process, e.g. **codebtors**<sup>1</sup>), as well as the document types provided to the applicants.

<sup>1</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.

Parameter	Description
Required from customer	<p>Documents that the applicant must provide in order to verify identity, income, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is <b>mandatory for applicant</b> and/or <b>mandatory for co-debtor</b>.</li> </ol> <p><i>E.g.: Income statement mandatory for debtor and mandatory for co-debtor.</i></p>
Provided to customer	<p>Documents that must be provided to the applicant typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired type of document.</li> <li>3. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in their original form, such as "General terms &amp; conditions". This will prompt you to upload a document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>4. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

The Service section is used to define the configuration structure and characteristics of your banking product. The following options are available:

- **Commission on unused amount:** The number of months after which the system starts calculating commissions for any unused amount.
- **Disbursement allowed:** The number of months for disbursing funds to the borrower.
- **Minimum principal for early repayment:** Minimal principal allowed for early repayment.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your SME Business Working Capital/Revolving product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Commercial Property Insurance

Property insurance is an insurance policy for business customers that provides coverage specific to properties and buildings, as well as their various structures or assets.

To create a property insurance product, in FintechOS Studio, go to **Main Menu > Products > +New Product > Business > Property Insurance**. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the <a href="#">"Insured Objects"</a> on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in <a href="#">"Product Formulas"</a> on page 865 e.g. for underwriting or pricing). Each attribute is based on a <a href="#">"Lexicon Term"</a> on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4edda; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas"](#) on page 865, allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

## Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖
Natural Earthquake, Floods, Hail, La...	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖	<input checked="" type="checkbox"/> ✖
Crime Vandalism, Burglary/Robbery	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✖ Optional	<input checked="" type="checkbox"/> ✖ Optional
Liability coverage	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> ✖
Liability Third Party Liability			<input checked="" type="checkbox"/> ✖

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three main plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has several coverage options with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Coverage	Status	Details
Nature's Wrath	Man-Made coverage	Off	
	Intentional	On	1.000 €, 200.000 €, 3 m, 2 / 100.000 €
	Accidental	On	100.000 €, 1
The Evil that Men Do	Natural coverage	On	15.000 €, 1.000.000 €, 1
	Water Damage	On	
Full	Natural coverage	On	15.000 €, 2.000.000 €, 2 / 1.000.000 €
	Water Damage	On	

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' settings panel. It includes fields for 'Deductible' (15000 €), 'Indemnity limit' (2000000 €), 'Waiting period' (months), 'Event limit' (1000000 €), and 'Event limit occurrence' (2). A 'Save' button is at the bottom.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

### Add Dynamic Dimensions to an Insurance Plan

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the ["Plan Dimension"](#) on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Home Insurance product. For details on versions, see "[Product Life Cycle](#)" on page 848.

Below you can check a real-life example of how a Home Insurance product is built in Product Designer.

## Household Insurance Use Case

1. For our use case, in the **Main Info** section, we've set the currency to British pound (GBP), the **Insured Object** is **Household**, defined by buildingSumInsured, Building type, and Year of construction. The product is available from 08/01/2023 to 12/30/2025.

Home Insurance

Version Closed

## Household Insurance

HH01

Available from 08/01/2023 to 12/30/2025 × Homeowners insurance is a form of property insurance t... ×

Main Info ⓘ

Insurance in British pound

Insured Object(s) are

Building defined by buildingSumInsured , Building type , Year of construction +

+ Add insured object

2. In the **Coverages** section, we've added the following:

**Household** coverage with multiple sub-coverages:

- **FLEXA** sub-coverage: Aircraft Fall, Explosion, Fire, Lightning
- **Nat-Cat** sub-coverage: Earthquake, Floods, Land Slide

**Content** coverage divided in: Other Natural Disasters sub-coverage of Hail, Heavy Rain, Snow & Ice Pressure / Snow Avalanches, and Windstorm.

- **Water Damage** sub-coverage: Break of pipes, Water Damage caused by carelessness, Water Damage due to a neighbor.
- **Malicious people** sub-coverage: Political Risks and Vandalism
- **Theft** sub-coverage: Burglary and Robbery

**Home Assistance** coverage with indemnity limit of 5000 GBP and Assistance Services

**Third party liability** coverage with indemnity limit of 10000 GBP and with waiting period of 1 day, with a Liability sub-coverage and a Third Party Liability.

3. The **Underwriting** rules that determine the applicant's eligibility are:

- **Age** (must be over 18)
- **Household coverage**: must have an insuredValue between 50000 and 1000000 GBP
- **Content coverage**: must have an insuredValue between 1000 and 20000 GBP

Underwriting

Building rules are +

age is over 18 ...

↳ Household coverage rules are +

insuredValue is between 50000 and 1000000 ...

↳ Content coverage rules are +

insuredValue is between 1000 and 20000 ...

4. In the **Pricing** section, we've added a **Household coverage** with a FLEX formula based off insuredValue / 100, where insuredValue is defined from input.

Pricing

Building [add formula](#)

↳ Household coverage [add formula](#)

↳ FLEX formula ⓘ

$\text{fx}$  insuredValue / 100

All 1 attributes have been defined

↳ Nat-Cat ↳ Content coverage

5. Lastly, the **Documents** section is configured to require signature of the Terms and Conditions for Household coverage.

Documents

Required Documents

Terms and conditions for coverage Household

## Commercial Employer's Liability

The Employer's Liability protects employers if an employee claims compensation for work-related injuries, illnesses, or damages caused by negligence, breach of duty, or failure to provide a safe working environment (e.g., unsafe workplace, lack of training, defective equipment).

To create an employer's liability insurance product, in FintechOS Studio, go to **Main Menu > Products > +New Product > Commercial > Employer's Liability**. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

### Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas" on page 865](#) that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets" on page 877](#) for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values" on page 903](#), e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets" on page 879](#) based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas" on page 865](#), allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

### Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has a set of coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a tooltip is visible over the '1' in the '15.000 €' field, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Man-Made coverage	Intentional	Accidental	Natural coverage	Water Damage	Others
Nature's Wrath	Off	On	On	On	On	On
The Evil that Men Do	On	On	On	On	On	On
Full	On	On	On	On	On	On

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' side-panel. It contains the following fields:

- Deductible: 15000 €
- Indemnity limit: 2000000 €
- Waiting period: months
- Event limit: 1000000 €
- Event limit occurrence: 2

A 'Save' button is located at the bottom of the side-panel.

3. Click **Save**.

## NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

#### Add Dynamic Dimensions to an Insurance Plan

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the "Plan Dimension" on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

## Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Home Insurance product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Commercial CMR

CMR insurance is an insurance policy for motor fleet owners that provides liability coverage for cargo loss or damage under the rules of the CMR convention. The CMR Convention is an international treaty that standardizes conditions for transporting goods by road across borders in participating countries.

To create a CMR insurance product, in FintechOS Studio, go to **Main Menu > Products > +New Product > Commercial > CMR**. In the title field, enter a name for the product. Optionally, you can click the ellipsis button (...) to also provide a description and/or an availability period. Below, fill in the following product configurations.

## Main Info

The Main Info section defines the currency used for the insurance's associated prices and values, the **cover term(s)**<sup>1</sup>, and the **insured object(s)**<sup>2</sup>.

Parameter	Description
Insurance	<p>Select the currency for the insurance's associated prices and values, and define the cover term(s) for the policy. You can configure up to three options for the cover term.</p> <p><i>E.g.: Insurance in Euro with a cover term of 1 year or 18 months or 2 years.</i></p>

---

<sup>1</sup>Period during which the insurance coverage is active and in effect. During this time, the insured party is protected under the terms of the policy, provided that the premiums are paid as required. Once the cover term expires, the policyholder is no longer protected unless the policy is renewed or extended.

<sup>2</sup>Item or entity that is covered by an insurance policy.

Parameter	Description
Insured Object(s)	<p>Choose the insured objects and their features that must be input by the applicant.</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add insured object</b> to select one of the "Insured Objects" on page 902 available in the product template.</li> <li>2. Choose any attributes of the insured object that must be input by the applicant (required in "Product Formulas" on page 865 e.g. for underwriting or pricing). Each attribute is based on a "Lexicon Term" on page 897 available in the lexicon context of the product template.</li> </ol> <div style="background-color: #d4e1d4; padding: 10px; margin: 10px 0;"> <p><b>HINT</b></p> <p>In the attributes' pop-up window, you can click <b>+New Attribute</b> to quickly add a new lexicon term or <b>Product Settings</b> to edit the current lexicon term.</p> </div> <p>If the product template has multiple insured objects available, you can repeat the process to add additional insured objects.</p> <p><i>E.g.: Household defined by Building Class, Building Materials, Property Condition.</i></p> <p><i>E.g.: Car defined by Car Condition, Engine Type, Mileage.</i></p> <p><i>E.g.: Property defined by Property Condition.</i></p>

## Coverages

Coverages configure the claim rules (**deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, and event limit) for the various **perils**<sup>4</sup> covered by the insurance.

- Each insured object can be divided into **coverages** based on its components. For instance, a home insurance can have coverages for the dwelling, garage, and personal belongings. Likewise, a health insurance can have separate coverages for the medical expenses and for the prescribed medications.
- Coverages can be divided into **sub-coverages** that group together perils of the same type. For instance, a dwelling coverage can have separate sub-coverages for natural disaster perils (such as earthquakes, floods, hurricanes) and for man-made perils (such as vandalism or burglary).

To set up coverages for your insurance, for each insured object:

Parameter	Description
Coverage	<p>Coverages allow you to break down the insured object into multiple components. For instance, a household insured object can be broken down into coverages for building, garage, courtyard, etc.</p> <p>In an insured object, click <b>+ Add coverage</b> and set a name for your coverage (default is <i>Untitled</i>).</p>

---

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

<sup>4</sup>Specific event or circumstance for which an insurance policy provides protection and financial compensation

Parameter	Description
Claim rules	<p>You can set up claim rules at either the coverage, sub-coverage, and/or peril level to configure the <b>disbursement</b><sup>1</sup> conditions:</p> <ul style="list-style-type: none"> <li>• <b>Indemnity limit:</b> The maximum amount to be paid for any claim or claims under the policy. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Deductible amount:</b> The amount of money that the insured party must pay before their insurance policy starts paying for covered expenses. Can be set as either a flat value, a percentage of a reference value, or based on "<a href="#">Product Formulas</a>" on page 865.</li> <li>• <b>Waiting period:</b> The amount of time an insured party must wait before some or all of their coverage come into effect.</li> <li>• <b>Event Limit:</b> The number of instances a claim can be made under the coverage or sub-coverage (the event limit cannot be set at the peril level). <ul style="list-style-type: none"> <li>◦ <b>Reimbursed ... each:</b> The maximum amount to be disbursed for a single event (the indemnity limit per event). Leave empty to allow flexible allocation of the indemnity amount between events.</li> </ul> </li> </ul> <p>E.g.: <i>Natural Disasters coverage with deductible of 1500 USD and with indemnity limit of 1000000 USD.</i></p> <p>E.g.: <i>Intentional Damage sub-coverage with deductible of 1000 USD and with indemnity limit of 200000 USD and with waiting period of 3 months and with event limit of 2 reimbursed 100000 USD each.</i></p>

---

<sup>1</sup>Payment or distribution of funds from an insurance company to a policyholder, beneficiary, or a third party, in accordance with the terms and conditions specified in the insurance policy.

Parameter	Description
Sub-coverages	<p>Sub-coverages group together specific types of perils, for instance FLEXA (Fire, Lightning, Explosion, Aircraft Fall) or Crime.</p> <p>Inside a coverage, click <b>+ Add sub-coverage</b> and set a name for your sub-coverage (default is <i>Untitled</i>).</p>
Perils	<p>Perils are specific events or circumstances for which the insurance policy provides protection and financial compensation. You can select perils like lightning, accidents, vandalism, etc. as defined in the <a href="#">"Insurance Peril" on page 895</a> settings.</p> <p>Inside a sub-coverage, click <b>+Add peril</b> and set a name for each event covered by the insurance (default is <i>Untitled</i>).</p> <div style="background-color: #d4edda; padding: 10px; margin-top: 10px;"> <p><b>HINT</b></p> <p>In the perils' pop-up window, you can click <b>Add perils in Product Settings</b> to open the <a href="#">"Insurance Peril" on page 895</a> settings and add a new peril.</p> </div>

## Underwriting

The underwriting rules determine an applicant's eligibility for the product and whether or not a manual approval process is required (available only for rules based on ["Product Data Sets" on page 877](#)). You can define underwriting rules at the insured object or at the coverage level to differentiate between eligibility for specific coverages or for the insured object as a whole.

To create a rule, click the **+** sign next to the desired insured object or coverage and select one of the following options:

- **Add Rule** - Follow the sentence-based interface to configure a condition based on a ["Lexicon Term" on page 897](#) (e.g. Property Condition is *Good* or *Excellent*).

### HINT

In the attributes' pop-up window, you can click **+New Attribute** to quickly add a new lexicon term or **Product Settings** to edit the current lexicon term.

- **Add Formula** - Use ["Product Formulas"](#) on page 865 that return a boolean result ("True" for approval and "False" for rejection);
- **Add Data Set** - Use ["Product Data Sets"](#) on page 877 for the evaluation. This is mandatory if the rule can return an outcome where the application must go through a manual approval process. The data set can return only the **Approved**, **Derrogation**, or **Rejected** results (or an equivalent terminology defined in the ["Underwriting Data Set Values"](#) on page 903, e.g. Passed, Manual Analysis, or Not Passed).

#### Data set

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☒ 1st column is Miles per year
☐ 1st row is Driving Experience

used in eligibility ☒ ☐

For each rule, you can select the **used in eligibility** option to mark it as a knock-out rule, which automatically disqualifies the applicant if its condition is not met. Otherwise, the rule is submitted to the final approval review.

For the manual approval result, you need to configure the journey to direct the application to a back-office manual approval process. If you are using ["Multi-Dimensional Data Sets"](#) on page 879 based on cascading data sets, the manual approval outcome must be defined in the top-level data set.

### Pricing

Pricing determines how the product's **insurance premiums**<sup>1</sup> are calculated. Pricing calculations are based on ["Product Formulas"](#) on page 865, allowing you to implement complex decision modeling. You can apply pricing formulas at the insured object,

---

<sup>1</sup>Amount of money the policy holder pays to the insurance company in exchange for insurance coverage. This payment is typically made on a regular basis, such as monthly or annually. It represents the cost of obtaining and maintaining insurance protection.

coverage, and/or sub-coverage level.

1. Select **add formula** for the chosen insured object, coverage, or sub-coverage.
2. Select a predefined formula, or write your own (see ["Product Formulas" on page 865](#) for details).

#### NOTE

If you try to add an attribute that does not exist, the platform will add it as a ["Lexicon Term" on page 897](#) with your definition so you can use it again. This way, you can expand and customize the lexicon based on your product requirements.

### Plans

Plans determine coverage variations that you want to include in your product's ["Offers" on page 852](#). They are based on the [coverages and sub-coverages](#) you have set up in your product.

An initial plan with all coverages disabled is set up by default. To configure your full commercial offering:

1. Click the **+ Add Plan** button to add additional plans.
2. Provide a name for each plan.
3. For each plan, toggle the available coverages and sub-coverages and mark the ones that are optional.

Plans

Basic

Extra

Full

+

Labor and Parts coverage

Natural

Earthquake, Floods, Hail, La...

Crime

Vandalism, Burglary/Robbery

Optional

Liability coverage

Liability

Third Party Liability

NOTE

By default, you can configure up to 4 plans. You can adjust the maximum number of plans using the kv/<environment name>/mkexp-bfff/appSettings/numberOfAllowedPlans Configuration Manger key.

Override the Default Claim Rules

To customize the claim rules (such as the deductible<sup>1</sup>, indemnity limit<sup>2</sup>, waiting period<sup>3</sup>, or event limits) of a coverage or sub-coverage for a specific plan:

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.  
<sup>2</sup>Maximum amount of the insurer’s liability for any claim or claims under a policy.  
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

1. Click the pencil icon next to the desired coverage or sub-coverage.

The screenshot shows the 'Plans' section of the FintechOS Studio. It displays three plans: 'Nature's Wrath', 'The Evil that Men Do', and 'Full'. Each plan has a set of coverages with toggle switches and a pencil icon for editing. The 'Natural coverage' row is highlighted, and a red circle is drawn around the pencil icon next to it. A tooltip is visible over the 'Natural coverage' row, showing details: 'Deductible value: 15.000 €, Indemnity limit: 1.000.000 €, Waiting period: not set, Event limit: 1'.

Plan	Coverage	Status	Details
Nature's Wrath	Man-Made coverage	Off	
	Intentional	On	1.000 €, 200.000 €, 3 m, 2 / 100.000 €
	Accidental	On	100.000 €, 1
The Evil that Men Do	Natural coverage	On	15.000 €, 1.000.000 €, 1
	Water Damage	On	
Full	Natural coverage	On	15.000 €, 2.000.000 €, 2 / 1.000.000 €
	Water Damage	On	

2. In the side-panel that appears, modify the **Deductible**, **Indemnity limit**, **Waiting period**, **Event limit**, and **Event limit occurrence** fields as needed.

The screenshot shows the 'Full' plan's 'Natural coverage' side-panel. It contains several input fields for configuring the coverage: 'Deductible' (15000 €), 'Indemnity limit' (2000000 €), 'Waiting period' (months), 'Event limit' (1000000 €), and 'Event limit occurrence' (2). A 'Save' button is at the bottom.

Field	Value
Deductible	15000 €
Indemnity limit	2000000 €
Waiting period	months
Event limit	1000000 €
Event limit occurrence	2

3. Click **Save**.

### NOTE

The updated claim rules remain persistent even if you toggle the coverage or sub-coverage on and off within the plan.

**Add Dynamic Dimensions to an Insurance Plan**

Dynamic dimensions, also referred to as Plan Dimensions, allow you to define additional amounts that fine-tune claims or other financial parameters more granularly at the plan level. To add a dynamic dimension:

1. Click the ellipsis button (...) next to the plan name.
2. Click **Dynamic dimensions** to open the side panel.
3. If the desired dimension doesn't exist yet:
  - a. Open the **Create Dimension** tab.
  - b. Provide a **Name** for the new dimension.
  - c. If you wish to be able to define the dimension as a percentage of another reference value, select the **Allows percentage value** checkbox.
  - d. Click **Save**.
4. In the **Dimensions** tab, click **+ Add value**.
5. Select the desired dimension.
  - If the dimension has a predefined value:
    - a. Leave the **value** option selected
    - b. Fill in the desired amount.
  - If the value is a percentage of another reference value:
    - a. Select the **percentage** option.
    - b. Fill in the percentage value.
    - c. Select the lexicon term that represents the reference value.
6. Click **Save**.

**HINT**

You can also manage your dynamic dimensions from the ["Plan Dimension"](#) on [page 903](#) settings menu.

**Payment**

The Payment section configures the payment frequency for the insurance premium. You can set up multiple payment options such as weekly, monthly, or one-time payments, allowing applicants to select their preferred payment frequency from the available periodicities.

E.g.: *Payments are taken Monthly.*

**Documents**

Specify the document types required from the customer, as well as the document types provided to the customer. These may include personal identification documents such as driving licenses, passports or ID cards, or other documents related to the insured object.

Document requests or provisions can be tailored based on the product, the insured object, or the coverage(s) sold.

Parameter	Description
Required from customer	<p>Documents that the customer must provide in order to verify identity, income, ownership, product eligibility, etc. To add a required document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> </ol> <p>E.g.: <i>Property title for Insured object Household.</i></p>

Parameter	Description
Provided to customer	<p>Documents that must be provided to the customer typically in order to obtain an agreement and/or signature. To add a provided document:</p> <ol style="list-style-type: none"> <li>1. Click <b>+Add document</b>.</li> <li>2. Select the desired document type.</li> <li>3. Select what the document applies to (<b>product, insured object, or coverage</b>).</li> <li>4. Choose if the document is: <ul style="list-style-type: none"> <li>• <b>static</b>: available for download during the journey in a fixed form, such as "General terms and conditions". This will prompt you to upload the corresponding document file.</li> <li>• <b>generated</b>: based on templates with specific tokens (the document templates must be previously configured using the <a href="#">Digital Documents Processor</a>). This will prompt you to select a <a href="#">digital document template</a>.</li> </ul> </li> <li>5. Choose if the document <b>requires signature</b> and/or <b>requires accord</b>.</li> <li>6. Select to either <b>Drop your file</b> or <b>click to upload</b>.</li> </ol> <p><i>E.g.: Terms and conditions is static requires accord and requires signature.</i></p>

### Service

Configure how the product will be serviced with regards to policy details, claims processing, servicing and managing quotes, or billing and collection operations:

## Billing & Collection

Use Billing & Collection to configure insurance premium payments.

- **Payment methods:** Specify the payment methods accepted for premiums, including the primary method.
- **Write-off:** Choose a premium greater than or less than the invoiced amount, or none.
- **Grace period:** Determine the number of days until penalties are imposed.
- **Invoice generation:** Choose when the premium invoice should be generated.
- **Payment schedule:** Select the configured payment schedule that should apply for the product. [Schedule types](#) are configured via **FintechOS Portal > Main Menu > Payment Schedule Type**.

## Claims management

Use Claims Management to configure how to process claims for compensation.

- **Update Indemnity limit:** Determine if a policy's insured amount should be adjusted after each paid claim.
- **Notification limit:** The maximum time that loss notices can be registered.

## Policy Admin

Use Policy Administration to manage the insurance policy and contract.

- **Renewal:** Configure the automated policy renewal process, if necessary.
- **Suspension duration:** Set the duration of the policy suspension, in days.
- **Free withdrawal:** Determine the days the policyholder can terminate and receive a full refund.
- **Pro rata calculation:** Define the calculation and duration of the Pro Rata.

- **Tariff:** Select whether the tariff is calculated on a per-coverage or per-product basis.
- **Underwriting:** Select whether the underwriting rules should be applied per coverage or per product.
- **Policy parties:** Select which previously configured parties shall be used for your product.

## Quote Admin

Use Quote Admin for quote management.

- **Issue Master Quote:** Determine whether the product has master quotes and, if so, how the tariff and underwriting regulations apply.

Once you've configured all the fields, change the status from **Draft** to **Approved** to save your Home Insurance product. For details on versions, see "[Product Life Cycle](#)" on page 848.

## Custom Product Templates

In addition to the "[Built-In Product Templates](#)" on page 528, you can create custom templates tailored to your business requirements. This can be done by duplicating an existing template (either a built-in template or a previously created custom template), then adding or removing sections and items according to your needs.

### Create a Custom Product Template

1. In FintechOS Studio, go to **Main Menu > Products**.
2. From the **Settings** drop-down, select **Product Types**.
3. Double click an existing product type to open it.
4. From the ellipsis menu (...), select **Duplicate**.

5. Provide a name for the new template and click **Duplicate**.
6. The template is created and opened in a "Draft" on page 840 state. As long as the template stays in this state you can edit it.

## Edit a Product Template

Any product template that is in a Draft state can be customized:

- Click the title and edit the text field to rename the **template title**.
- Click the icon at the top left corner of the screen and select one of the available options to change the **template icon**.
- Use the drop-down at the top right corner of the screen to switch the template class between **Personal**, **Business**, and **Corporate**.
- Select **Show the class name as part of the product type name** to prepend the class name to the template name when editing a product in the Product Designer. E.g.: A product in the *Personal* class that uses the *Holiday Loan* template will be displayed as *Personal Holiday Loan* in the designer.

### Customize Product Template Sections

#### IMPORTANT!

Insurance product template sections cannot be customized. Insurance product templates always inherit their sections from the duplicated template. You are only allowed to reorder insurance product sections (with the exception of the Main Info section which is always the first).

For banking product templates, you can customize the displayed sections.

#### Customize the Main Info Section

The Main Info section is always displayed at the top of the template. Unlike the other sections, it cannot be removed or repositioned further down the page.

Customize the fields and default values you wish to display in the Main Info section.

Main statement is Loan up to value

Includes down payment ☒

Use default statements ☒

☒ Use autoDisbursed with value True

☒ Use refinancing with value True

☒ Use withdrawal with value True

☐ Use topUps with value True

☒ Use disbursements with value Unlimited

☒ Use restructuring with value False

☐ Use revolving with value True

☐ Use autoRenew with value True

☐ Use rollOver with value True

☐ Use codebtors with value Unlimited

- **Main statement is** - This option allows you to configure the currency, amount limits, and time ranges for loan repayments or deposit terms.
  - Amount and Term Interval - For loan products, sets limits for the amount that can be loaned in a specified currency and the repayment time range. E.g.: *Loan amount from 1,000 to 10,000 Euro with a term of 1 to 5 years.*
  - Deposit Amount and Term Interval - For deposits, sets limits for the amount that can be deposited in a specified currency and the time range for the deposit's term. E.g.: *Deposit amount up to 100000 \$ with a term of 1 to 5 years.*
  - Only Currency - For products where you only specify the currency with no limits on amounts or time ranges.
  - Loan up to value - For loan products where you only set limits for the amount that can be loaned in a specific currency, but don't set limits for the repayment time range.
- **Includes down payment** - Adds a field to configure an up-front partial payment (e.g.: for loan products such as mortgages).
- **Use default statements** - Additional options you wish to include in the main section. Tick the checkmark on the left of the option to include the option in the template and configure the setting on the right of the option to set up a default value for it.

Option	Description
Use autoDisbursed with value	Configure if, once a loan is approved, the loan amount is transferred to the borrower's bank account automatically.
Use refinancing with value	Configure loan <b>refinancing</b> <sup>1</sup> .
Use withdrawal with value	Configure if a depositor can take out money from the deposited funds before maturity.
Use topUps with value	Configure if a depositor can add additional funds to a deposit before maturity.
Use disbursements with value	Configure limits on the number of disbursements.
Use restructuring with value	Configure if a loan allows <b>restructuring</b> <sup>2</sup> .
Use revolving with value	Configure if a business can borrow money as needed to fund working capital needs or continuing operations (e.g. meeting payroll).
Use autoRenew with value	Configure deposit <b>auto-renewals</b> <sup>3</sup> .
Use rollOver with value	Configure automatic <b>roll-over</b> <sup>4</sup> .
Use codebtors with value	Configure loan <b>codebtors</b> <sup>5</sup> .

#### Customize Regular Sections

To add a section to the template, scroll down to the bottom of the page and select it from the **Add section** drop-down.

---

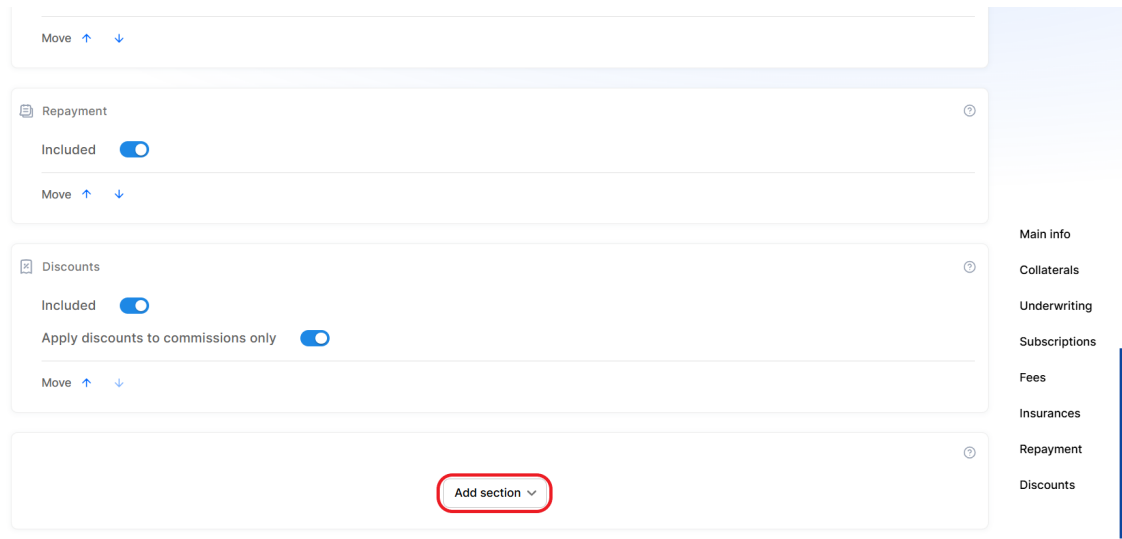
<sup>1</sup>Taking out a new loan to replace an existing loan, usually to obtain more favorable loan terms, such as a lower interest rate, extended repayment period, or different monthly payment amounts.

<sup>2</sup>Modifying the original terms and conditions of the loan to provide a more affordable repayment plan, typically when a borrower is facing financial difficulties or is struggling to meet their loan obligations. Can provide relief in the form of lower interest rates, extended repayment periods, deferred payments, changes in payment frequency, debt consolidation, waiving of penalties or fees, partial write-offs, etc.

<sup>3</sup>Automatically renew a deposit for the same tenure at the prevailing interest rate when it matures, without any action required from the account holder.

<sup>4</sup>Reinvestment of the principal (and sometimes accrued interest) of a matured financial product, such as a certificate of deposit (CD), into a new term product of the same type but potentially with different terms (like updated interest rates or maturity period).

<sup>5</sup>Individual who assumes joint responsibility for repaying a loan alongside the primary borrower. If the primary borrower defaults on the loan, the codebtor becomes liable for the remaining debt.



To reorder sections on the product template, use the move arrows on each section.



To delete sections from the product template, disable the **Included** toggle button. This removes the section from the template editor and adds it to the **Add section** drop-down (from where you can enable it again).



## IMPORTANT!

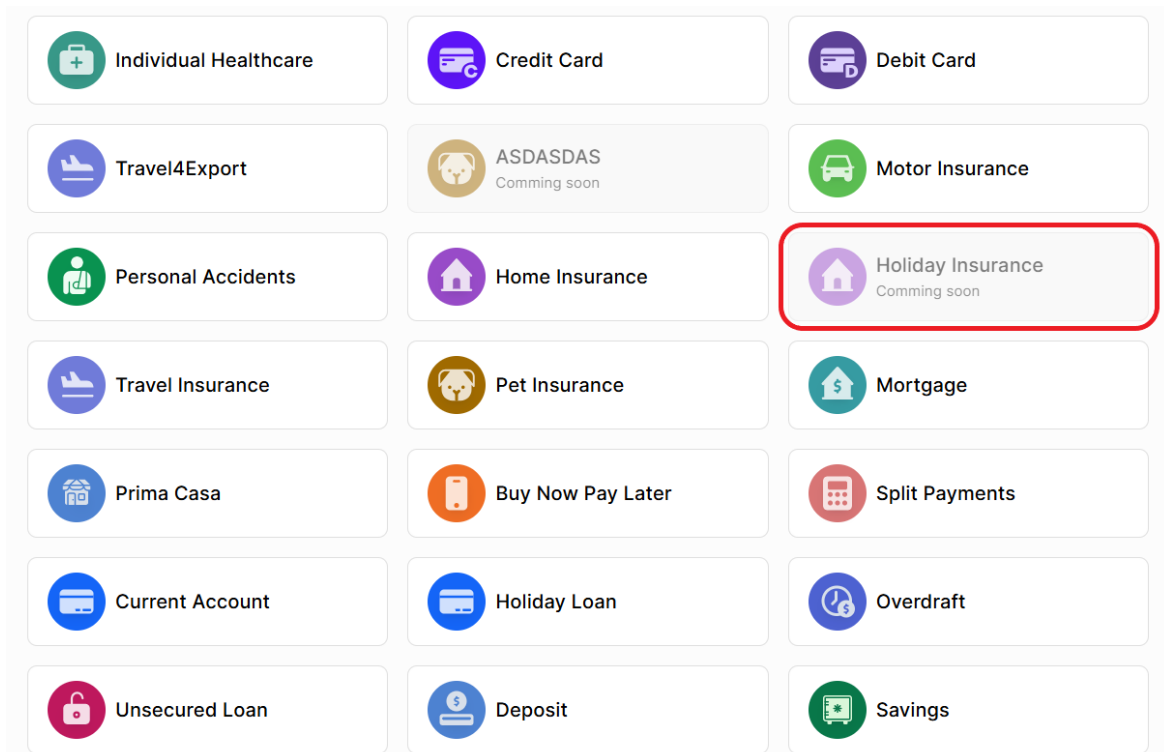
You cannot remove Interest, Fees, and Underwriting sections from banking product templates.

## Product Template Life Cycle

## Draft

When you create a new product template (by duplicating an existing template), it begins in the Draft state. This is the only state where the template can be edited and customized.

When selecting the template for a new product, draft templates are displayed in the selection window as grayed out, inactive buttons tagged as *Coming soon*.



## Active

A product template in the Active state is available in the Product Designer, allowing the creation of products based on it. To activate a template, click the ellipsis button (...) and select **Active**.

### IMPORTANT!

Once a product template is activated, it cannot revert to the "Draft" above state. Ensure all edits are complete before activation. To make further changes after activation, you will need to create a duplicate and work on a new template.

## Inactive

If you want to disable a product template (prevent its use in the Product Designer to create new products), you can deactivate it by clicking the ellipsis button (...) and selecting **Inactive**. Templates in the Inactive state can be reactivated by selecting the Active option again from the ellipsis menu.

# Product Analytics (Portfolio)

### NOTE

In order to populate the product analytics page with data, check the ["Product events" on page 843](#) section.

Product analytics provide key performance indicators that help with monitoring the banking or insurance products sales performance. The page displays all products that have a ["Product events" on page 843](#) published on their workflow.



#### Product Analytics

Product Name	Product Code	Currency	Product Type	New Customers	New Business	Total Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Prime Business	BP3_SME	GBP	Bank Account	79	58.22%	36,508,762
Start-Up Leap	BP1_SME	GBP	Bank Account	85	54.19%	34,153,259

Field	Description
Product Name	The name of the product.
Product Code	The code of the product.
Currency	The currency for the total value.
Product Type	The type of product.
New Customers	The number of signed contracts/issued policies by new customers.
Total Value	The total value of the signed contracts/issued policies.
New Customers	The number of signed contracts/issued policies by new customers.

Field	Description
New Business	The percentage of the value of signed contracts/issued policies by new customers out of the total contracts/issued policies value. For instance, if there are two contracts signed, with a total value of \$5,000, and only one of the contracts is signed by a new customer, but the value of this contract is \$4,500, the new business is 90%.

The KPI values are the total values calculated based on the product event date: if the product event is published at the current date, the product analytics list shows today's data only. For existing journeys, "[Add historic data to Product Analytics](#)" on page 845 to list the KPIs' total values starting from the past up until the current date.

## Product Analytics Details

Double click a product to open a dashboard for a more detailed overview of the product's performance. The product dashboard includes the following KPIs:

KPI	Description
Application In Progress	The number of product journeys that are in progress at the current date.
Conversion Rate	The percentage of signed contracts/issued policies from the total number of journeys initiated.
Rejection Rate	The percentage of rejected applications out of the total number of applications.
Contracts Signed	The number of signed contracts/issued policies.
Time to Yes	The average time between the journey start date and contract sign.
Total Value	The total value of the signed contracts/issued policies. Unlike in the product analytics list, here the KPI value is for the current month/day.
New Customers	The number of signed contracts/issued policies by new customers. Unlike in the product analytics list, here the KPI value is calculated for the current month/current day .
New Business	The percentage of the value of signed contracts/issued policies by new customers out of the total contracts/issued policies value. Unlike in the product analytics list, here the KPI value is calculated for the current month/day.

## Monthly view

For the monthly view, the KPIs are calculated for the current month.

The percentage reflects the difference between the KPI value calculated for the number of days that have passed in the current month and the value of the KPI calculated for the same number of days in the previous month.

For instance, if the current date is 14th March 2022, the percentage represents the comparison of the KPI values for 1-14 March 2022 and 1-14 February 2022.

The graphs on the page display the calculated KPI values for the last 12 months.

## Daily view

For the daily view, the KPIs are calculated for the current day.

The percentage reflects the difference between the KPI value calculated for the current day and the value of the KPI calculated for the previous day.

The graphs on the page display the calculated KPI values for the last 30 days.

## Product events

Product KPIs are calculated based on events that need to be raised on each banking/insurance product you want displayed in the Product Analytics page.

Event Name	Event Description	Impacted KPI	Raise event
JourneyStart	Journey event that occurs when a banking/insurance journey begins.	Application In Progress and Conversion Rate.	When the first business status change occurs.
JourneyAbandoned	Journey event that occurs when a banking/insurance journey expires.	Application In Progress.	When the journey business status changes after a period of inactivity.

Event Name	Event Description	Impacted KPI	Raise event
JourneyUnsuccess	Journey event that occurs when a banking/insurance journey is finished but no contract/policy is signed.	Application In Progress.	On the journey step or the business status change that indicates the customer has refused the contract/policy offer.
JourneySuccess	Journey event that occurs when a banking/insurance journey is finished and the contract/policy is signed.	New Customers, New Business, Time to Yes, Total Value, Conversion Rate, Contract Signed, and Application In Progress.	On the journey step or the business status change that indicates the contract/policy is signed or generated.
ProductApproved	Journey event that must be raised after an offer is submitted and after the customer application/ quote is approved.	Rejection Rate.	After the business status change that indicates the product is approved.
ProductRejected	Journey event that must be raised after the customer application/quote is rejected.	Rejection Rate.	After the business status change that indicates the product is rejected.

## Raise events

Use the [ftos.journey.publish](#) server SDK method to trigger the product events when a journey changes one business status to another, or on a journey step. In case your product is offered on more than one journey (for instance on both an assisted and a self-assisted journey), make sure the events are triggered on each journey.

## Raise events at business status change (recommended)

1. In the business workflow transition configuration, add or edit the [action groups](#) for the workflow.
2. In the **Action After** tab, insert a workflow transition action of the **Run Custom Script** type.
3. Use the Code section to raise the event using the [ftos.journey.publish](#) method.

## Raise events on the journey step

1. Create an [on-demand server automation script](#) for [ftos.journey.publish](#) method.
2. Call the script in the **AfterJs** section of the journey step using [ebs.callActionByName](#).

### NOTE

[GIGO](#) applies to the product analytics KPIs, as their accuracy depends on the rigor with which events data is populated. Therefore when configuring analytics, make sure there are no missing, fictitious, inaccurate or duplicate events.

## Add historic data to Product Analytics

For banking/insurance products that are currently running in production, there is the option to bring in historic data for the calculation of the product KPIs.

### Prerequisite

Add the Product Analytics Utility [security role](#) to your profile.

### 1 Populate events Excel template

### HINT

The product utility import template Excel file can be found by unpacking the [SysDigitalSolutionPackages](#).

1. Fill in the .xlsx template columns with the data you want to import in Product Analytics.

### NOTE

In order to ensure a successful import, make sure the columns are completed based on each event type, add the event date in YYYY-MM-DD format and all numeric values, apart for the contract value, as whole numbers. Do not alter the column order.

## 2 Upload historic data

2. In the **Product Factory** menu item, scroll down to **Product Configuration**. Double click **Analytics Utility**.

The analytics utility list opens.

Analytics Utility

Created On	Created by user	Operation	Status
01/11/2022 13:00	host	Upload	ERROR
01/11/2022 11:58	host	Upload	ERROR
01/11/2022 10:33	host	Upload	ERROR
31/10/2022 16:27	host	Upload	ERROR
31/10/2022 14:22	host	Delete	SUCCESS
31/10/2022 14:19	host	Upload	SUCCESS
31/10/2022 14:17	host	Upload	SUCCESS
31/10/2022 14:08	host	Upload	SUCCESS
31/10/2022 14:04	host	Upload	SUCCESS
31/10/2022 13:46	host	Delete	SUCCESS

5 10 20 1 2

Field	Description
Created on	The date when the operation is created.
Created by user	The user that inserted the operation.
Operation	The performed action. Can be either Upload or Delete.
Status	The status of the operation. Can be either Success or Error.

3. Select **Upload** from the dropdown.
4. Add the filled-in Excel file.
5. Click **Start import**.

The analytics utility page now displays your operation with an updated status: success or error.

6. Double click the operation to see more details on the import. In case the import has a success status, check the message field to see how many events were imported successfully.

Status	SUCCESS
Message	9 events added
Operation	Upload
File	events.xlsx

If the status is error, the toast error message is listed in the message field.

Status	ERROR
Message	JavaScriptException Status: 400 Event date should be a valid date lower than or equal with today!
Operation	Upload
File	events - err.xlsx

After upload, the product analytics metrics are calculated taking into account the product data from the past up until the current date.

## Delete imported data

1. On the Analytics Utility page, click **Insert** to add a new operation.
2. Select **Delete** from the dropdown. A list with all the products in product analytics appears.

Operation			
Delete			
Product Name	Product Code	Currency	Product Type
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Premium Payments	EXT4_SME	GBP	Bank Account
Prime Business	BP3_SME	GBP	Bank Account
Start-Up Leap	BP1_SME	GBP	Bank Account
Start delete			

3. Select the product for which you want to delete data.

4. Click **Start Delete**.

If the operation is successful, the product performance historic data is erased from the product analytics page.

## Product Life Cycle

You can use product approval and versioning to track product iterations through their various development and release stages. A product version can be in one of the following states:

- Draft - A work-in-progress version or candidate to become the approved product version
- Approved - The currently active version of the product
- Closed - A discontinued prior version

The approval and versioning process is illustrated below:

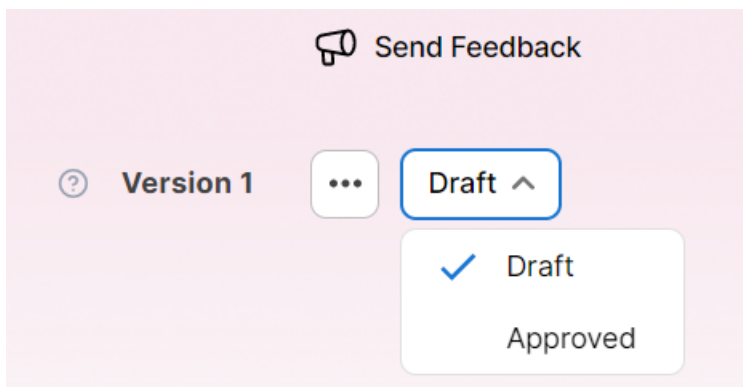
1. The product starts in a **Draft** state. This is the only state in which a product can be edited or deleted.
2. When the product design is finalized and validated, you can promote it to the **Approved** state, making it the active version of the product.
3. To update an approved product, you need to create a new version based on it. The new product version starts out in the **Draft** state (while the previous version remains in the **Approved** state).
4. Once the development of the new product version is finalized and validated, you can promote it to the **Approved** state (this also causes the previously approved version to

go into the **Closed** state).

5. Repeat the process from step 3 whenever you want to create a new product version.

## Approve a Product

All your product configurations are saved automatically, so you don't need to worry about losing any of your progress. When configuring your product, your settings are automatically saved in *Draft* status. To activate the product, change its status from **Draft** to **Approved** using the drop-down.



Your product will be digital journey-ready once it has been approved, allowing you to incorporate it into digital journeys and expose it to potential customers through digital touchpoints, channels, or portals.

### IMPORTANT!

Once approved, no additional changes are possible without creating a new version of the product. Also, you will no longer be able to change the product code (even if you create a new version).

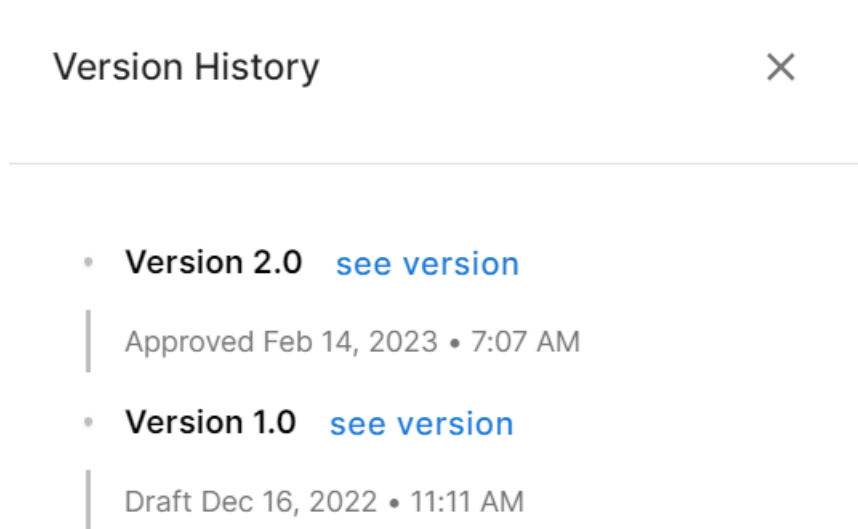
## Approved Products Summary

When selecting an approved product from the list of products, the offer view opens by default (instead of the product editor which is displayed by default for draft products). To display a summary of the product settings, click the **Product View** button at the top right corner of the page.

## Create a New Product Version

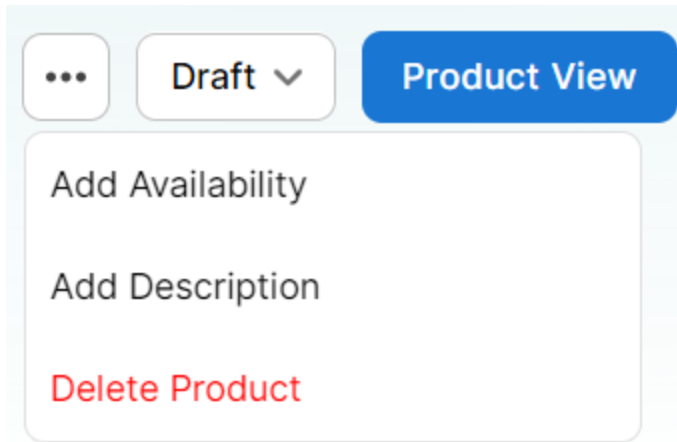
To create a new product version, on an approved product, click the **Create New Version** at the top right corner of the screen.

You can check different versions of your product by clicking the **Version** number in the top-right corner of the screen. The version you select opens in the same browser window.



## Delete a Product Draft

When editing a product in the draft state, the ellipsis menu (...) includes an option to delete the draft.



Any previously approved product versions will not be deleted.

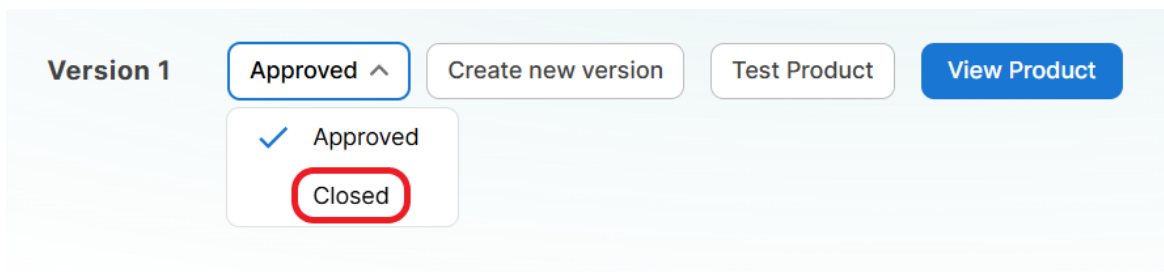
## Close a Product

If you wish to deactivate a product, so that it can no longer be incorporated into digital journeys or exposed through digital touchpoints, channels, or portals, you can close it.

### IMPORTANT!

- Only approved products can be closed.
- A product can be closed only if it has no active offers.
- Closed products cannot be exported/imported.

To close an approved product, select the **Closed** option from the product status drop box at the top of the screen.



**HINT**

You can still version a closed product.

## Offers

Offers allow you to decide how to sell your products (either individually or by bundling multiple products together), create user-friendly descriptions, target specific audiences, or provide additional discounts. With offers, you can make changes to products (commercially) with autonomy and agility.

### Create Offers

The instructions below explain how to configure each offer component. You can customize your offers according to your business requirements, or pair a distinctive combination of financial products into a bundle.

**IMPORTANT!**

To create an offer, you must first define and approve at least one product in Product Designer.

To create an offer:

1. In FintechOS Studio, go to **Main Menu > Products**
2. Select an approved product from your **Product list**. This will be the offer's main product.
3. Click **+New Offer**.
4. To set the offer's name and availability period, click and type over the corresponding fields at the top left of the screen.

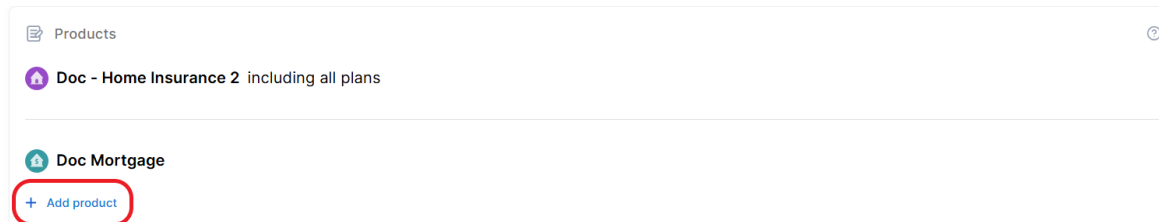
## Products

When you create an offer, the offer's main product is automatically added to the Products section. This primary product is a mandatory component of the offer and cannot be removed.

### Product Bundles

To provide comprehensive financial solutions, you can enhance the main product with additional products that you offer together as a bundle. For example, you may include a home insurance alongside a mortgage loan main product to improve customer convenience and streamline the purchasing process.

To bundle additional products in the same offer, click **+Add product** in the offer's **Products** section.



### NOTE

You can bundle additional insurance products in the same offer only if they have single pricing plans.

## Content

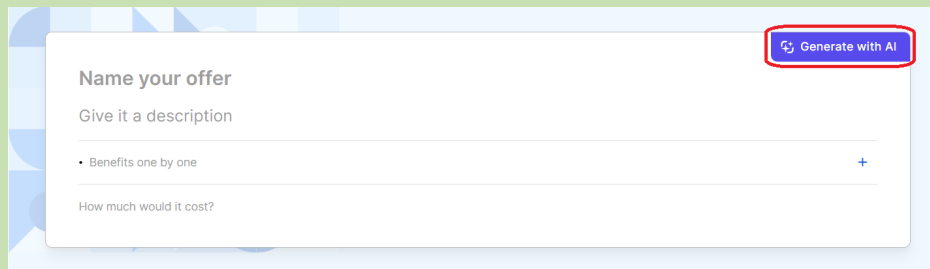
The offer content includes a distinctive name, a comprehensive description, and appealing benefits designed to attract customers. The content cards determine how the offer will appear in your customers' digital journeys.

- **Name:** The distinctive title of your offer, used to efficiently identify and market your products.
- **Description:** An optional summary outlining the content, features, and purpose of your offer. This helps customers understand what is included and how it aligns with their needs.

- **Benefits:** The advantages and perks associated with your offers, such as potential cost savings, convenience, or added value, designed to enhance the customer experience and meet specific preferences and goals.

## HINT

You can pre-populate the Name, Description, and Benefits fields by clicking **Generate with AI**. This functionality draws data from your offer's product (s) and generates a comprehensive introduction for your offer.

A screenshot of a form titled "Name your offer" with a light blue background. The form has four input fields: "Give it a description", "Benefits one by one" (with a plus icon), and "How much would it cost?". A red box highlights the "Generate with AI" button in the top right corner of the form.

To activate AI generation, you must enable the *sys-pf-enable-ai* system parameter (see ["System Parameters" on page 1285](#) for details).

## IMPORTANT!

Offer AI Content Generator uses OpenAI's generative artificial intelligence to allow your users to create content for offers, based on the details of your products and services offering. This AI feature is optional and enabling it may offer the functionalities intended for any user across your organization. By enabling the Offer AI Content Generator feature, you agree to the [Supplemental Terms of Service](#).

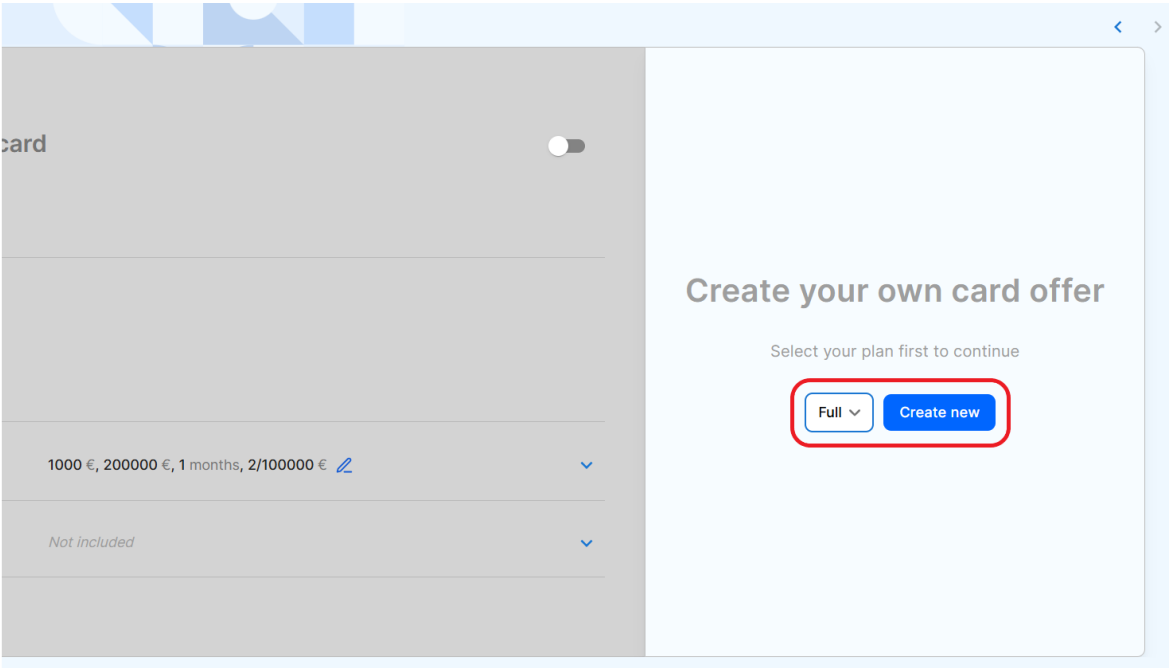
- **Plans:** If an offer includes insurance products, the insurance plans configured at product level will automatically appear in the offer's content. See the "[Insurance Plans](#)" [below](#) section below for information on how to configure plans in your offer.
- **How much would it cost:** This section allows you to include additional information that could be relevant, such as pricing.

## Insurance Plans

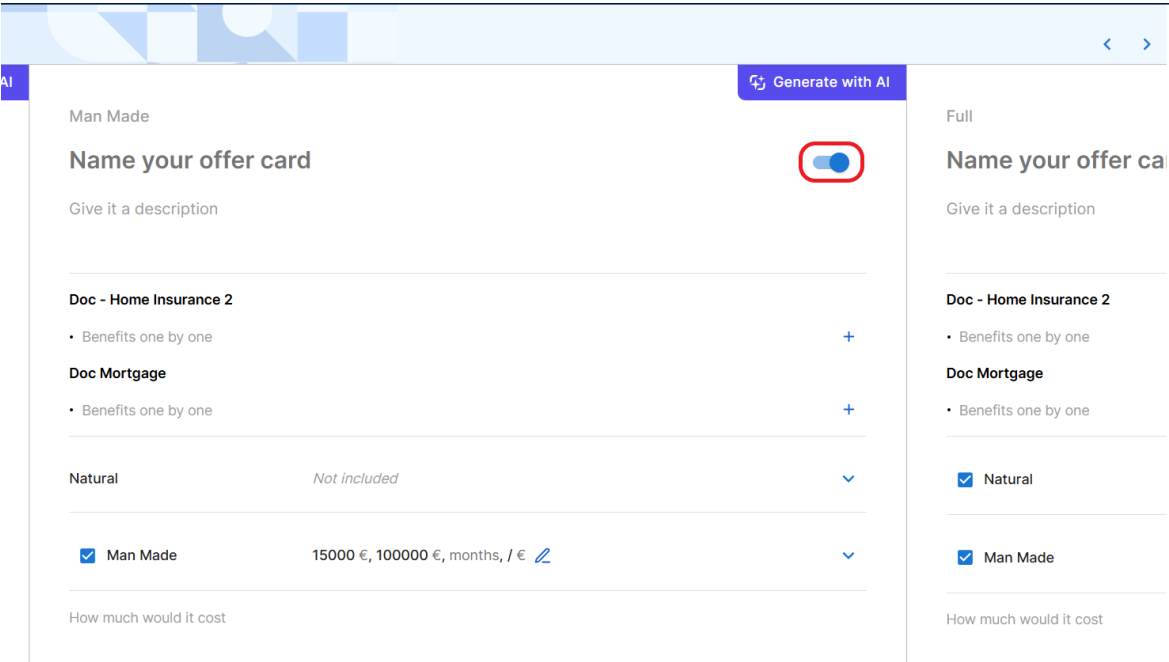
If your offer includes insurance products, you can configure the coverage choices that you want to appear on the content card(s). You can set up multiple cards, one for every coverage variation you wish to provide, each with its own name, description, benefits, plan, and pricing information.

By default, a content card is created for each insurance plan of the main product. (use the < > buttons to navigate between cards).

If your offer includes an insurance product with multiple plans, you can manually create additional cards. To create a new card, go to the last available card slot, select an insurance plan as template, and click **Create New**.



To enable or disable a card on the offer, use the toggle at the top right corner of the card. This allows you to create targeted offers with specific plans.



You can configure coverage/subcoverage options using controls similar to those in the plans interface:

- Use the checkmarks to enable or disable coverages and subcoverages.

### NOTE

You cannot enable a coverage that was not included in the card's template plan.

- Use the pencil icons to customize the claim rules (the **deductible**<sup>1</sup>, **indemnity limit**<sup>2</sup>, **waiting period**<sup>3</sup>, or event limits), overriding the values set at the plan or product level.
- Use the **preselected** checkbox to mark the optional coverages or subcoverages that should be enabled by default when a user opens the card.

<input checked="" type="checkbox"/> Natural	1000 €, 200000 €, 1 months, 2/100000 €	<a href="#">🔗</a>	<a href="#">^</a>
<input checked="" type="checkbox"/> Water Damage	€, €, months, / €	<a href="#">🔗</a>	
<input type="checkbox"/> Other	optional €, €, months, / €	<a href="#">🔗</a>	
<input type="checkbox"/> preselected			
<input checked="" type="checkbox"/> Man Made	15000 €, 100000 €, months, / €	<a href="#">🔗</a>	<a href="#">^</a>
<input type="checkbox"/> Unintentional	€, €, months, / €	<a href="#">🔗</a>	
<input checked="" type="checkbox"/> Intentional	optional €, €, months, / €	<a href="#">🔗</a>	
<input checked="" type="checkbox"/> preselected			

<sup>1</sup>Amount of money that the insured party must pay before their insurance policy starts paying for covered expenses.

<sup>2</sup>Maximum amount of the insurer's liability for any claim or claims under a policy.

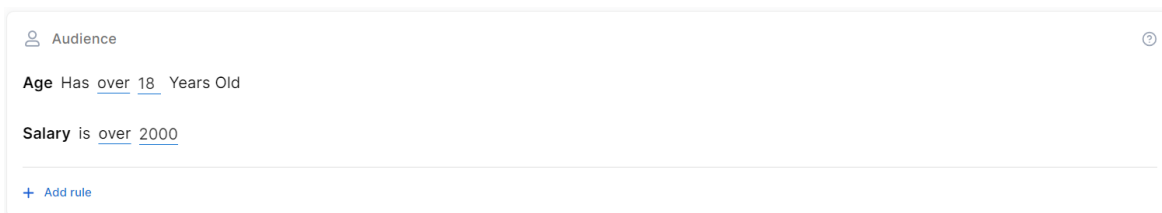
<sup>3</sup>Amount of time an insured party must wait before some or all of their coverage comes into effect.

## Audience

A target audience for your commercial offering refers to the specific group of consumers that you aim to reach with your advertising efforts.

Click **Add rule** and select one or more attributes that are directly linked to the product for which you are creating the offer. The attributes function as a filter mechanism to help you efficiently focus on a particular segment of customers.

For example, you can divide the target audience into groups based on demographic factors like age and income. This makes your offer only available to people 18 years old or older, and earning at least EUR 2,000 a month.



Audience

Age Has over 18 Years Old

Salary is over 2000

+ Add rule

## Discounts

The Discounts section allows defining discounts over the pricing of the underlying product, in part or for all the pricing. There is one sub-section per product (in the case of bundles). Including discounts in your offers is a great method for maintaining market competitiveness while attracting new clients and addressing the needs of existing ones.

While applying a discount to your offer, you have three options to choose from:

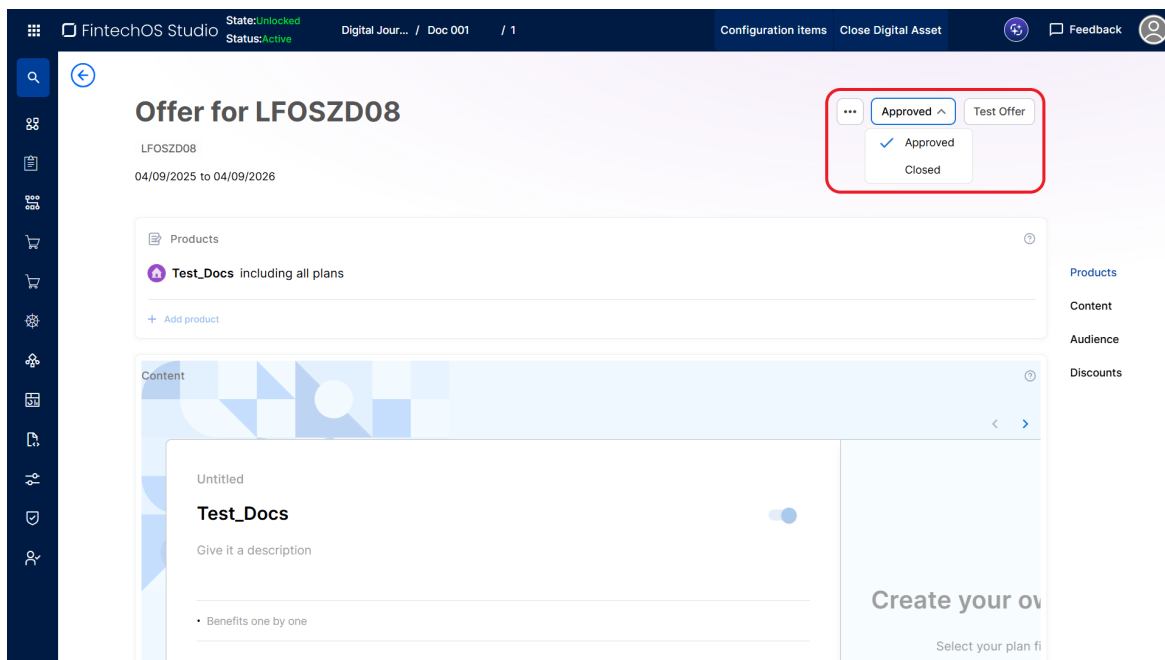
- **Accumulate with product:** Choose this option to combine the discounts assigned during product configuration with the discounts you create in this section.
- **Override the product:** Choose this option to override the discounts initially set during the product configuration for which you create the offer, allowing you to define a new discount.
- **Apply highest discount:** Choose this option to compare and apply the highest discount assigned during the product configuration process for which you create the offer, to the one you are configuring in this section.

## Offer Life Cycle

An offer progresses through a defined sequence of states during its life cycle. Each state indicates the current maturity and availability of the offer for use within digital journeys and customer-facing channels:

1. **Draft** - The initial state of an offer. It represents a work-in-progress or a candidate for approval. This is the only state in which an offer can be edited.
2. **Approved** - Once the offer design is finalized and validated, it can be promoted to the Approved state. At this stage, the offer can be incorporated into digital journeys and exposed to potential customers through digital touchpoints, channels, or portals. Once an offer is approved, no further modifications are allowed. To create a variant, you need to create a duplicate (see "[Duplicate or Delete Offers](#)" on the next page).
3. **Closed** - A discontinued offer, unavailable for further use.

To change an offer's state, select the desired option from the drop-down at the top right corner of the screen.



## Duplicate or Delete Offers

In the offer screen, use the ellipsis button (...) to select one of the following options:

- **Duplicate** - This creates a new offer with the same settings. The name of the duplicate is the original name preceded by the wording *Copy of*. For instance, a duplicate of the *Special Promotion* offer will be called *Copy of Special Promotion*.
- **Delete offer** - Deletes the offer.

## Test Products and Offers

You can run simulations for your [approved](#) products and offers using various inputs to evaluate how your eligibility, underwriting, and pricing rules perform under different scenarios.



## Banking product simulation

Type

Products

Item

My Unsecured Loan TLU274

General

Amount 1,000 EUR from 1,000 EUR to 180,000 EUR

Period 12 months from 12 months to 180 months

Interest Interest

Payment Schedule Equal Installment Monthly ...

Eligibility

Age 25

Type of Employment Temporary Employment

Check Eligibility

Name	Result
Age	Approved
Type of Employment	Rejected

Underwriting

Age 25

Type of Employment Temporary Employment

Check Underwriting

Name	Type	Result
Age	Eligibility	Approved
Type of Employment	Eligibility	Rejected

Pricing

Term 12

Credit Amount 5000

PaidAmount 1000

Age 25

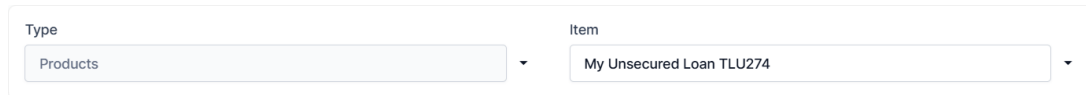
Pricing Simulation

	APR (%)	First Installment (EUR)	Interest (%)	Total Repayment (EUR)
	11.37	86.9	6	1058.26

Commission Type	Fee	Periodicity	Value (EUR)	Percentage (%)
Administration Fee	Administration Fee	Monthly	0	1% over RemainingValue
Front-end Fee	Front-end Fee	Once	20	2% over Amount

No.	Due Date	No of days	Interest (EUR)	Remaining Value (EUR)	Principal (EUR)	Total Installment (EUR)	Management F
1	18/02/2025	30	5.00	1,000.00	81.07	86.90	0.
2	18/03/2025	30	4.59	918.93	81.48	86.84	0.
3	18/04/2025	30	4.19	837.45	81.88	86.77	0.
4	18/05/2025	30	3.78	755.57	82.29	86.70	0.
5	18/06/2025	30	3.37	673.28	82.70	86.63	0.
6	18/07/2025	30	2.95	590.58	83.12	86.56	0.
7	18/08/2025	30	2.54	507.46	83.53	86.49	0.
8	18/09/2025	30	2.12	423.93	83.95	86.42	0.
9	18/10/2025	30	1.70	339.98	84.37	86.35	0.
10	18/11/2025	30	1.28	255.61	84.79	86.28	0.
11	18/12/2025	30	0.85	170.81	85.22	86.21	0.
12	18/01/2026	31	0.44	85.60	85.60	86.11	0.

1. In FintechOS Studio, go to **Main Menu > Products**, and select an approved product or offer from the list.
2. Click the **Test Product** button at the top right corner of the screen to open a simulation page for the selected product.
3. To test a different product or an offer, use the **Type** drop-down menu to switch between products and offers, and the **Item** drop-down menu to select the respective product or offer.



The screenshot shows two dropdown menus. The first menu, labeled 'Type', has 'Products' selected. The second menu, labeled 'Item', has 'My Unsecured Loan TLU274' selected.

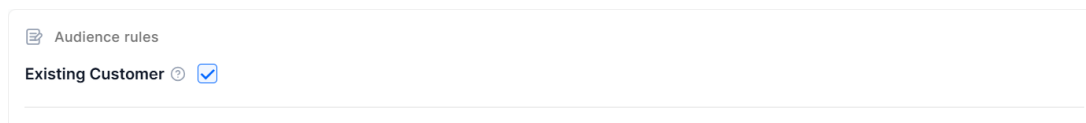
4. In the **General** section, provide values that you wish to simulate for the various inputs in the product configuration.



The screenshot shows the 'General' section of the product configuration. It includes the following fields:

- Amount**: 1,000 EUR (range: from 1,000 EUR to 180,000 EUR)
- Period**: 12 months (range: from 12 months to 180 months)
- Interest**: Interest (dropdown menu)
- Payment Schedule**: Equal Installment Monthly ... (dropdown menu)

5. If testing an offer, define the audience settings that you wish to simulate in the **Audience rules** section.



The screenshot shows the 'Audience rules' section. It includes the following field:

- Existing Customer**: ☒ (checkbox)

6. To check eligibility, enter the eligibility rules inputs in the **Eligibility** section and click **Check Eligibility**.

Eligibility

Age ⓘ 25

Type of Employment ⓘ Temporary Employment ▼

Check Eligibility

Name	Result
Age	Approved
Type of Employment	Rejected

7. To check underwriting, enter the underwriting rules inputs in the **Underwriting** section and click **Check Underwriting**.

Underwriting

Age ⓘ 25

Type of Employment ⓘ Temporary Employment ▼

Check Underwriting

Name	Type	Result
Age	Eligibility	Approved
Type of Employment	Eligibility	Rejected

8. To assess pricing, enter the pricing rules inputs in the **Pricing** section and click **Simulate**

**Pricing.**

Pricing

Term

12

Credit Amount

5000

PaidAmount

1000

Age

25

Pricing Simulation

APR (%)	First Installment (EUR)	Interest (%)	Total Repayment (EUR)
11.37	86.9	6	1058.26

Commission Type	Fee	Periodicity	Value (EUR)	Percentage (%)
Administration Fee	Administration Fee	Monthly	0	1% over RemainingValue
Front-end Fee	Front-end Fee	Once	20	2% over Amount

No.	Due Date	No of days	Interest (EUR)	Remaining Value (EUR)	Principal (EUR)	Total Installment (EUR)	Management F
1	18/02/2025	30	5.00	1,000.00	81.07	86.90	0.
2	18/03/2025	30	4.59	918.93	81.48	86.84	0.
3	18/04/2025	30	4.19	837.45	81.88	86.77	0.
4	18/05/2025	30	3.78	755.57	82.29	86.70	0.
5	18/06/2025	30	3.37	673.28	82.70	86.63	0.
6	18/07/2025	30	2.95	590.58	83.12	86.56	0.
7	18/08/2025	30	2.54	507.46	83.53	86.49	0.
8	18/09/2025	30	2.12	423.93	83.95	86.42	0.
9	18/10/2025	30	1.70	339.98	84.37	86.35	0.
10	18/11/2025	30	1.28	255.61	84.79	86.28	0.
11	18/12/2025	30	0.85	170.81	85.22	86.21	0.
12	18/01/2026	31	0.44	85.60	85.60	86.11	0.

# Test Product APIs Locally

To test the "Products APIs" on page 904 for a specific product:

1. In FintechOS Studio, go to **Main Menu > Products**, and select an approved product from the list.

2. Click the **Test Product** button at the top right corner of the screen to open a simulation page for the selected product.
3. Click **Download Postman Collection**.

This downloads a JSON file containing the [Postman](#) collection for the product's APIs, allowing you to test API endpoints locally or share them with your team. The collection contains requests for:

- Product metadata
- Product eligibility, underwriting, and pricing
- Offer eligibility, underwriting, and pricing for all related offers

## Product Formulas

Product formulas process relevant product origination inputs (such as income, age, assets, risk class, etc.) to calculate desired outputs (such as credit scores, interests, discounts, etc.).

Product formulas are linked to their parent products, enabling: automatic association with the correct product during formula import into the target environment, synchronized state management between formulas and their parent products, and automatic deletion of formulas when the associated product or product version is removed.

## Basic Product Formulas

Product formulas allow you to create product-centric computations using a simple sentence-based user interface.

To write a product formula

1. Enter the formula expression.
2. Enter a formula name.

3. Specify the source of each attribute you used in the formula expression expression (either an input, data set, or another formula):

- **input** - This is a "Lexicon Term" on page 897 value provided in the product origination journey at run-time. The input value must be set up in the journey through user input, service integration, or data query.  
There is an extensive dictionary of predefined lexicon terms available for each product template that you can choose from (Loan amount, Base Rate, Total Floor Area, etc). Alternatively, you can create new ones or reuse existing lexicon terms in the context of new product templates straight from the formula editor.
- **data set** - At their most basic level, "Product Data Sets" on page 877 map product origination inputs to secondary values (e.g.: map a driver's experience input to a risk coefficient output). More advanced data sets can use formulas or other data sets as inputs, creating multi-step path from the product origination journey inputs to the data set output.
- **formula** - Just like data sets, you can use another formula as a source for a formula attribute.

## HINT

Both formulas and data sets are used to derive secondary values from inputs.

- **Formulas** are useful when you can express this relationship through a mathematical computation, e.g.:  
`risk coefficient = engine displacement * 0.002.`
- **Data sets** are useful when you need to define the relationship explicitly, e.g.:

```
risk coefficient = (3 if engine displacement <= 1500) or
(4.5 if engine displacement > 1500).
```

**IMPORTANT!**

Follow these formatting rules for formulas:

1. Attribute names cannot have spaces;
2. Use simple mathematical operations (+, -, \*, /, <, <=, >, >=).

## Interest Formula Example

**Interest = (FixedInterest + InterestByCustomerRating) \*  
MerchantPortfolioYield**

All three formula arguments are defined as data set types.

The customer's rating adds to a fixed interest rate to determine your product's interest rate. It also considers the historical yield your financial institution had with a certain type of merchant who sold the product to customers.

Sample data for the FixedInterest, InterestByCustomerRating, and MerchantPortfolioYield data sets are provided below:

**FixedInterest data set:**

	Loan Term (months)				
Loan Amount		<5000	5000- 9999	10000- 19999	>20000
	<6	20	15	12	10
	6 - 12	15	12	10	8
	12-24	12	10	8	6
	24>	10	8	6	5

**InterestByCustomerRating data set:**

Customer Rating	A2	B2	B3	C3
Interest	3	3.5	3.7	4.5

**MerchantPortfolioYield data set:**

Merchant Type	Merchant Gold	Merchant Standard
Yield	1	1.1

Assume your customer has a B2 credit rating and needs a \$10,000 loan for three years (36 months). Considering that the merchant type is gold, the system will calculate the interest by extracting values from the data sets and entering them into the formula using these variables. This is the result:  **$(6 * 3.5) * 1 = 21$**

The screenshot shows a configuration window for a variable named "Interest". It states "Interest calculations are" and "Interest is based on formula". The formula entered is  $(FixedInterest + InterestByCustomerRating) * MerchantPortfolioYield$ . Below the formula, three variables are listed with their definitions: "FixedInterest" is defined from "input", "InterestByCustomerRating" is defined from "input", and "MerchantPortfolioYield" is defined from "input". Each variable has buttons for "input", "formula", and "data set".

## Discount Formula Example

**$Credit\ Amount * (1 - (Income\ Level\ Factor * Loan\text{-}to\text{-}Value))$**

You calculate the discount based on two factors:

**Income Level Factor:** This factor represents the borrower's income level since you want a higher income level factor to indicate a stronger income profile.

**Loan-to-Value Ratio Factor:** This factor represents the loan amount's ratio to the collateral's appraised value. A lower loan-to-value ratio factor indicates a lower risk for the lender.

Sample data for the Income Level Factor data set is provided below:

	Income Level Factor			
Annual Income		<5000	5000- 9999	>10000
	A	0.01	1.0	1.2
	B	0.08	1.1	1.2
	C	0.07	1.0	1.1

Assume a B income level factor customer with a \$7,000 annual income applies for a mortgage to buy a \$300,000 home. They have saved \$60,000 for a down payment and require a \$240,000 loan.

First, the Loan-to-Value Ratio Factor would be calculated as follows:

Loan-to-Value Ratio Factor = Credit Amount / Property Value

Loan-to-Value Ratio Factor = \$240,000 / \$300,000

Loan-to-Value Ratio Factor = 0.8

Then, you must extract the value for your customer from the data set and replace the arguments in your formula. This is the result:

$$240000 * (1 - (1.2 * 0.8)) = 9600$$

The final amount the customer needs to pay is the initial amount, minus the calculated discount. In this example, if the loan amount was \$300,000 and the discount was \$9,600, the customer would need to repay \$290,400.

## Pricing Formula Example

You want to calculate the premium for a property insurance policy. The premium amount is determined based on the Building sum insured, Building flexa coefficient, Building NatCat coefficient, and Payment frequency coefficient. These coefficients are obtained from data sets that account for various factors, such as property characteristics, location, and payment preferences.

Let's walk through an example to demonstrate how the next formula works:

**InsuredAmount \* 0,00048 \* Building Flexa coefficient \* Building Natcat coefficient \* Payment Frequency Coefficient**

Sample data for the needed data sets is provided below:

**Building Flexa coefficient data set:**

Property Type	Residential	Commercial	Industrial
Coefficient	1	1.1	1.2

**Building NatCat coefficient data set:**

Property Type	Residential	Commercial	Industrial
Coefficient	0.8	0.9	0.95

**Payment frequency data set:**

Payment Frequency	Monthly	Quarterly	Annual
Coefficient	1.1	1	0.95

We are insuring a building situated in an industrial area and the premium will be paid in one installment (annual). Assume these details for the specific insurance policy; the coefficients will have the values below:

Building sum insured: \$200,000

Building flexa coefficient: 1.2

Building NatCat coefficient: 0.9

Payment frequency coefficient: 0.95

You will use these variables to compute the pricing by inserting them into the formula. This is the result:

$$200,000 * 0.00048 * 1.2 * 0.9 * 0.95 = 98.496$$

Based on this scenario, you will charge a premium of \$98.496 for the specified property insurance policy. The formula incorporates the relevant coefficients from the data sets to price the insurance coverage.

## Underwriting Formula Examples (Banking and Insurance)

### Banking

#### **CurrentDTI < MaxDTI**

Let's assume you want to use the Debt-to-Income (DTI) ratio as one of the criteria for assessing a new loan's eligibility. The DTI ratio measures the borrowers' ability to manage additional debt based on their current financial obligations and income.

The current DTI initial formula argument will be based on another formula, which is the following:

$$\text{CurrentDTI} = (\text{ExistingInstallmentsAmount} + \text{LoanInstallmentAmount}) / \text{Income}$$

Let's consider the following details for a customer loan application:

- Existing monthly installments (such as credit card payments, car loans, etc.): \$500
- Loan installment for the new loan: \$200
- Customer monthly income: \$2,000

This is the current DTI formula result:  $(500 + 200) / 2000 = 0.35$

Consider the data set for the MaxDTI formula argument provided below, resulting in a value of 0.4 based on the customer's income.

Income	<4999	5000–10000	>10001
MaxDTI ratio	0.3	0.4	0.5

Next, the formula will evaluate the initial underwriting rule by replacing the formula arguments with values. This is the result: **0.35 < 0.4**.

Since your customer's current DTI of 0.35 is less than the maximum DTI ratio of 0.4, the customer meets the eligibility criteria for your product.

## Insurance

As part of the underwriting process, you must assess an insurance application to determine eligibility.

You consider the risk factor allocated to the applicant based on their insurance history, the requested coverage level, the length of the policy's term, and the premium rate.

The risk factor and premium rate are data set formula arguments, whereas the coverage amount and premium rate are defined as input type.

Let's walk through an example to demonstrate how the next formula works:

$$\frac{(\text{RiskFactor} * \text{CoverageAmount})}{(\text{PolicyDuration} * \text{PremiumRate})} > 1.5$$

Below is sample data for the risk factor and premium rate data sets:

**Risk Factor data set:**

Risk Factor	A1	A2	B1	B2
Value	0.9	0.8	0.7	0.1

**Premium rate data set:**

	Health Condition				
Age Group		18-25	26-35	36-45	<46
	Excellent	0.05	0.04	0.03	0.02
	Good	0.05	0.04	0.03	0.03
	Fair	0.02	0.01	0.01	0.01

Assume your A2 risk factor customer wants coverage of \$100,000 for the duration of one year. This is the result:

$$(0.8 * \$100,000) / (1 * 0.03) > 1.5$$

$$19.2 > 1.5$$

Based on this result, you can offer the insurance policy with the desired coverage while meeting your guidelines.

## Advanced Product Formulas

For complex calculations, you can use the ["Formula Editor" on page 469](#) to define product formulas that go beyond the basic expressions available in product templates. This allows you to take advantage of the advanced ["Business Formulas" on page 450](#) capabilities, such as multi-step logic, built-in functions, recursion, and formula testing across various scenarios.

### Convert a Basic Product Formula into an Advanced Product Formula

You can use a basic product formula as a starting point for an advanced product formula. This will automatically convert your formula expression to the advanced format and create the corresponding input parameters and data sets.

#### **IMPORTANT!**

Once you edit a formula in the advanced interface, its original expression is no longer visible or editable within the product template.

To convert and open a basic formula in the advanced editor:

1. In the product template, locate the desired formula field.
2. Click the **fx** drop-down button on the left side of the formula field.
3. Select the name of the formula. This will replace the formula expression with the formula name in the formula field.
4. Click the **Edit** button at the right side of the formula field.

This opens the formula in the ["Formula Editor" on page 469](#) where you can further edit it. You can access and further customize the ["Formula Inputs" on page 451](#) and ["Data Sets" on page 513](#) from the FintechOS Studio Business Formulas submenu.

## Create an Advanced Product Formula from Scratch

To create an advanced product formula:

1. In the product template, locate the desired formula field.
2. Click the **fx** drop-down button on the left side of the formula field.
3. Click **+ Create Formula**.
4. Select **New Formula**.
5. Enter a name for the formula.
6. Click **Create Formula**.

This opens the formula in the ["Formula Editor" on page 469](#) where you can further edit it. Use the ["Formula Inputs" on page 451](#) and ["Data Sets" on page 513](#) to define the arguments required by your formula.

## Reuse a Product Formula Within the Same Product

A product formula can be reused across multiple components of the same product. For example, two different insurance coverages may share the same pricing calculation logic.

### **IMPORTANT!**

Editing a reused formula affects all its instances across the product.

To reuse a formula from the same product:

1. In the product template, locate the desired formula field.
2. Click the **fx** drop-down button on the left side of the formula field.
3. Select an existing formula from the drop-down menu. This list includes all formulas defined in the current product.

## Clone a Product Formula

If you wish to use one of the existing ["Business Formulas" on page 450](#) in your products, you can clone it. This creates a new, independent copy of the selected formula, allowing you to build formula templates that you can customize for various products.

### IMPORTANT!

The cloned business formula must be in the **Active** status.

To clone a business formula:

1. In the product template, locate the desired formula field.
2. Click the **fx** drop-down button on the left side of the formula field.
3. Click **+ Create Formula**.
4. Select **Clone Existing**.
5. Enter a new **Formula Name**.
6. Use the **Select Formula to Clone** field to choose your source formula. This list includes formulas from both the current and other products.
7. Click **Clone Formula**.

This opens the formula in the ["Formula Editor" on page 469](#) where you can further edit it. Use the ["Formula Inputs" on page 451](#) and ["Data Sets" on page 513](#) to configure the arguments used by your formula.

## Product Data Sets

Product data sets allow you to define evaluations or custom attributes for your ["Product Formulas" on page 865](#) based on mappings to ["Lexicon Term" on page 897](#) attribute values.

## One-Dimensional Data Sets

You may need, for instance, to define a *risk coefficient* attribute for a "Motor Insurance" on page 657 underwriting formula. The risk coefficient is based on the driver's experience. You can map the specific values of the *driving experience* lexicon term (e.g.: under 2 years, 2 to 5 years, and over 5 years) to the values of a custom attribute representing the *risk coefficient* (e.g.: 1.5, 1.25, and 1 respectively).

### Formula ?

*fx* ▾ Miles per year \* Risk coefficient < 100000

↳ Miles per year is defined from input

↳ Risk coefficient is defined from data set ?

<2	2;5	>5
1.5	1.25	1

☐ 1st row is Driving Experience ▾ ✓ ✕


used in eligibility


## Two-Dimensional Data Sets



You can also create two-dimensional data sets that map pairs of input values from two different lexicon terms to an output value found at the intersection of the two inputs. For instance, based on the driver's experience combined with the number of miles driven per year, you may decide if a motor insurance application is accepted, rejected, or deferred for manual analysis.

Data set 

	<2	2;5	>5
<50000	Manual Analysis	Passed	Passed
50000;100000	Not Passed	Manual Analysis	Passed
>100000	Not Passed	Not Passed	Manual Analysis

☐ 1st column is Miles per year 


☐ 1st row is Driving Experience 

[used in eligibility](#)  

## Multi-Dimensional Data Sets

If you need to create multi-dimensional mappings, you have two options:

- You can cascade two-dimensional data sets by providing a data set as a source for another data set's attribute. This way, you combine composite attributes to expand the total number of dimensions that go into the mapping.

**Data set** [First Owner vs Engine Type vs Car Condition](#) 

↳ First Owner is defined from input

↳ Engine Type is defined from input

↳ Car Condition is defined from

[used in eligibility](#)

- You can paste Excel grids with multiple header rows or header columns that match your source attributes into the data set field. The grid can include merged cells, so that

the data is easy to read.

Data set [?](#)

		Excellent	Good	Fair	Bad
yes	Diesel	Passed	Passed	Passed	Manual Analysis
	Gasoline	Passed	Passed	Manual Analysis	Manual Analysis
	Electric	Passed	Manual Analysis	Manual Analysis	Not Passed
no	Diesel	Passed	Manual Analysis	Manual Analysis	Not Passed
	Gasoline	Manual Analysis	Manual Analysis	Not Passed	Not Passed
	Electric	Manual Analysis	Not Passed	Not Passed	Not Passed

1st column is [First Owner](#) ▾

1st row is [Car Condition](#) ▾

2nd column is [Engine Type](#) ▾

↳ [First Owner](#) is defined from [input](#)

↳ [Engine Type](#) is defined from [input](#)

↳ [Car Condition](#) is defined from [input](#)

[used in eligibility](#)

You can also mix the two methods, e.g. multi-dimensional data sets with source attributes based on other multi-dimensional data sets, etc.

## How to Fill In a Product Data Set

Populate a data set by copying and pasting a table from Excel or Word.

### IMPORTANT!

Follow these formatting rules for data sets:

1. Don't include column or row names in the Excel source, only the values. You select the attributes that represent the header rows/columns in the Product Designer drop-downs.
2. For two-dimensional or multi-dimensional data sets, the cell(s) at the top left corner of the grid must be empty. They are used to determine the header rows/columns for the source attributes.
3. Use the <, >, <=, >=, or ; relational operators to define numeric intervals. The intervals are considered closed, so don't use brackets when defining them.

The dimensions used in data sets can be numeric, numeric intervals, or text, depending on the data type of the lexicon term. The data set output values can be numeric, text, or boolean.

## Product Settings

Product settings allow you to configure the option sets available in your "[Product Templates](#)" on [page 526](#) for selecting product features like payment schedules, types of collateral, insurance perils, types of fees, interest indexes, etc.

## Payment Schedule Type

A payment schedule represents the complete table of periodic loan payments, showing the amount of principal, interest, and other payments that comprise each installment until the loan is paid off at the end of its term.

The payment schedules are a quintessential part of a loan. Below you can see how to configure the installments, dates and calculations.

## Adding Payment Schedule Types

In FintechOS Studio, payment schedule types come with a business workflow attached in order to block changes to schedule types that are actively used on banking products. Read more about [Business Workflows](#).

Follow the steps below to create a new payment schedule type:

1. In the main menu, select **Products > Settings > Payment Schedule Type**, and the **Payment Schedule Types List** page opens, showing all the already created schedule types.
2. Click **Insert** to add a new type.
3. In the newly opened **Add Payment Schedule Type** page, fill in the following fields:

### Edit Schedule Type Definition

**Main Information**

Payment schedule code EPW	Name Equal principal Weekly	Banking Product Type Term Loan	Interest Calculation Type 52 weeks	Measurement Unit Weeks
Is With Equal Installments <input type="checkbox"/>	Installment Value Custom <input type="checkbox"/>	Use Fix Maturity Date <input type="checkbox"/>		

**Payment Schedule Type Details**

<input type="checkbox"/>	Column Repayment Schedule	Title	Calculation Method	Fee	Insurance class	Source	Display
<input type="checkbox"/>	RemainingValue	RemainingValue	RemainingFormula				(All) ✓
<input type="checkbox"/>	Interest	Interest	Effective Rate				✓
<input type="checkbox"/>	Principal	Principal	Linar				✓
<input type="checkbox"/>	Totalinstallment	Totalinstallment	ColumnFormula				✓

- **Payment schedule code:** Insert a code for the type to keep track of them.
- **Name:** Insert a suggestive name for the type.
- **Banking Product Type:** Select a product type to associate with the payment schedule type. Depending on the Product Type, different calculation rules are triggered. For example, the product type Overdraft has only the payment at maturity.
- **Day Count Conversion:** Determines how annual interests are converted to daily interests when calculating installments for time intervals measured in days. FintechOS Studio supports industry-standard conversions, such as 30/360, 30/365, Actual/Actual, where Actual for years can be either 365 or 366. You can define other schedule interest calculation types, as needed. In practice, calculation types such as Actual/360 or Actual/365 may also be encountered.
  - **Equal Installment Monthly 30/360** - payment schedule with equal installments, where every period is a month of 30 days, and every year has 360 days;

- **Equal Installment Monthly 30/365** - payment schedule with equal installments, where every period is a month of 30 days and every year has 365 days;
- **Equal Installment Monthly 30/366 (30/Actual)** - payment schedule with equal installments, where every period is a month of 30 days and every year has 366 days;
- **Equal Installment Monthly Actual/365** - payment schedule with equal installments, where every period is the month's actual number of days (28,29, 30, or 31) and every year has 365 days;
- **Equal Installment Monthly Actual/366 (Actual/Actual)** - payment schedule with equal installments, where every period is the month's actual number of days (28,29, 30, or 31) and every year has 366 days;
- **Equal Principal Monthly 30/360** - payment schedule with equal principals, where every period is a month of 30 days and every year has 360 days;
- **Equal Principal Monthly 30/365** - payment schedule with equal principals, where every period is a month of 30 days and every year has 365 days;
- **Equal Principal Monthly 30/366 (30/Actual)** - payment schedule with equal principals, where every period is a month of 30 days and every year has 366 days;
- **Equal Principal Monthly Actual/365** - payment schedule with equal principals, where every period is the month's actual number of days (28,29, 30, or 31) and every year has 365 days;

- **Equal Principal Monthly Actual/366 (Actual/Actual)** - payment schedule with equal principals, where every period is the month's actual number of days (28,29, 30, or 31) and every year has 366 days.

#### NOTE

The payment schedule projection takes into consideration both the payment schedule type defined at the banking product level, and whether to collect the interest accrued on the contract until the date of an early repayment event, thus repaying the accrued interest, or not, thus adding the accrued interest to the repayment amount. Set the **Collect accrued interest** field in the **Event** page at the contract level to **False** to add the accrued interest to the repayment amount.

- **Time Unit:** The type of measurement unit applicable for the payment schedule type: Days, Weeks, Months, Years, Once. The default is Months.
- **Annual Effective Rate Calculation:** Whether the annual cost of borrowing is calculated based on the Annual Percentage Rate (APR) or the Annual Percentage Rate of Charge (APRC).
- **Is With Equal Installments:** Select the checkbox if the installments are equal, so the annuity (also known as PMT) value calculation is needed, and the Principal is calculated with formula  $\text{Annuity} \text{ minus } \text{Interest}$

**Interest.** If there are Commissions that appear on the Payment Schedule, these Commissions are not included in the annuity calculation.

- **Installment Value Custom:** If you select the checkbox with multiple disbursements, the Principal component of the installments is the one calculated for the entire Financed Amount, even if it was not entirely disbursed.

For example, if Financed Amount is 10,000 EURO and the value calculated for Principal component of the Installments is 800 EURO, and the customer disburses only 5,000 EURO, the Principal component remains 800, but the Interest is calculated for the 5,000 EURO that were disbursed.

If Installment Value Custom = False, then the Is Manual Value and Royalty/ Initial Principal Value fields at the contract level are read only.

If Installment Value Custom = True, then the Is Manual Value field at the contract level is editable, with False default value.

- **Use Fix Maturity Date (from Activation Date):** If you select the checkbox, then the Maturity Date equals to Activation Date plus the Contractual Period in Months, i.e. the number of installments depends on the Activation Date.

If the checkbox remains unselected, the number of installments are fixed, the Maturity Date is equal to the First Installment plus the Contractual Period in Months, e.g. Installment date is on the first day of the month, this results in the Maturity day to be the first day of the month.

**NOTE**

This is only valid for Period Type = Months.

4. Click **Save and reload**.
5. In the newly displayed **Payment Schedule Type Details** section, add fields for any additional columns you want to display in the payment schedule (fees, insurances, or other payment dimensions). Note that the fields change according to the selection of **Column Type** and **Calculation Method**.

- **Title:** Insert a suitable title for the detail.
- **Column type:** Select one of the following types: **Payment Schedule Dimension**, **Fee Dimension** or **Insurance Dimension**. Note that the selected type triggers changes in the displayed fields.
- **Column Repayment Schedule:** Select one of the columns that is assigned in the Payment Schedule.
- **Calculation Method:** Select a method. It triggers changes in the fields. There are the predefined values that are taken into consideration when calculating the Interest:
  - **Linear:** calculated as Amount / Number of Installments.
  - **Effective Rate:** percent applied to the Source field (see below), usually to the Remaining Value.

- **Fixed Value:** a constant value that is completed in the Payment Schedule.
- **Column Formula:** formulas having other columns as parameters. Formula used is specified in the field **Formula** that is displayed when this option is set.
- **Remaining Formula:** specific to a column of Remaining.
- **Once:** the value is paid once.
- **LinearOnYear:** a value calculated based on the Remaining Value at the beginning of the credit year and a given percentage, divided in installments with the specified periodicity.
- **FeeOnce:** takes into account a fee that is paid once.
- **ProductLevelFixedValue:** method that calculates the capital percent defined for cards and credit cards type of products.
- **Operation Item:** It is the item allocated to a column in the Payment Schedule on which the amount calculated at this step is allocated. Select an item from the list.
- **Fee:** Select a fee configured in the Product Dimensions.
- **Fee Type:** Select a fee type configured in the Product Dimensions. This is helpful when, for different banking products, there are different values applicable for the same type of fee.
- **Source:** It is the source element from the JSON file returned by the Payment Schedule calculation from where to take the value.
- **Insurance type:** Select an insurance type.

- **Is Calculated In Advance?:** Select the checkbox if the calculation is done in advance.
- **Is calculated Upfront:** Option available for Insurance Dimension and Fee Dimension. Select the checkbox for insurance or commissions that need to be paid at the activation of the lending contract.

**NOTE**

This is only valid for products with Period Type = Months.

- **Totalization For Grace Period:** This checkbox is available when the **Calculation method** is set to Effective Rate or Fixed Value. If you select it, then the values that are not applied during Grace period are added to the first non-grace installment.
  - **Adjust Last Installment:** This checkbox should be checked most of the time, as rounding loses of precision can exist, and the sum of Principal paid monthly should be the same as the Financed Amount.
  - **Operation Item:** Select the operation item from the list for which this schedule type details should apply.
  - **Formula:** This checkbox is available when the **Calculation method** is set to Column Formula or Remaining Formula or Once or LiniarOnYear. The formula can use other Schedule Details Names. For example, for a total installment calculated as Principal+Interest+ManagementFee that were added to Payment Schedule Type Details earlier.
6. If you selected **Fee Dimension**, then fill in the following fields: **Fee**, **Fee Type** or **Source**.

7. If you selected **Insurance Dimension**, then fill in the following fields:

**Insurance class** or **Source**.

8. Click **Save and close**. Repeat to add more details.

**NOTE**

The first installment can be adjusted in situations when the algorithm forces an equal installment, by adjusting the Principal paid in the first interval.

For a Payment Schedule with equal installments, the first installment interest is calculated for the number of days between Activation Date and First Installment Date.

In general, for this first installment, the number of days can be lower or higher than the normal interval, so the Interest calculated makes the first installment to be different than the others.

The following picture illustrates the payment schedule type and the details defined for Credit Card Installments calculation:

Edit Schedule Type Definition

Main Information

Payment schedule code

CC

Name

Credit Card Installments

Banking Product Type

Interest Calculation Type

30/360

Measurement Unit

Months

Is With Equal Installments

Installment Value Custom

Use Fix Maturity Date

Payment Schedule Type Details

Export

Refresh

Column Repayment Sc...

Title

Calculation Method

Fee

Insurance class

Source

Display

Q1

Q1

Q1

Q1

Q1

Q1

IAH

CapitalPercent

CapitalPercent

ProductLevelFixed...

ManagementFee

MygFee

FixedValue

Management Fee FL...

SourceValue

RemainingValue

RemainingValue

RemainingFormula

Interest

Osoboda

Effective Rate

Principal

Principal

ColumnFormula

TotalInstallment

TotalInstallment

ColumnFormula

Schedule Template Detail

Main Information

Title

Principal

Column Type

Payment Schedule Dimension

Column Repayment Schedule

Principal

Calculation Method

ColumnFormula

Adjust Last Installment

Operation Item

Loan Principal

Formula

(remainingValue + managementFee)\*CapitalPercent/100.00

Schedule Type Column

TABLE OF CONTENTS

889

To configure the calculation steps, use the **Schedule Type Column** option in the **Settings** menu. The order of the steps is determined by the **Calculation column**.

1. In the main menu, click **Products > Settings > Schedule Type Columns**, and the **Schedule Type Column List** page opens, showing all the already created schedule type columns.
2. Click **Insert** to add a new column type.
3. In the newly opened **Add Schedule Type Column** page, select a **Column Type** from the list: **Payment Schedule Dimension**, **Fee Dimension** or **Insurance Dimension**.

Edit Schedule Type Column

**Main Information**

Column Type: Payment Schedule Dimension Name: InterestCapitalized

**Payment Schedule Types**

+ Insert X Delete Export Refresh

Name	Payment schedule code	Calculation Method	Column Type
deposit	deposit	Effective Rate	Payment Schedule Dimension
Deposit Actual	DA	Effective Rate	Payment Schedule Dimension

4. Enter a **Name** for the column.
5. Click **Save and reload**.
6. The **Payment Schedule Types** section is displayed after saving, showing all the payment schedule types using this column, with the following information:
  - **Name:** The name of the Payment Schedule element.
  - **Column Type:** The column type: Payment Schedule Dimension, Fee Dimension or Insurance Dimension.
  - **Payment Schedule Code:** The code of the payment schedule.

- **Calculation:** The calculation method.
- **Operation Item:** Select an operation item for this schedule type.

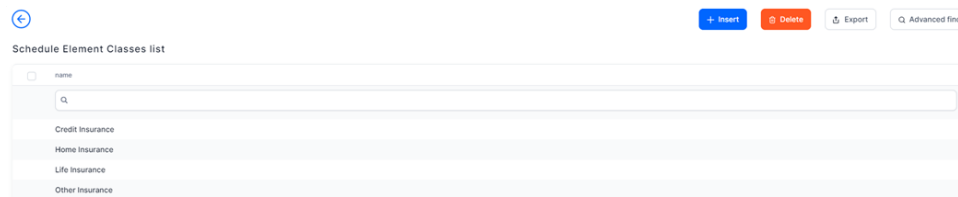
## Bancassurance Class

Bancassurance allows you to define classes of insurance for insuring your banking products. When associated to a payment schedule type, a corresponding insurance payment column will be displayed in the corresponding payment schedules.

### NOTE

Insurances with a periodicity set to *once* will not appear in payment schedules. The insurance must have a recurring periodicity to appear in a payment schedule.

1. In the main menu, click **Products > Settings > Schedule Element Class**, and the **Schedule Element Classes list** page opens.



2. Click **Insert** to add a new class.
3. In the newly opened **Add Scheduled Element Class**, select an **Element Type**: Insurance or Commission.

4. Add a **name** for the element class.
5. Select an **Operation Item**.

6. Modify the default values filled in for the following fields:

- **Mandatory for Applicant** - If selected, the schedule element class is applicable to the applicant customer. Default value: True.
- **Mandatory for Debtor** - If selected, the schedule element class is applicable to the debtor. Default value: False.

7. Click **Save and close**. Repeat as many times as needed.

### NOTE

FintechOS Studio uses the values of a saved schedule element class record to automatically fill in the fields of an insurance during the creation of a new record.

## Loan Periodicity

Loan periodicity is the dictionary used to define the regularity of payments. For example, payments related to loan contracts, commissions, or installments can be performed once, daily, monthly, yearly, and so on.

To manage loan periodicity records:

1. In the main menu, click **Products > Settings > Loan Periodicity**, and the **Loan Periodicity** page opens.

Loan Periodicity

Name	Periodicity	Measurement Unit
Once	0	Once
1DAY	1	Days
Weekly	1	Weeks
Monthly	1	Months
Bimonthly	2	Weeks
Trimestrial	3	Months
4 Weeks	4	Weeks
Semestrial	6	Months
Annual	12	Months
15Days	15	Days

5 10 20 1 2

2. On the **Loan Periodicity** page, you can add new loan periodicity record or search, edit, and delete existing ones.

**NOTE**

You can insert, update, or delete records if you have the associated role of [Banking Product Admin](#).

## Create Loan Periodicity Records

1. Click **Insert** on the **Loan Periodicity** page. The **Add Loan Periodicity** page is displayed.
2. Fill in the following fields from the **Loan Periodicity** section:

← Add Loan Periodicity Save and close Save and reload Save and new

**Main Information**

Name	Periodicity	Measurement Unit
Trimestrial	3	Months

- **Name:** Enter the name of the periodicity.
  - **Periodicity:** Enter the number of measurement units for the periodicity.
  - **Measurement Unit:** Select the measurement unit applicable for the periodicity from the drop-down. Possible values: Days, Weeks, Months, Years, and Once.
3. Click **Save and Reload**. The new loan periodicity is created and ready to be used.

## Collateral Type

The **Collateral Type** dictionary is used to define the types of collateral used in the banking products' definition.

To manage collateral type records:

1. In the main menu, click **Products > Settings > Collateral Type**, and the **Collateral Type List** page opens.

←

+ Insert

⊖ Delete

⬇ Export

🔍 Advanced find

Collateral Type list

<input type="checkbox"/>	Name	Adjust Percent
<input type="text"/>	<input type="text"/>	<input type="text"/>
	Cash	100.0000
	Commercial Real Estate	80.0000
	Equity	70.0000
	FixDeposit	100.0000
	Fixed Income	60.0000
	Government	90.0000
	Industrial Real Estate	80.0000
	Others	50.0000
	Residential Real Estate	80.0000
	Vehicles	20.0000

Or you can manage collateral types from FintechOS Portal's **Core Banking Operational > Collateral Type menu**.

2. On the **Collateral Type List** page, you can add new allocation methods or search, edit, and delete existing ones.

**NOTE**

You can insert, update, or delete records if you have the associated role of [Banking Product Admin](#).

Follow these steps to create new collateral type records:

1. Click **Insert** on the **Collateral Type List** page, and the **Add Collateral Type** page is displayed.
2. Fill in the following fields:

←

Save and close

Save and reload

Save and new

Add Collateral Type

Main Information

Name

Code

Adjust Percent

Priority

Cash

CA

100

4

Banking Products

- **Name:** Enter the name of the collateral type.
- **Code:** Enter the code of the collateral type.
- **Adjust Percent:** Enter the adjusted value of the collateral type.

#### NOTE

When a collateral type is defined, you can set an adjustment percent. The adjustment percent is a customization applied for that collateral. Further, when a collateral is linked to a secured loan contract, its market value is automatically adjusted by Loan Management using the adjustment percent.

- **Priority:** Enter the priority of the collateral type to be considered within a contract.

3. Click **Save and Reload**.

In the newly displayed **Banking Products** section, you can see the banking products that have this collateral type associated.

## Banking Product Document

This section allows you to track all the documents that have been set up in various banking "[Product Templates](#)" on [page 526](#) to be requested from or provided to banking customers, such as identification documents or Terms and Conditions.

## Insurance Peril

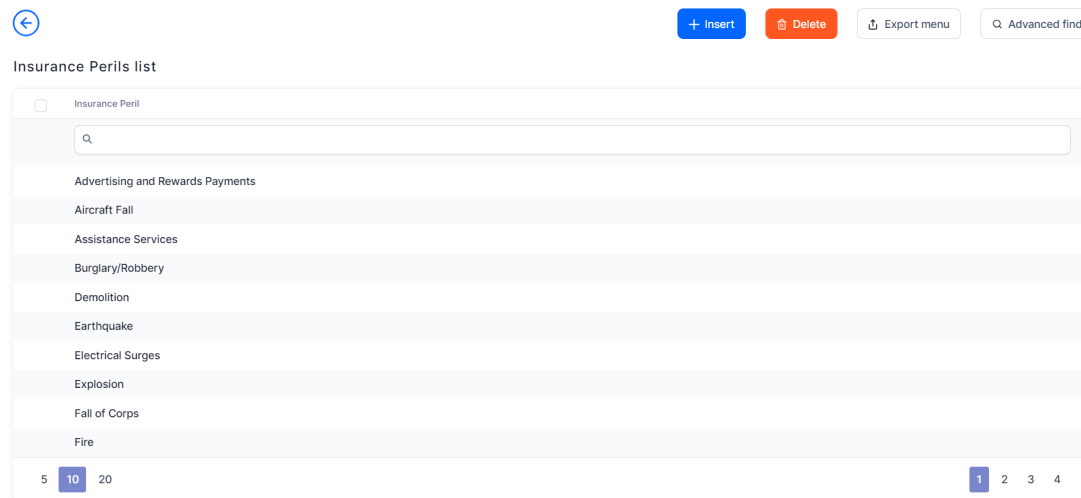
An **Insurance Peril**, or condition - for example earthquake, car accident, tornados, theft, death, or disability, informs about the type of coverage for a particular insurance product, or product item (coverage). You can define perils independently

from products, so that they can be used in conjunction with multiple insurance product templates. Depending on your product, you can attach one or multiple perils, or conditions to it.

## Add Perils

In **FintechOS Studio**, you can find an overview of all the insurance perils registered in your system - your nomenclature of perils, or conditions. You can also create and configure new records. Follow the steps below to add a new peril record.

1. In the main menu, navigate to **Products > Settings > Insurance Peril**. The **Insurance Peril List** is displayed, containing all the existing records.



2. Click **Insert** to add a new peril.
3. Fill in the following fields to configure the peril:
  - **Insurance Peril Name:** Name of the insurance peril.
  - **Max Notify Period:** Maximum period for the notification of the peril.
  - **Event Count Limit:** Number of events covered by the policy.
  - **Implicit Reserve:** Amount of the prudential reserve to be deposited for the current policy.
  - **Implicit Reserve Currency:** Currency of the prudential reserve.

Insurance Peril

Insurance Peril Name

Aircraft Fall

Max Notify Period

1

Event Count Limit

3

Implicit Reserve

2,000

Implicit Reserve Currency

EUR

4. Click **Save and Close**. You can view the newly created record in the **Insurance Peril List**.

## Insurance Document

This section allows you to track all the documents that have been set up in various insurance "[Product Templates](#)" on [page 526](#) to be requested from or provided to insurance customers, such as identification documents or Terms and Conditions.

## Lexicon Term

Lexicon terms are parameters (like the debt-to-income ratio, credit score, current credit amount, risk zone, etc.) used in the context of specific "[Product Templates](#)" on [page 526](#) for rules, data sets, or formulas. Lexicon terms are the building blocks of product evaluations, such as requiring an insurance, calculating a price, offering a discount, or determining eligibility.

To set up the lexicon terms that are available for your products, in FintechOS Studio, go to **Main Menu > Products > Settings > Lexicon Term**.

Each lexicon term has the following parameters:

Parameter	Description
Name	Name of the lexicon term.

Parameter	Description
Type	<p>Data type of the lexicon term.</p> <ul style="list-style-type: none"> <li>• <b>Boolean</b> - A selection between two mutually exclusive choices. Can be used for terms such as whether or not an applicant is an existing customer or if an insured house has or doesn't have an alarm system.</li> <li>• <b>Date</b> - A calendar date. Can be used for terms such as birth dates.</li> <li>• <b>Numeric</b> - A numeric value. Can be used for terms such as a credit amount or insured value.</li> <li>• <b>Numeric Array</b> - A list of numeric values. Can be used for terms such as the historical monthly account balances or premium payments history.</li> <li>• <b>Object</b> - A structured data item with multiple related key-value field pairs. Can be used for terms such as a single insurance policy containing fields like policy number, coverage type, and expiration date.</li> <li>• <b>Object Array</b> - A list of objects. E.g.: list of a customer's bank accounts, each with its own account number, balance, and account type.</li> <li>• <b>Text</b> - An alphanumeric string. Can be used for terms such as license plate numbers or VINs.</li> <li>• <b>Text Array</b> - A list of alphanumeric strings. Can be used for terms such as a list of previous insurers, a collection of policy reference numbers, or a set of beneficiary names.</li> <li>• <b>Value list</b> - A list of preset values. Can be used for terms such as home types, engine displacement ranges, or risk zones.</li> </ul>

Parameter	Description
Data Category	Select a classification for the type of data that the lexicon term is modeling (personal information, digital and behavioral data, legal representatives, employment and financial information, etc.).
Status	<ul style="list-style-type: none"> <li>• <b>Draft</b> - The lexicon term hasn't been used yet and can still be edited.</li> <li>• <b>Active</b> - The lexicon term is currently in use. It can no longer be modified.</li> <li>• <b>Inactive</b> - The lexicon term is suspended from use.</li> </ul>
Description	Enter a description of the lexicon term. This explanation is visible when designing a product and selecting the lexicon term.
Context Enriched Before	<ul style="list-style-type: none"> <li>• First choice of a <b>boolean</b> lexicon term (e.g.: <i>is, has, agrees with</i>). If selected, this choice is displayed before the lexicon term name (e.g.: <i>is in accordance with GDPR</i>).</li> <li>• For <b>non-boolean</b> lexicon terms, this is an optional text displayed before the term (typically a verb), e.g.: <i>is, has, covers</i>. You can combine it with the Context Enriched After parameter to create complex syntax such as: <i>covers ... % of the total surface</i>.</li> </ul>
Context Enriched After	<ul style="list-style-type: none"> <li>• Second choice of a <b>boolean</b> lexicon term (e.g.: <i>is not, doesn't have, doesn't agree with</i>). If selected, this choice is displayed before the lexicon term name (e.g.: <i>is not in accordance with GDPR</i>).</li> <li>• For <b>non-boolean</b> lexicon terms, this is an optional text displayed after the term (typically a descriptor), e.g.: <i>meters long, % of the total surface, consecutive days</i>. You can combine it with the Context Enriched Before parameter to create complex syntax such as: <i>covers ... % of the total surface</i>.</li> </ul>
Lexicon Term Values	Value list lexicon terms only. Use this grid to configure the values that are available in the list.
Lexicon Context	Use this list to set up the " <a href="#">Product Templates</a> " on <a href="#">page 526</a> where the lexicon term is available.

## Fee Types

To set up the fee types that are available for your products, in FintechOS Studio:

1. Go to **Main Menu > Products**.
2. Click **Settings**.
3. Select **Fee Types**.

Each type of fee has the following parameters:

Parameter	Description
Commission Schema	<p>Indicates the conditions under which the fee is applied.</p> <ul style="list-style-type: none"> <li>• <b>Administration Fee</b> - Management fee.</li> <li>• <b>Commission Unusage</b> - Fee applied for the amount unused from the contract's amount. Usually this is applied for unused overdraft amounts.</li> <li>• <b>Commission Usage</b> - Fee applied for the amount used from the contract's amount.</li> <li>• <b>Front-end Fee</b> - Fee paid at the beginning of the loan agreement to cover the costs of underwriting, evaluating, and processing the loan application. E.g.: loan origination fees, application fees, credit check fees.</li> <li>• <b>Management Fee</b> - Fee paid for administering and managing the loan throughout its duration, to cover operational costs such as regulatory compliance, maintaining the loan account, or providing ongoing customer support.</li> <li>• <b>Payment Holiday</b> - Fee charged when the borrower requests to temporarily suspend or defer making regular loan payments for a specific period.</li> <li>• <b>Repayment Fee</b> - Fee for repaying the loan in full or making larger payments than scheduled before the loan's maturity date.</li> <li>• <b>Reschedule</b> - Fee applied when the borrower seeks to modify the original repayment schedule, loan duration, or other terms specified in the loan agreement.</li> <li>• <b>Return Fee</b> - Fee charged when the borrower's payment is returned or cannot be processed.</li> <li>• <b>Versioning Fee</b> - Fee applied once when creating a new version</li> </ul>

Parameter	Description
	of an approved contract.
Name	Type a name for the fee.
Periodicity Type	How often the fee is charged. Either only once or at regular intervals (annually, bimonthly, trimestrial, weekly, etc.).
Mandatory for Applicant	This fee must be payed by the applicant.
Automatic Load on Contract	If selected, this fee is automatically loaded on contracts.
Is for Contract Version	Fee applied when creating a new contract version.
Is for Unusage	Fee for when an account or credit line remains unused or inactive for an extended period of time.
Is Returnable	Fee is refundable if the loan application is denied.
Is Return	Fee applied for Return Amount or Goods transactions on contracts.

## Interest Indexes

The Interest Indexes option allows you to track all the individual interests that have been set up in various banking "Product Templates" on page 526. To see the interests, in the FintechOS Studio main menu, go to **Products > Settings > Interest Indexes**.

## Insured Objects

### IMPORTANT!

For future use. Do not change.

This item allows you to see the **insured objects**<sup>1</sup> that can be covered in various insurance product templates.

---

<sup>1</sup>Item or entity that is covered by an insurance policy.

## Underwriting Data Set Values

This item allows you to define lists of terms that constitute valid terminology for the **Approved**, **Derogation**, and **Rejected** underwriting outcomes. For each outcome, you can provide the corresponding terms separated by commas:


Outcome	Data Set Values Examples
Approved	Approved, Passed
Derogation	Derogation, Manual Analysis
Rejected	Rejected, Not Passed

## Plan Dimension

Plan Dimensions, also referred to as Dynamic Dimensions, are values used to fine-tune claims or other financial parameters of an insurance plan (e.g., copay amounts).

Each plan dimension includes the following parameters:

- **Name** - The identifier of the plan dimension.
- **Allows Percentage** - Specifies how the plan dimension amount is defined:
  - **Unchecked** - The plan dimension amount is entered as a fixed value.
  - **Checked** - The plan dimension amount is calculated as a percentage of a reference value (e.g., a copay based on a percentage of the insured amount).



Save and close

Save and reload

Edit Product plan dimension

Product plan dimension

Allows percentage

☒

Name

copay

## Products APIs

Besides the Product Designer's intuitive interface, you can also manage your products through APIs. For more information, see the developer documentation below:

- Product Factory API V2
  - [Products APIs](#)
  - [Offers APIs](#)
- Legacy Product Factory API
  - [Products APIs](#)
  - [Offers APIs](#)
  - [Calculations APIs](#)

## Web API Client Library

**Products**, **offers**, and **calculations** APIs are also available in the form of a Web API Client Library. This provides easy integration within the FintechOS Platform development environment complete with automatic authorization. You can find the library in the Web API Client Libraries list under the name **PFAPI**. For more information, see ["Web API Client Libraries" on page 960](#).

## Server Automation Script Library

For **products** and **offers** APIs, you can further simplify your development workflow via a dedicated server automation script library. You can find the library in the Server Automation Script Libraries list under the name **PF\_Service\_Library**. For more information, see ["Server Automation Script Libraries" on page 1206](#).

This library eliminates the need to specify the baseUrl setting during import and offers more intuitive function names, such as `getMetadataByProductId` instead of `myImportedWebApiClientLibrary.getMetadataByProductId`.

### IMPORTANT!

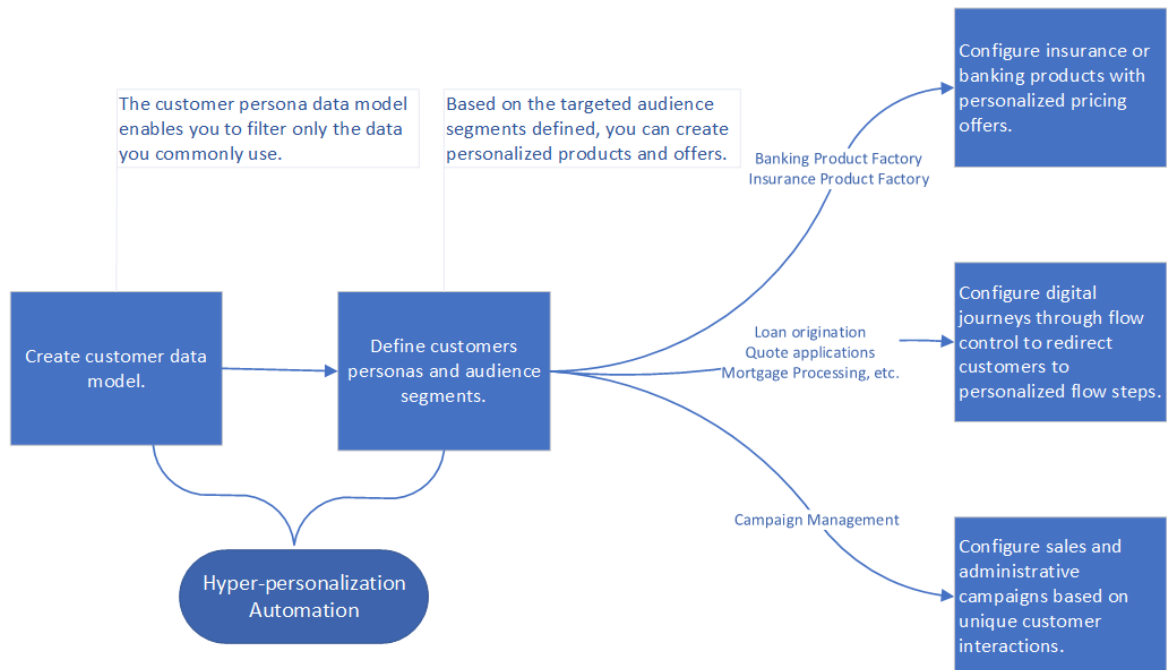
To use the library, configure the API's base URL in the `kv/<environment name>/<portal name>/app-settings/PFAPIBaseUrl` [Configuration Manager](#) key for any ["Digital Experience Portals" on page 402](#) or ["B2C Portals" on page 381](#) where the library's code will run.

# Hyper-personalization Automation

Businesses gather big amounts of information about customers' life-cycle and online behavior. Therefore, the way the data is used for driving personalized customer experiences across the buying journey is important for achieving hyper-personalization.

The Hyper-personalization automation block allows you to leverage customer data to create personalized customer experiences. You can define innovative operating models, with banking and insurance products that are data-driven and personalized based on the customer context. In addition, you can create personalized content and business-tailored segments of audience (customer personas) to further create effective sales or administrative campaigns, and more meaningful interactions with customers, thus ensuring customer loyalty. Based on the customer persona, you can configure a digital journey to include personalized banking or insurance products and offers.

Furthermore, you can add multiple sets of data to your digital solutions (loan applications, quote applications, mortgage processing, etc.) through third party integrations or use data from internal systems.



## Features

- Tailor content that fits each and every customer's wants and needs;
- Define customer personas;
- Define audiences to use personalized content in omni-channel campaigns.

## Customer Personas

Customer personas are customers who have a specific profile: demographic, location, age, needs, previous interaction with a business, etc. For example: customers with active contracts, which accepted marketing terms and conditions and which have more than five installments until the end of contract.

In FintechOS enables you to define the customer personas, filtering only the data (entities) you commonly use, leaving out all the structures that are meaningless for the targeted business cases, and audience segmentation needs.

It empowers you with the ability to create business-tailored segments of audience.

Using personalized content and delivery addresses for specific customer personas, you can orchestrate and customize omni-channel campaigns, easily change the rules of queries, and adapt any campaign to your business needs at any time.

To create customer personas:

1. [Define the customer persona data model.](#)
2. [Create customer persona.](#)

## Add Customer Persona Data Model

The customer persona data model enables you to filter only the data (entities) you commonly use, leaving out all the structures that are meaningless for targeted business cases and audience segmentation needs.

FintechOS enables you to define the list of entities that you might want to use for creating advanced audience segmentation (customer personas).

To define the data required for customer personas:

1. Click the main menu icon at the top left corner.
2. In the main menu, click **Hyper-Personalization > Persona Data Model Settings**. The **Audience Data Model Configurations List** page opens.
3. Click the **Insert** button at the top right corner of the page. The **Add Audience Data Model Configuration** page opens.
4. Select the **Entity** whose attributes you'll be using when creating the customer personas. The **Name** field will be automatically filled in with the display name of the selected entity.

ADD AUDIENCE DATA MODEL CONFIGURATION

Entity

Account

↓

Name

Customer

Description

ATTRIBUTES

5. Click the **Save and reload** button at the top right corner to save the persona data model settings. The **Edit Audience Data Model Configuration** page opens and all attributes in the selected entity data model are listed.

EDIT AUDIENCE DATA MODEL CONFIGURATION

Entity

Account

↓

Name

Customer

Description

ATTRIBUTES

+ Insert

X Delete

Export

Refresh

<input type="checkbox"/>	Name	Attribute	Description	Include In Audience Segment Data	View
	Q	Q	Q	(All) ▾	
	Account Type	typeId		<input type="checkbox"/>	View
	Accountid	Accountid		<input type="checkbox"/>	View
	Age	Age		<input type="checkbox"/>	View
	Annual Income	AnnualIncome		<input type="checkbox"/>	View
	Business Status	businessStatusId		<input type="checkbox"/>	View
	Business Unit	businessUnitId		<input type="checkbox"/>	View
	City	City		<input type="checkbox"/>	View
	Commercial registration number	CommercialRegistration		<input type="checkbox"/>	View
	Country	accountCountryId		<input type="checkbox"/>	View
	Created by user	createdByUserId		<input type="checkbox"/>	View

6. If for the data segmentation (customer personas), you only need specific entity attributes, in the **Attributes** section, tick the **Include In Audience Segment Data** checkbox for the attributes that you will be using during the creation of customer personas.

NOTE

When defining audience segments, you will have access only to the

attributes for which you have ticked the **Include In Audience Segment Data** checkbox.

7. Optionally, you can provide a relevant description to all the attributes on a context entity using inline editing.
8. Click the **Save and Close** button at the top right corner to save the persona data model settings..

## Create Customer Personas

Customer personas apply advanced filtering criteria on the information available in the [Add Customer Persona Data Model](#) to segment your customer base depending on your communication strategy.

To create a customer persona, follow these steps:

### 1 Insert a new customer persona record

1. In FintechOS Studio, click the main menu icon at the top left corner.
2. In the main menu, click **Hyper-Personalization > Customer Personas**. The **Audience Segments List** page opens.
3. Click the **Insert** button at the top right corner of the page. The customer persona digital journey appears.

### 2 Define the customer persona

By default, the customer persona digital journey displays the **Define** step.

1. Type the **Name** of the customer persona. Provide a descriptive name for the customer persona so that you can easily identify the target group at a first glance.
2. Click the **Save and reload** button at the top right corner of the page to save the record.

### 3 Define audience segments

You can use an audience segment for various audiences. This way you will define a set of conditions once and you will be able to use it in different combinations when defining an audience. Repetitive conditions are easy to use in segmentation, once you created a segment you can simply re-use it anytime later.

A common use case is the audience segment covering the set of conditions for minimum eligibility criteria, that you have to add to all your campaigns. For example, customers with age between 24 – 73, that have Marketing Acceptance, that are not deceased, that have a mobile phone, etc.

#### HINT

Create simple segments that can be easily combined in various digital audiences later on.

To define an audience segment, follow these steps:

1. Click the **Segment Expression** step.
2. Set conditions to get specific records for specific entity (entities). For more information, see [Get Records for Entity](#).
3. Define aggregate data to use as token or to add new conditions to filter the data set. For more information, see [Define Aggregate Data](#)
4. Set the tokens that will be used in communication templates. For more information, see [Set Tokens for Personalized Content](#).

## Preview your audience segment

Once you have defined and saved your segment expression, you can generate an Excel file that lists all the audience segment data. The preview is generated asynchronously by a job scheduler and you have the option to receive an email notification when the Excel file is ready. To create an audience segment preview:

1. In the audience segment editor, select the **Segment Preview** tab.
2. Click **Insert** to create a new segment preview.
3. In the Add Generated Preview window, enter a **Begin Date** for when you wish to schedule the processing for the preview and select the **Send Mail** checkbox if you wish to send an email notification when the preview is ready.
4. Click **Save and Reload** at the top right corner of the page.
5. In the Edit Generated Preview page, you can modify the **Begin Date** and **Send Mail** options and customize the **Email Address** where you wish to receive the notification once the preview is ready.
6. Click **Save and Close** at the top right corner of the page.

The Segment Preview tab lists all the audience segment previews tasks that have been finished, are in progress, or are scheduled to run in the future.

Once a preview generation is finished, if you select it from the list, you will be able to download the Excel file containing audience segment preview data.

## Customer Personas Examples

### Segment with Group Condition Filter

Customers not deceased, with valid mobile phone, who accepted marketing terms and conditions.

Entity	Part	Expression
<b>Customer</b>	Filters	(isDeceased equal true AND MobilePhone is not blank AND (AcceptanceForMarketing equals true OR AcceptanceForMarketing is blank))
	Tokens	Phone, Email, First Name, Last Name
<b>Aggregates</b>	Defined	-
	Filters	-

Get all records for entity

Customer

Use following filters on data

And +

- Is Deceased Equals false
- Mobile Phone Is not blank
- Or +
  - Acceptance for marketing Equals true
  - Acceptance for marketing Is blank

Make following data available for tokens

Last name ✕ First Name ✕ Email ✕ Mobile Phone ✕

## Segment with More Linked Entities and Filtered by the Second Entity

Customers with applications in the past 6 months.

Entity	Part	Expression
<b>Customer</b>	Filters	-
	Tokens	Phone, Email, First Name, Last Name
<b>Credit Application</b>	Filters	ApplicationDate.MonthsSince is less than or equal to 6.
	Tokens	Application Status, Product Name, Application Value, Application No, Application Date
<b>Aggregates</b>	Defined	
	Tokens	

Get all records for entity

Customer

Use following filters on data

And +

Make following data available for tokens

Acceptance for marketing X Last name X Title X Name X First Name X Email X Phone X

Link related entity

ThirdParty Credit Application

Use following filters on data

And +

X Application Date.Months since Is less than or equal to 6

Make following data available for tokens

Application Status X Product Name X Application Value X ApplicationNo X Application Date X

## Segment with Aggregates Data

Customers with active contracts, which accepted marketing terms and conditions and which have more than five installments until the end of contract.

Entity	Part	Expression
<b>Customer</b>	Filters	AcceptanceForMarketing equals true OR AcceptanceForMarketing is blank.
	Tokens	Phone, Email, First Name, Last Name
<b>Contract</b>	Filters	Status equals active
	Tokens	-
<b>Installment</b>	Filters	PayedValue equals 0 AND DueDate.DaysSince is less than 0.
	Tokens	-
<b>Aggregates</b>	Defined	Count(Installments) as InstallmentsNo.
	Tokens	InstalmentsNo is greater than or equal to 6.

Get all records for entity

Customer

Use following filters on data

Or

Acceptance for marketing

Equals

true

Acceptance for marketing

Is blank

Make following data available for tokens

Acceptance for data processing

Acceptance for marketing

Account Type

First Name

Function

Gender

Link related entity

TSTContract

Use following filters on data

And

Status

Equals

Active

Make following data available for tokens

Select attributes to include

Link related entity

TSTInstallment

Use following filters on data

And

Payed Value

Equals

0

Due Date.Days since

Is less than

0

Make following data available for tokens

Select attributes to include

Define Aggregates Data

Aggregate Type	Entity Source	Source Attribute	Date function	Aggregates Level	Alias	
COUNT	TSTInstallment	TSTInstallmentid		Customer	InstallmentsNo	Delete

Aggregates Filter

And

InstallmentsNo

Is greater than or equal to

6

## Get Records for Entity

To set up a query and define a segment, for each level, select the entity and set all the conditions that will be applied on the selected entity.

In the entity section, you can add all entities involved in the query. Entities start with the main entity and continue with related entities, one for each level. Usually, the first entity is the Customer entity, but can be any entity from the data model.

### NOTE

- When defining a segment, the first entity selected to define conditions and tokens becomes the context entity of that segment.
- Only one related entity can be linked to a parent entity. You cannot add several related entities to the same parent entity. For example, if you select the Customer entity as your context entity, you can only add conditions for “Loan Applications” or for “Loan Agreements”, not for both in the same segment.

We recommend users to create two segments using the following route:

Customer

→ Loan Application

Customer

→ Loan Agreements

Correct use cases:

- You can link Customer -> Contract -> Installments. That means three deep levels, one entity on each level (customer on the first, contract linked by customer on the second and installment linked by contract on the third).
- You cannot link Customer -> Contract, Financial Assessment because both Contract and Financial Assessment are linked by the same entity, Contract.

You can set complex conditions using simple or nested group conditions. The figure below shows the Customer entity filtered by age between 18 and 74 and acceptance for marketing equal true.

You can set up a complex filter by clicking the plus (+) sign and **Add Group**

A new nested group will be added to the filter condition you can add a new expression similar to: ((a>=10 or b='condition') and c=true).

The table below provides all the settings you can do in the entity section:

Setting	Description
<b>Get all records for entity</b>	Select the main entity that will be the first level of query. On the second level of query, you can insert only one related entity.
<b>Use following filters on data</b>	Add filters on the selected entity. You can add a condition by clicking the plus sign and selecting <b>Add Condition</b> . A new condition will be inserted.
<b>Condition sign</b>	The AND condition is displayed by default; however, you can select another condition: OR, NOT AND and NOT OR.
<b>Delete a condition</b>	Click the X sign to delete an inserted condition.

Setting	Description
Condition attribute	<p>The next field represents one of the entity's attributes. You can select it by clicking on the field and opening a list with all the attributes. The date type attributes are marked with an arrow and you can select one of these functions:</p> <ul style="list-style-type: none"> <li>• Days since – the count of the days which have past from the date until now.</li> <li>• Months since – the count of the months which have past from the date until now.</li> <li>• Years since – the count of the years past from the date until now.</li> <li>• Year of – the year of the date.</li> <li>• Month of – the month of the date.</li> <li>• Date of – returns only the date part (without time) from the date.</li> <li>• Weekday of – the weekday number of the date.</li> <li>• Has anniversary today – returns true if today has the same day and month as the date.</li> <li>• Days until – the count of the days from today until the future date</li> <li>• Months until – the count of the months from today until a future date</li> <li>• Years until – the count of the years from today until a future date.</li> <li>• Day Of – the day of the date.</li> <li>• Days until Anniversary – the count of the days from today until the date.</li> </ul>

Setting	Description
	<ul style="list-style-type: none"> <li>For “since” functions, if the date is a future date, then the return count will be negative.</li> <li>For “until” functions, if the date is a past date, then the return count will be negative.</li> </ul>
<b>Condition operator</b>	<p>Select the conditional operator by clicking on the label and selecting the desired one from the list.</p> <p>For the date type attributes there are two more operators: @Equals and @Does not equal, that compare the date with some predefined parameters such as current_year, current_month, current_day, current_weekday. The values for these parameters are evaluated at runtime.</p>
<b>Condition value</b>	<p>The last part of a condition is represented by value. There are different types of values for each type of attribute and operator. You can add a simple text or number, an interval of numbers, values from other entity for lookup attribute type, option set items and values based on current date as those above.</p>
<b>Group condition</b>	<p>Add a group condition by clicking the plus sign and selecting <b>Add Group</b>.</p> <p>You can also add multiple levels of groups (nested groups).</p>
<b>Make following data available for tokens</b>	<p>Select attributes from the selected entity and use them as tokens in digital content templates to tailor the communication with targeted customers. An attribute can be searched typing the first letters or browsing it in the list.</p>
<b>Add Entity</b>	<p>Add a related entity to current query.</p>
<b>Links related to entity</b>	<p>The list is populated with entities related to the main entity and gives you the possibility to choose only one. If more than one related is needed for the same entity, you should create another segment.</p>

## Define Aggregate Data

In the **Define Aggregates Data** section, you can define the data to use as token or to add a new condition to filter the dataset. To add aggregate data, next to the **Define Aggregates Data** section title, click the **Add a row** icon. An empty row is added in the table. Using inline editing, set the following fields:

Field	Description
<b>Aggregate Type</b>	Select an aggregate function applied over the dataset: SUM, MAX, MIN, COUNT, AVG
<b>Entity Source</b>	Represents one of the selected entities, the source for the aggregate data. The drop-down list is populated only with the entities selected in the Get all records for entity section.
<b>Source Attribute</b>	Select one of the entity source attributes. It is used to calculate the aggregate function.

Field	Description
<b>Date function</b>	<p>If the source attribute is a datetime attribute, you can use a date function to obtain different kind of information starting with the date:</p> <ul style="list-style-type: none"> <li>• Days since – the count of the days which have past from the date until now</li> <li>• Months since – the count of the months which have past from the date until now</li> <li>• Years since – the count of the years past from the date until now</li> <li>• Year of – the year of the date</li> <li>• Month of – the month of the date</li> <li>• Date of – returns only the date part (without time) from the date</li> <li>• Weekday of – the weekday number of the date</li> <li>• Has anniversary today – returns true if today has the same day and month as the date</li> <li>• Days until – the count of the days from today until the future date</li> <li>• Months until – the count of the months from today until a future date</li> <li>• Years until – the count of the years from today until a future date.</li> <li>• Day Of – the day of the date.</li> <li>• Days until Anniversary – the count of the days from today until the date.</li> <li>• For “since” functions, if the date is a future date, then the return count will be negative.</li> <li>• For “until” functions, if the date is a past date, then the return</li> </ul>

Field	Description
	count will be negative.
<b>Aggregates Levels</b>	Represents one of the query levels used by the function to aggregate. It works like a group by clause.
<b>Alias</b>	Alias that you can use in aggregates filter.

You can add segmentation conditions based on aggregated attributes. This is similar to filtering entities' attributes. You can define aggregated fields only by using the related entities added in the entity conditions definition section.

Example of aggregated fields:

- The total number of active contracts the customer has – use “Count” Function for “Loan Agreement”, aggregated at the customer level. Conditions for the Contract status = Active will be defined in the initial condition definition area at the “Loan Agreements” level.
- The rejected applications at the customer level - use “Count” Function for “Loan Application”, aggregated at the customer level. Conditions for Loan Application Status = Rejected will be defined in the initial condition definition area at the “Loan Applications” level.

Example of a condition added for an aggregated field:

- The number of active contracts <= 1 – to select the customers that have at least 1 active contract
- Number of Rejected Applications = 0 – to select the customers with no rejected applications

SEGMENT		CONTEXT ENTITY	
007.New Segment		Account	
Define Aggregates Data <span style="float: right;">+</span>			
Aggregate Type	Entity Source	Source Attribute	Aggregates Level
SUM	TSTContract	CTR Amount	Customer
			Contract Amount <span style="float: right;">Delete</span>
Aggregates Filter			
<div> <span>And</span> <span>+</span> </div> <div> <span>×</span> Contract Amount Is greater than or equal to 10000         </div>			

## Set Tokens for Personalized Content

In the **Return Following Tokens** section, select all the tokens that will be used in personalized content. The section allows inline editing records.

### NOTE

In order to set tokens for personalized content, you need to first create the content tokens. For information on how to add content tokens, see [Content Settings](#).

To set a token to be used in personalized content, follow these steps:

1. Next to the **Return Following Tokens** section title, click the **Add** icon (+). An empty line is added to the list.
2. In the empty line, click in the **Source Name** column and select the entity which contains the attribute whose value will be returned in personalized content.
3. Click in the **Attribute** column and select the attribute whose value will be returned in the personalized content.
4. Press **ENTER**. The alias of the attribute you selected at the previous step will be automatically displayed in the **Alias** column.

Follow the steps above to set all the token that you want to use.

Return Following Tokens <span>+</span>			
Entity Source	Attribute	Alias	
Customer	Account Type	CustomerAccount_Type	Delete
Customer	Last name	CustomerLast_name	Delete
Customer	Name	CustomerName	Delete
Customer	First Name	CustomerFirst_Name	Delete
Customer	Email	RECEIVER_EMAIL	Delete
Customer	Mobile Phone	RECEIVER_PHONE	Delete
TSTContract	CTR Date	TSTContract.CTR_Date	Delete
TSTContract	CTR Amount	TSTContract.CTR_Amount	Delete
TSTContract	CTR Type	TSTContract.CTR_Type	Delete

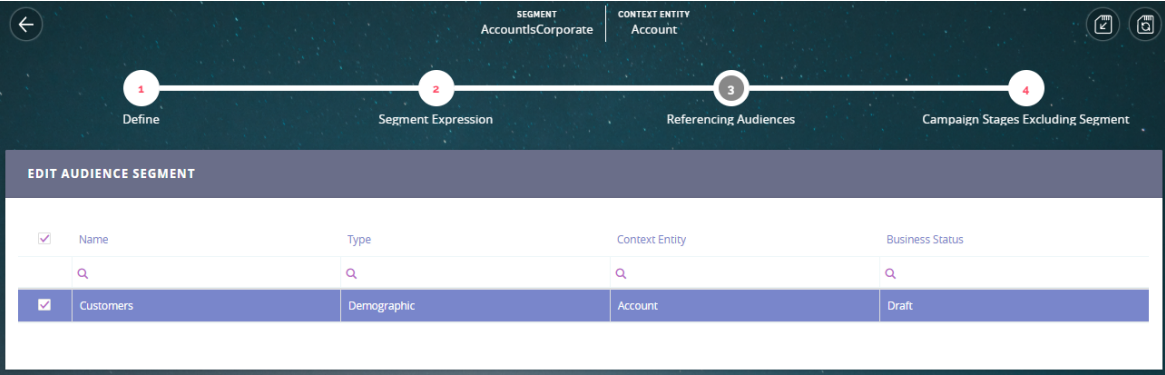
**IMPORTANT!**

If you add the customer persona to audiences which will be used in campaigns, you should provide two tokens, as follows: the email and phone information. You can do so by adding for the Email and Mobile Phone attributes the alias RECEIVER\_EMAIL and RECEIVER\_PHONE.

5. Click the **Save and Close** button at the top right corner to save the content type.

## View Audiences that Use Current Customer Persona

To view the list of audiences that use the current customer persona, in the customer persona configuration page, click the **Referencing Audiences** step.



**NOTE**  
For the **Referencing Audiences** list to show any records, you should add the segment to at least one digital audience.

The **Referencing Audiences** list provides the following details (field values):

Field	Description
Name	The name of the audience.
Type	The audience type.
Context Entity	The entity used to combine all the segments into an audience.
Business Status	The audience status.

# View Campaign Stages Excluding Segment

To view the list of campaigns that exclude the data returned by the current segment, in the customer persona configuration page, click the **Campaign Stages Excluding Segment** step.

**NOTE**  
For the **Campaign Stages Excluding Segment** list to show any records, you should exclude the segment from at least one campaign.

The Campaign Stages Excluding Segment list provides the following details (field values):

Field	Description
<b>Name</b>	The campaign stage name.
<b>Start Date</b>	Campaign stage start date.
<b>End Date</b>	Campaign stage end date.
<b>Schedule Type</b>	The campaign stage schedule type as it has been defined into a campaign.
<b>Recurrence Type</b>	Campaign stage recurrence type.
<b>Business Status</b>	The state of the campaign stage state.
<b>Created On</b>	Campaign stage creation date.
<b>Created by</b>	The user who created the campaign stage.

## Audiences Management

Audiences are groups of people with specific interests, intents, demographics or education levels and can be reached based on who they are and how they have interacted with the business. The resulting list of individuals based on the defined conditions in the definition of customer personas and audiences will represent the target group of a campaign.

FintechOS provides you with three types of audiences:

- **ImportedList.** If the campaign has started and activities have been sent and you need to correct the list, you should redo the campaign.

### NOTE

Once activities have been generated, excluding the preview ones, you can no longer delete the list.

- **Static.** The audience is selected and the list is generated. Once the campaign is started, it behaves similar to audience of type ImportedList.
- **Dynamic.** Every time a campaign is executed, the audience will also be executed, meaning that new members and member instances will be generated in case new content settings have been exposed by the audience.

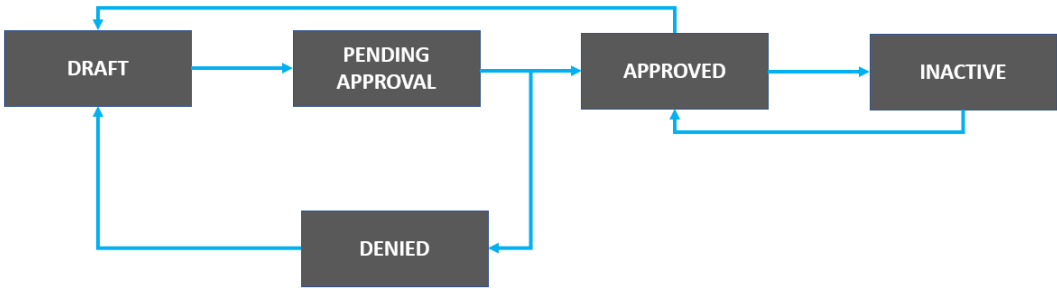
**IMPORTANT!**

Campaigns with dynamic audience always take into account how the audience looks like, the customer personas and personalized content at the moment the campaign stage is run.

The table below describes the states of an audience:

State	Description
Draft	The first state of an audience. Only in this state, an audience can be modified.
Pending Approval	In this state, an audience is waiting to be approved or denied.
Approved	The audience has been approved to be used in campaigns. You can only use approved audiences in campaigns.
Denied	The audience has not been approved to be used in campaigns.
Inactive	The audience has been approved but has been removed.

The figure below presents the audience state transitions:



## View Audiences

To view the list of defined audiences, in FintechOS Studio, click the main menu icon at the top left corner and click **Hyper-Personalization > Audiences**. The **Audiences List** page opens listing all audiences that have been created by you. If none, the list will show no data.

### Edit audiences

#### NOTE

You can only modify audiences that are in the initial state, that is **Draft**.

To edit an audience, in the **Audiences List** page, double-click the record that you want to edit. The audience configuration page appears. You can edit audience settings from the following two sections: **Define** and **Setup** and you can also change the audience status. Make the desired changes and click the **Save and Close** button at the top right corner to save the changes.

### Preview audiences

Once you have defined and saved an audience, you can generate an Excel file that lists all the audience data. The preview is generated asynchronously by a job scheduler and you have the option to receive an email notification when the Excel file is ready. To create an audience preview:

1. In the audience editor, select the **Preview** tab.
2. Click **Insert** to create a new audience preview.
3. In the Add Generated Preview window, enter a **Begin Date** for when you wish to schedule the processing for the preview and select the **Send Mail** checkbox if you wish to send an email notification when the preview is ready.
4. Click **Save and Reload** at the top right corner of the page.

5. In the Edit Generated Preview page, you can modify the **Begin Date** and **Send Mail** options and customize the **Email Address** where you wish to receive the notification once the preview is ready.
6. Click **Save and Close** at the top right corner of the page.

The Segment Preview tab lists all the audience previews tasks that have been finished, are in progress, or are scheduled to run in the future.

Once a preview generation is finished, if you select it from the list, you will be able to download the Excel file containing audience preview data.

## Create Audiences

An audience represents a combination of multiple customer personas, that have the same persona data model settings (i.e. the audience segments are based on the same entity).

### NOTE

Before creating audiences, you should create at least one persona data model setting and at least one customer persona.

To create an audience that you can later use in your campaigns, follow these steps:

### 1 Insert audience

In the **Audiences List** page, click the **Insert** button at the top right corner of the page. The audience configuration page opens.

### 2 Define the audience

The audience configuration page opens by default on the **Define** tab.

Type the **Name** of the audience and select the **Context Entity**, that is the entity you added the [customer persona data model](#), then click the **Save and reload** button at the top right corner of the page to save the audience. The audience will have the status **Draft**:

You can now set up the audience.

### 3 Set up the audience

Click the **Setup** tab.

You can combine the desired customer personas using union, intersect or except, directly or creating subgroups.

#### NOTE

You cannot combine in the same audience definition customer personas which have context entities. Audiences can be defined only by combining customer personas with the same context entity.

The table below provides all the settings you can do in the Setup section:

Setting	Description
<b>Include</b>	Allows you to combine segments using union and intersect operators.
<b>Include Operator</b>	To combine segments, you should select at least one intersect or union operator. Intersect operator adds only common data from combined segments to audience. Union operator adds all data from all combined segments to audience.
<b>Add Audience Segment</b>	Allows you to add an audience segment by clicking the plus (+) sign and selecting Add Audience Segment. After selecting the desired audience segment, it will be inserted into the digital audience.
<b>Add Group</b>	Allows you to add a group of audience segments by clicking the plus (+) sign and selecting Add Group.
<b>Delete an audience segment</b>	Click the X sign to delete an inserted audience segment.
<b>Choosing another audience segment</b>	Allows you to use another audience segment by clicking on audience segment and selecting another one from the drop-down list.
<b>Exclude</b>	Allows you to combine segments using union and intersect operators. The resulted data will be used to exclude records from the audience. You can combine more audience segments with union and intersect and you can also create nested groups of audience segments.

## Audience Setup Examples

### Audience with Segments Combined in Nested Groups

Customers with some eligibility conditions, that have cash contracts OR no application in the last 3 months.

Frame	Segments Expression
<b>Include</b>	Minimum Eligibility Segment INTERSECT (Cash Contract Segment UNION No Application in the last 3-month Segment)
<b>Exclude</b>	-

### Audience with Exclude Segments

Audience with clients over 40 years old with children and annual income between 50000 and 85000, that don't have contracts and accepted marketing terms and conditions.

Frame	Segments Expression
<b>Include</b>	Clients Over 40 Segment INTERSECT Clients with children and income between 50000 and 8500 Segment
<b>Exclude</b>	Client with contract Segment UNION Client without Data Processing Acceptance Segment

## View Referenced Segments

To view the list of segments used in the audience, click the **Referenced Segments** tab. All the segments listed have been used when setting up the audience (**Setup** section).

## View Campaigns Referencing Current Audience

To view the list of campaigns that use a specific audience, in the audience configuration page (**Audiences List** page, double click on the desired audience), click the **Referencing Campaigns** tab.

**HINT**

For the **Referencing Campaigns** list to show any records, you should be using the digital audience in at least one campaign.

The **Referencing Campaigns** list provides the following attributes:

Attribute	Description
Name	The name of the campaign.
Campaign Type	The type of campaign.
Campaign Subtype	The subtype of campaign.
Start Date	The date when the campaign will start.
End Date	The date when the campaign will end.
Business Status	The state of campaign.
Template	The personalized content used in campaign.

# Servicing

The Servicing menu hosts the app data forms and back-office flows.

**App data forms** allow you to create digital applications, typically for middle and back office operations, where users can navigate freely between form steps and save, close, or reload the form at will. Unlike ["Form Driven Flows" on page 221](#), app data forms are not constrained by a predefined flow map and don't require navigation buttons.

For more complex workflows, you can extend app data forms with **back-office flows**—dedicated ["Digital Journeys" on page 189](#) that users can trigger from an app data form as needed. Back-office flows run within the data context of the app data form that initiated them. Once the flow is completed, users are returned to the form at the point where they left off.

---

## App Data Forms

App data forms allow you to create digital applications, typically for middle and back office operations, where users can navigate freely between form steps and save, close, or reload the form at will. Unlike ["Form Driven Flows" on page 221](#), app data forms are not constrained by a predefined flow map and don't require navigation buttons.

## Prerequisite

You need to have the underlying data model defined (entity and attributes), including any required data extensions (presented in the ["Extend the Data Model" on page 1](#)).

## Create an App Data Form

1. In FintechOS Studio, from the main menu, click **Servicing > Service Extension Apps > App Data Forms**.
2. Click **Create**. The form configuration page appears which is comprised of two sections.

It displays by default on the **General** section.

Save and close

Save and reload

Save and new

1 General

2 Data Model

Name

IS\_App\_DataForm

Display Name

IS\_App\_DataForm

Description

Show Tooltips

User Settings

▼ Properties

Hide Business Workflow

Default

Read Only

Default

Disable Save Keyboard Shortcut

Default

Show Bullets Progress Bar

Yes

Flow Title

Default

Style Sheets

Select items to include

Hide Business Transaction

Default

Hide Action Buttons

Default

Disable prompt for unsaved changes

Default

Save automatically data on leave

No

## 3. Provide the general information:

Field	Description
Name	<p>The name of the app data form used internally by the system. This field is mandatory.</p> <div> <p><b>NOTE</b></p> <p>A naming convention is an important part in a well-built data model; therefore, we recommend you to use PascalCaseNames (upper camel). The Name starts with an uppercase letter, as do all additional words. For example: FTOS_CMB_BaseAccount.</p> </div>
Display Name	This is the name that displayed in the user interface. Choose a suggestive name.
Description	Optional description of the form.
Show Tooltips	<p>Select how you want the tooltips to be shown for specific attributes in the user interface. The following values are available:</p> <ul style="list-style-type: none"> <li>• User Settings (default value). Give users the possibility to show tooltips by toggling on / off the Tooltips button displayed at the top-right corner of the UI.</li> <li>• YES. Always show tooltips in the Portal UI when users hover their mouse on the attributes on fields which have tooltips. The users do not have the possibility to toggle the tooltips off.</li> <li>• No. Never show tooltips in the Portal UI.</li> </ul>

Field	Description
Hide Business Workflow	Hides the entity record's state and state transition options in the end-user interface. For more information about business workflows, see the <a href="#">Business Workflows Processor documentation</a> .
Read Only	Prevents end-users from making changes to the displayed form fields.
Disable Save Keyboard Shortcut	Prevents end-users from saving and reloading the form by pressing the Ctrl+S keyboard shortcut.
Show Bullets Progress Bar	<p>The numbering of the steps appear on the progress bar. Choose one of the following:</p> <ul style="list-style-type: none"> <li>• Default. The progress bar is not shown.</li> <li>• Yes. The progress bar is shown and the navigation is done through the Next/ Previous buttons.</li> <li>• No. The progress bar is not shown.</li> </ul>
Render section tabs as a bullet list	If you tick this checkbox, the steps are displayed in the Digital Experience Portal as a bullet list and you can navigate between steps by clicking on the round numbered bullets.
Flow Title	<p>Select a title from the list:</p> <ul style="list-style-type: none"> <li>• use display name (this name appears everywhere, on every step this name is exhibited)</li> <li>• show only step display name (each step has its own name exhibited).</li> </ul>
Style sheets	Select the style sheet you wish to use. Multiple style sheets can be added. The order of how they are added is respected in the execution. For details, see <a href="#">"Style Sheets" on page 1209</a> .

Field	Description
Hide Business Transaction	Hides the Business Transactions button at the top right corner of the screen, which displays the business workflow transitions of the record. For more information about business workflows, see the <a href="#">Business Workflows Processor documentation</a> .
Hide Action Buttons	Hides the Actions button at the top right corner of the screen, which allows you to open action groups or reports associated with the app data form (for details, see <a href="#">"Defining Action Groups" on page 263</a> and <a href="#">"Add Power BI Reports to Digital Journeys" on page 1035</a> ).
Disable prompt for unsaved changes	When navigating away from the form, for instance by clicking the Go Back button in the top left corner of the screen, if the form contains unsaved data, the user is not be prompted to save changes.
Save automatically data on leave	When navigating away from the data form, for instance by clicking the Go Back button in the top left corner of the screen, any unsaved data in the flow is automatically saved.

4. Click the **Data Model** tab and from the **Entity** field, select the entity whose data model is used.

The screenshot shows the 'Data Model' tab in Fintechos Studio. The 'Entity' dropdown is set to 'FTOS\_PWC\_Loan'. Below it, the 'Business Entity Extensions' section is expanded, showing a table with columns: Name, Extension Type, and Relation Attribute. The table contains one entry: 'Applicant' with 'One to One' extension type and 'FTOS\_PWC\_Applicantid' as the relation attribute. Below this, the 'ALLOWED Attributes' section is also expanded, showing a table with columns: IsReadOnly, Name, and Display Name. The 'IsReadOnly' column has a dropdown set to '(All)'. The table lists various attributes with checkboxes in the 'IsReadOnly' column. The 'Name' column lists attributes like 'Affordable', 'Email', 'FirstName', 'FTOS\_PWC\_Productid', 'Installment', 'LastName', 'LoanValue', 'MobileNumber', 'MonthlyExpenditure', and 'MonthlyIncome'. The 'Display Name' column shows the corresponding display names. At the bottom right, there are pagination controls showing '1' of '2' items.

Name	Extension Type	Relation Attribute
Applicant	One to One	FTOS_PWC_Applicantid

IsReadOnly	Name	Display Name
(All)	Affordable	Affordable
<input type="checkbox"/>	Email	Email
<input type="checkbox"/>	FirstName	First name
<input type="checkbox"/>	FTOS_PWC_Productid	FTOS_PWC_Productid
<input type="checkbox"/>	Installment	Installment
<input type="checkbox"/>	LastName	Last name
<input type="checkbox"/>	LoanValue	Loan value
<input type="checkbox"/>	MobileNumber	Mobile number
<input type="checkbox"/>	MonthlyExpenditure	Monthly expenditure
<input type="checkbox"/>	MonthlyIncome	Monthly income

- Click **Save and reload**. The app data form page displays on the **Data Model** tab which has the Business Entities Extensions section expanded. For more information on how to add Business Entities Extensions, see ["Extend the Data Model"](#) on page 1.

If you want to render data extensions on the digital journey, you need to first register them by adding the entity extension. For more information, see ["Render Custom Data Extensions"](#) on page 276.

- After selecting the entity, select the attributes from it that you wish to use in particular. Click the **Insert existing** button and select from the list the attributes.
- To add virtual attributes, in the third grid of the Data Model, click **Insert pre-existing** and tick the virtual attributes to be added to the form.

8. If you wish to restrict the form's access only to specific attributes in the data model, insert the corresponding values in the **Allowed Attributes** and **Allowed Virtual Attributes** sections. By default, all attributes in the data model are available to the form's context. Once you start populating the allowed attributes sections, the form is restricted from accessing any attributes except the ones explicitly defined in these sections.
9. Click **Save and close** to save your app data form.

## Set the App Data Form Default Type

Click the **General** tab and set the app data form default type by ticking the appropriate checkbox:

- **IsDefault** to set the data form by default on Insert mode.
- **Is Default For Edit** to set the data form by default on Edit mode.

Each entity has a default auto-generated data form which contains all the attributes of the data model you're using. If you want to make changes to the default data form, leave the **Auto Generate Template** checkbox unticked, otherwise on entity updates (e.g., add a new data form attribute), the data form is overwritten and all the updates are lost.

You can set to automatically generate the data form template by ticking the **Auto Generate Template** checkbox and selecting from the Auto Generate Template Type field one of the options available: Inherit, 1 Column, 2 Columns, 3 Columns and 4 Columns.

If **inherit** is selected, the data form layout inherits the value from the entity which is parent for the current entity.

For backwards compatibility, a default auto-generate data form template is available at application level in the web.config file, 1-column data form template.

**NOTE**

If the checkbox is selected, on data form save, the existing data form template is overwritten with the auto-generated one.

## Define Who Has Access to the Form

If your business case requires that the app data form is available to designated roles within your organization, click the Security Roles tab and add the security roles who should have access to them. If no security roles are added here, all users are able to view the app.

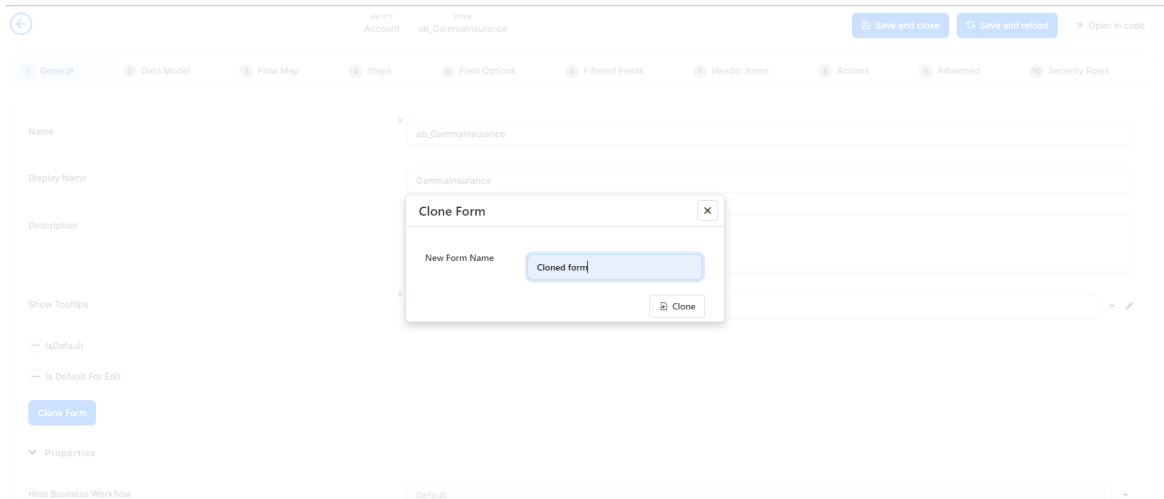
## Clone an App Data Form

To duplicate an app data form with all of its configurations, in the General tab, click **Clone Form**.

The screenshot shows the 'General' tab of the Fintechos Studio interface. The top bar includes tabs for General, Data Model, Flow Map, Steps, Field Options, Filtered Fields, Header Items, Actions, Advanced, and Security Roles. The 'General' tab is active, displaying the following fields and options:

- Name:** ab\_GammaInsurance
- Display Name:** GammaInsurance
- Description:** This is an application for a home insurance.
- Show Tooltips:** User Settings
- IsDefault:** ☐
- Is Default For Edit:** ☐
- Clone Form:** A blue button.
- Properties:**
  - Hide Business Workflow:** Default
  - Read Only:** Default
  - Disable Save Keyboard Shortcut:** Default
  - Show Bullets Progress Bar:** Default
  - Flow Title:** Use Display Name

Insert a name for it and click **Clone**. The new app data form opens immediately for editing.

**NOTE**

The data set behind it is not cloned, simply the form and its settings.  
The Is Default and Is Default For Edit settings are not replicated in the cloned form, since you can have only one default app data form per entity.

For detailed procedures on how to do extensive configuration of app data forms, expand the below sections:

## Steps

Grouping entity information steps, based on specific criteria (business, operational or other relevant to you) is useful especially in complex financial activities when you have to display a lot of information on digital journeys. You can add as many default steps as you need in the Steps section (tab) based on your criteria.

If you only need one section on your journey, add it to the Steps section; it is not marked as step in the Digital Experience Portal.

To avoid issues with the steps loading order, we recommend you to define functions in the journey and only call them within the steps. Rendering a step automatically loads and renders the main journey.

**NOTE**

Before changing the behavior of an element existing on a step by using

logic in a different step, make sure that both steps were previously rendered.

## Field Options

This feature allows you to customize the fields displayed in the user interface and to create rules which apply to specific fields based on how users have filled out other fields in the form driven flow: show field values, show or hide specific fields, etc. For more details on how to configure and customize field options check the [Configure Field Options](#) section.

## Filtered Fields

In the Filtered Fields tab you can control the values that can be used in a field based on the value selected in another field. For example, the Countries and States / Provinces fields. When a user selects a country, you might want to display the states / provinces corresponding to the selected country. Check the [Define Filtered Fields](#) section to get more details on how to add filtered fields.

## Header items

When creating app data forms, the Header Items tab allows you to make the app data form sticky on scroll, which is useful when a data form has many attributes and users have to scroll-down to complete it. In order to see how you can add a header item to an app data form, check the ["Add a new header item" on page 269](#) section.

## Configs

Back office flows allow you to enhance app data forms with dedicated ["Digital Journeys" on page 189](#) that users can trigger from the form on demand. Back-office flows run within the data context of the app data form that initiated them. Once the flow is completed, users are returned to the form at the point where they left off. For more information, see ["Attach a](#)

[Back-Office Flow to an App Data Form" on page 956.](#)

Form actions provide a no-code method for generating a report, run a server side script, run a business decision matrix and call a formula. In order to understand how to create a form action check the "[Define Form Actions](#)" on [page 949](#) section.

FintechOS Studio also allows you to create a custom group of actions that can be triggered on demand on app data forms, when a button is clicked. Check the [Defining Action Groups page](#) for details on how to create the action groups.

## Advanced tab

For advanced scenarios that require elaborate customizations, the **Advanced** tab allows you to attach client-side or server-side JavaScript code to your form. For details, see "[Code Execution Sequence](#)" on [page 301](#).

## App Data Mock-up Forms

App data mock-up forms allow you to design a forms without having an underlying data model. This lets consultants & developers quickly define the general layout of the user interface. Developers can then attach a data model to the mock-up, map entity attributes to the corresponding form fields, and work on any additional back-end configurations.

To create an app data mock-up form, go to **Main Menu > Servicing > Service Extension Apps > App Data Forms** and click **Create Mock-up**.

App data mock-up forms are related to app data forms in the same way form driven mock-up flows are related to form driven flows. See "[Form Driven Mock-up Flows](#)" on [page 280](#) for information on how to create, display, and convert app data mock-up forms into regular app data forms.

# Add and Configure Steps

Breaking down a user experience into manageable steps, allows you to simplify complex tasks based on business or operational considerations.

This section walks you through the configurations that you need to follow to add steps and configure them.

## 1 Add step

Entity Form Steps

+ Insert

X Delete

Export

Refresh

<input type="checkbox"/>	Name	DisplayName	Order <sup>①</sup>	Digital Asset
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	07_VerifyDocuments	Verify Documents	1	
	08_ApproveLoan	Approve Loan	2	

1. On the configuration page of the form, click the **Steps** tab.
2. Click **Insert**.
3. In the General tab, fill in the following fields:
  - **Name** - The name of the step used by system. *We recommend you to use PascalCaseNames (upper camel). The Name starts with an uppercase letter, as do all additional words.*
  - **Display Name** - The name of the step. It is displayed in the Digital Experience Portal.
4. Click **Save and reload**. The step configuration page appears, containing more tabs, displayed by default in the General tab. You can configure the step, by clicking the tabs

and making the desired settings.

## 2 Design the step layout

Click the **UI** tab . Use the "[UI Designer](#)" on [page 323](#) to configure the step's user interface.

## Legacy Template Designer (deprecated)

For steps created in FintechOS Platform versions prior to v.24, the UI tab displays the legacy template designer which, in addition to the UI designer, provides an HTML editor that allows you to configure the step's HTML elements.

Legacy steps also allow you to use the "[Advanced Code Editor \(deprecated\)](#)" on [page 1142](#) or the "[Code Editor](#)" on [page 1108](#) respectively to edit their HTML code directly.

The step's UI template supports the following tokens:

Token	Description
{AttributeName}	<p>Displays the corresponding field on the step.</p> <div> <b>NOTE</b>            The attribute name must be included between curly brackets; otherwise, a simple text will be displayed on the page instead of the actual field.         </div>
{#RelationshipName, view: viewName#}	<p>Generates a view provided by relationship and by view.</p> <p>The viewName is optional and specifies which view to generate. If viewName is not provided, the default view will be displayed on the data form.</p>
{#RelationshipName, view: viewName, editmode:cell#}	<p>Generates a view provided by relationship and by view. This view allows inline editing meaning that you can edit cells one by one directly in the grid, without opening specific records.</p> <div> <b>NOTE</b>            In order to activate inline editing for a specific cell, you must tick the <b>Allow Editing</b> checkbox displayed on the entity view column (View &gt; View columns).         </div>
{#MKT_CampaignResponse_MKT_Campaign,nodelete,noinsert#}	<p>Generates a view provided by relationship and by view, but the <b>Delete</b> and <b>Insert</b> buttons are not displayed on the view. You can apply the same logic (similar tokens) for the Export (noexport) or Refresh (norefresh) if you no longer need them on the view.</p>
{\$ChartName\$}	<p>Generates a chart based on the provided chart name.</p>
{? entityName, view: viewName ?}	<p>Allows you to display a view from another entity. For information on how to use it, see <a href="#">Display View from Another Entity</a>.</p>

In the HTML template, you can link HTML elements (labels) to attributes. For information on how to do it, see ["Link Labels to Attributes \(deprecated\)" on page 270](#).

### 3 Configure event handlers

You can use logic blocks to configure event handlers through an intuitive no-code interface. You can define complex actions and conditional logic that can be triggered by attribute field updates, custom events, or at different stages of a step's execution sequence. For more information, see ["Logic Blocks" on page 313](#).

### 4 Provide the code to be executed after the step is generated (optional)

Click the **Advanced** tab and, in the **After Events** field, provide the code to be executed after the step is generated (opened).

Click **Save and Close**. The new section is displayed in the Entity Form Sections list.

#### NOTE

Variables and functions declared in other steps or in the form driven flow's Advanced section are not visible in the current step.

To access such variables and functions, use the `formData.formScope` object.

### 5 Define who has access to the step

If your business case requires that specific steps are available to designated roles within your organization, click the **Security Roles** tab and add the security roles who should have access to them. If no security roles are added here, all users are able to view the section.

### 6 Save step

Click **Save and close**. Add as many steps as you need, then either continue with the journey configuration, or save the changes by clicking **Save and close**.



## Define Form Actions

Form actions enable you to:

- change a record's business status
- generate a report
- run a server side script
- run a business decision matrix
- call a formula.

Once defined, you can use form actions that can be triggered on-demand, for example by adding a form action to a form button. For more details on how to add the form actions to the buttons, check the ["Buttons" on page 347](#) section.

### Create a Form Action

1. On the configuration page of the form, click the **Configs** tab.
2. At the top of the **Form Actions** section, click **Insert**.
3. In the **Form Action** window:
  - a. Enter a name for the form action.
  - b. Click the Plus (  ) sign next to the execute label to add a command.
  - c. Click the labels in the command to select the desired operands such as change business status from status/ generate digital document/ call custom processor/ call business matrix/ call a formula with mapping. Then, select the status/ the document/ the processor/ the matrix.
  - d. Go back to step b. if you wish to add additional commands. To remove commands from the list, click the  button.
  - e. Click **Save** .

## Available Form Action Commands

Command	Description	Operands	Examples
Change Business Status from Status	Changes the record's workflow status based on the entity's attached business workflow. For details, see the <a href="#">Business Workflows Processor documentation</a> .	<ul style="list-style-type: none"> <li>Initial status</li> <li>Final status</li> </ul>	Change the status from Draft to Active
Generate Digital Document	Generates a predefined report. For details, see <a href="#">"Analytics" on page 1015</a> .	Report name.	Generate a contract or an agreement
Call Custom Processor	Runs a predefined on-demand server automation script. For details, see <a href="#">"Create On-demand Server Automation Scripts" on page 1168</a> .	Server automation script name.	Call the E-sign processor or an endpoint.
Call Business Matrix	Runs a predefined business decision matrix. For details, see the <a href="#">Business Decisions Processor documentation</a> .	Business decision matrix name.	Call the eligibility matrix.
Call Formula with data mapping	Runs a predefined formula for calculation of input data. See <a href="#">"Use Business Formulas in a Digital Journey" on page 521</a> .	Business Formulas	Call the formula for calculating the policy of an insurance.

## Attach an endpoint in Form Action

In app data forms, in Form Actions, a user can map between an automation script input/output parameters and form attributes.

1. Create one Platform Data Entity that is default for a app data form.
2. Create a server script with Input Parameters and the Output structure is one of the following, depending on your needs:
  - [none]
  - "Entity"
  - "Custom"
  - "Boolean".

If the endpoint has no output structure type, then no output mapping form is displayed. For more details, see ["Create On-demand Server Automation Scripts" on page 1168](#).

- 3. In the FintechOS Studio main menu, navigate to **Servicing > Service Extension Apps > App Data Forms**
- 4. Open the **Configs** tab and, in the first grid, click **Insert**.
- 5. Insert a name for the action.
- 6. Click the execute button and select the "Call custom processor".
- 7. Select from the drop-down the endpoint created.
- 8. Map the input attributes.
- 9. Map the output attributes.

Form Action

Action bool endpoint

execute +

✖

Call Custom Processor

⋮

Custom Processor

BD\_DPA\_13086\_04\_endpoint

MAP INPUT PARAMETERS

input\_p1\_bool\_bool

Is In Black List

MAP OUTPUT STRUCTURE

Result type: Boolean

output

Is In Black List

Apply

- 10. Click **Apply**. Click **Save**.

11. Click **Save and close**.

## Define Action Groups

FintechOS Studio allows you to create a custom group of actions that can be triggered on demand on app data forms, when you click a button.

### Prerequisite

- You need to have an on-demand automation script defined on the entity for which you create the app data form.

### 1 Add action group

1. On the **Configs** tab of the app data form, click the **Action Groups** tab.
2. At the top of the Action Groups section, click **Insert** . The Add Action Group page appears. Provide the following properties:
  - **Name** - The name of the action group. Make sure that you use the following naming convention: pascal case, no special characters and no blank spaces.
  - **Display Name** - The name of the action group that is displayed in the Digital Experience Portal.
  - **Entity Form** - Select the data form for which you define the action group.
3. Click **Save and reload** . The Add Action Group page is replaced by the Edit Action Group page and the Actions section becomes available.

### 2 Add endpoints

Add the endpoints on which actions defined in the selected on-demand automation scripts is run on button click.

1. From the Endpoints section, click **Insert** . The Add Endpoint page is displayed. Provide the following action properties:

- **Name** - The name of the endpoint used by the system.
- **Display Name** - The name of the endpoint displayed in the Portal UI.
- **Script** - From the **Script** drop-down, select the on-demand automation script run on button click. If you have no on demand scripts defined, you can add one by clicking the **Script** drop-down and clicking the **Insert** in the page that lists the available scripts and in the Add Scripts page providing the script properties.
- **Executes Save** - Saves after the script execution.
- Click **Save and close**.

The screenshot shows the 'Edit Endpoint' form. It includes the following elements:

- Name:** A text input field containing 'Endpoint1'.
- Display Name:** A text input field containing 'Endpoint1'.
- Action Group:** A dropdown menu showing 'Test'.
- Script:** A dropdown menu showing 'af\_context.result=true'.
- Executes Save:** A checkbox that is currently unchecked.
- Security Roles:** A section with a search bar and a table. The table has one row with the role name 'testrole'.
- Buttons:** '+ Insert existing' and 'X Remove existing' buttons are located to the right of the Security Roles section.

You can add as many endpoints as you need. The endpoints are displayed in order of their index. To change their order index, drag and drop the desired action row.

The figure below shows the actions defined in the Action Groups for the credit bureau interrogation in the user interface.

## Hide the action button

To hide the action button, click the **Advanced** tab, then click the **After Events** tab and in the field type the following JavaScript code:

```
$("#div[data-action-group-name=" <the name of the action group you want to hide>"]).hide();
```

Click one of the save icons to save the changes. The action button is hidden.

## Define Filtered Fields

The **Filtered Fields** tab allows you to dynamically filter the options available in a lookup attribute field based on another field. For instance, you can limit the cities available in a drop-down based on a previously selected country. Thus, you can ensure data accuracy and streamline the user experience when the lookup entities you are referencing are also referencing each other.

For more information, see ["Defining Filtered Fields" on page 266](#).

# Back-Office Flows

Back Office Flows allow you to enhance ["App Data Forms" on page 933](#) with dedicated ["Digital Journeys" on page 189](#) that users can trigger from the form on demand. Back-office flows run within the data context of the app data form that initiated them. Once the flow is completed, users are returned to the form at the point where they left off.

For example, an online banking application based on an app data form might include steps for managing personal information, current accounts, cards, payments, etc. Additionally, a customer may wish to check their credit score. A dedicated digital journey can be attached to the app data form, where the customer consents to personal data processing, provides required details, and queries the Credit Bureau for their credit score. Once the journey is complete, the customer is returned to the banking application.

## Create a Back-Office Flow

1. In FintechOS Studio, go to **Main Menu > Servicing > Back-Office Flows**.
2. Click **+Insert**.
3. Create the desired back-office flow as you would ["Create a Digital Journey" on page 189](#), with the following differences in mind:

## Differences Between Back-Office Flows and Digital Journeys

When editing a back-office flow, you will notice the following differences from the digital journey editor.

- Where appropriate, the tabs for editing the back-office flow reflect the differences in naming. The Analytics are not available for back-office flows.

Digital Journey Tabs	Equivalent Back-Office Flow Tabs
Journey Designer	Back-Office Flow Designer
General	General
Digital Journey Map	Back-Office Flow Map
Digital Journey Navigation Rules	Back-Office Flow Navigation Rules
Analytics	n/a

- In the General tab:
  - The **Exposed API Journey** option is renamed to **Exposed API**.
  - The **Digital Journey Flows** section is renamed to **Subflows**.
  - The **Digital Journey States** section is not available.
- The "Journey Settings" on page 192 only include options for the Actor.

### Doc 002 BOF



Actor

Who will be using this journey

Branch Employee x

Select or type to add an actor

## Attach a Back-Office Flow to an App Data Form

1. In FintechOS Studio, go to **Main Menu > Servicing > App Data Forms**.
2. In the **App Data Forms** grid, double click the desired form.
3. Open the **Configs** tab.
4. In the **Back-Office Flow** section, click **+ Insert**.

5. Select the desired **Back-Office Flow**. Optionally, add a **Description** for reference.

### IMPORTANT!

Since back-office flows run within the data context of the app data form that initiated them, the app data form and the associated back-office flows must use the same source entity for their data models. If a back-office flow contains multiple subflows, only the first subflow must use the same source entity as the app data form.

6. Click **Save and reload**.
7. Optionally, you can use the **Availability** tab to define restrictions for who can run the back-office flow and when.
  - Provide a **Name** and **Description** for the availability rule.
  - Use the point-and-click **Rule Expression** designer to define boolean logic for the rule.

← Save and close Save and reload

Edit App Data Form X Backoffice Flow Rule

Name User Manager

Description Access restrictions based on user role and entity status

Rule Expression

And +

× Current User Has any Role Registered Users, User Admin, User Management, Security Role Management

× Status Equals Active

### IMPORTANT!

Users who will initiate the back-office flow require **"Security Roles"** on page 1245 with security items that grant them access to the

**BackOfficeFlowExecution** entity (in addition to the flow's source entity).

- You can set up multiple availability rules for the same back-office flow.

8. Click **Save and Close**.

You can repeat the process above to attach multiple back-office flows to your app data form.

## Run a Back-Office Flow from an App Data Form

1. In FintechOS Portal, open the desired App Data Form.
2. Make sure the form is in the Edit mode (if you are inserting a new record, click **Save and Reload** at least once).
3. From the ellipsis menu (...), select **Show Flows**. This displays a list of back-office flows associated with the form that you are authorized to run.
4. In the **Flows** tab, use the **Start** button to initiate a flow or the **Continue** button to resume an interrupted flow.  
In the **History** tab, you can review the previously run or currently running flows.
5. Proceed with the back-office flow. Once the flow is finished, you will be returned to the app data form.

# Ecosystem Hub

FintechOS Ecosystem Hub offers access to a variety of connectors that you can use in your customer journeys. Use API connectors to connect to external APIs where you could for instance, use a credit reporting agency's API to automatically check the creditworthiness of an applicant.

## External Services via Service Pipes

FintechOS Service Pipes are the integration layer of the FintechOS Platform. They use [Apache Camel](#) as a routing and mediation engine to integrate the FintechOS Platform with external systems. Apache Camel is an integration framework that allows easy implementation of routing and mediation logic using a variety of domain-specific languages (DSLs). Read more about this topic on the dedicated [Call External Services via Service Pipes](#) page.

## Web API Client Libraries

[Web API client libraries](#) allow you to work with external APIs using proxy methods native to the FintechOS Platform development environment. Web API client libraries are generated automatically by FintechOS Studio from OpenAPI specification files supplied by the web service provider. Once generated, you can import the library into any server automation script that requires access to the API.

Some of the advantages of Web API client libraries are:

- less code to write and maintain
- simplified authorization and authentication
- better code consistency and robustness

## External APIs

External APIs allow you to access third party resources that are available through RESTful web services. In addition to defining a sequence of API calls, some of the External API's advanced features include:

- Manage the API calls' paths and authentication in a centralized and secure mode, from the *web.config* file.
  - Rules for the execution sequence of the API calls, such as canceling subsequent calls in the sequence in case of failure or establishing dependencies where one call only executes after the successful completion of other calls.
  - Custom JavaScript code to be executed before or after a call.
  - Call timing configurations such as expiration time of the authentication token, retries on error, number of retries, or delays between retries.
  - Logging for the API calls' requests and responses.
- 

## Web API Client Libraries

Web API client libraries allow you to work with external APIs using proxy methods native to the FintechOS Platform development environment. Web API client libraries are generated automatically by FintechOS Studio from OpenAPI specification files supplied by the web service provider. Once generated, you can import the library into any server automation script that requires access to the API.

### HINT

You can access the API specification files of a FintechOS Platform instance at the following location: `<host_address>/ftosapi/swagger/index.html`

## Create a Web API client library from an OpenAPI specification file

1. In the Main Menu, go to **Ecosystem > Web API Client Libraries**.
2. In the Web API Client Libraries List page, click **Insert**.
3. Fill in the library's details.

The screenshot shows the 'General' tab of a form for creating a Web API Client Library. The form has two tabs: 'General' (selected) and 'Typescript Definition'. The 'General' tab contains the following fields:

- Name:** A text input field containing 'AC\_OpenAPI'.
- Description:** A large text area for optional library description.
- Api Type:** A dropdown menu set to 'OpenApi'.
- Min Platform Version:** A text input field containing '20.2'.
- Is Used For Service Pipes:** A checkbox that is currently unchecked.
- Api Definition:** A large text area containing a JSON OpenAPI specification snippet. The snippet includes metadata like 'openapi: 3.0.1', 'title: FintechOS API', and a path for '/fiosapi/authentication/keycloakToken' with a 'post' method and 'Authentication' tags.

- **Name** - Enter a name for the Web API client library.
- **Description** - Optional library description.
- **Min Platform Version** - Optionally select the oldest API version endpoints you wish to extract from the source file.
- **Use Service Account Token** - Select if the library is used to ["Call External Services via Service Pipes"](#) on page 967, ["Call an Asynchronous Flow via API"](#) on page 1190, or ["Products APIs"](#) on page 904. This will automatically manage authorization when a server automation script invokes the library. Otherwise,

you will have to add the authorization code in your scripts (e.g.: "[Create a Server Automation Script that Invokes the Web API Client Library](#)" on page 971).

- API Definition - Paste the contents of the OpenAPI specification file.
4. Click **Save and Reload**. After the library is generated, you can review its definitions in the Typescript Definition tab.
  5. Click **Save and Close**.

## Use a Web API client library in server automation scripts

In your automation script, use the **importWebApiClient** function to import the Web API client library in a JSON object that contains the API specifications. The object exposes methods matching the API's endpoints and has full IntelliSense auto-complete support.

For details, see the [Server SDK Reference Guide](#).

## Add certificate support to WebApi client and WCF client

WebAPI client offers a way for consuming external web services by providing a definition for the external service with an OpenAPI file and import it into the platform and provide an object to use. This feature allows a user to work with web services and web sites that require a certificate authentication for example Certsign.

### HINT

To do so, the user needs a valid certificate.

```

1 let client = importWebApiClient('SecureFTOS', setCertificate(workflowClientCertificate:
2     I WorkflowClientCertificate): void
3 client.setCertificate(server.clientCertificates.get());
4
5 try {
6     var authToken = client.authorize.getToken({
7         client_id: 'client_id',
8         username: 'host',
9         password: '1234567'
10    });
11    if (authToken && authToken.access_token) {
12        let data = client.openApi.query({
13            apiInfo: {
14                userName: "host",
15                token: authToken.access_token
16            },
17            request: {
18                entity: {
19                    name: "webApiClientLibrary",
20                    alias: "api"
21                },
22                distinct: false
23            }
24        });
25        log(data);
26    }
27    else
28        throw new Error('Invalid authentication!');
29 }
30 catch(err) {
31     log(err);
32     throw err;
33 }

```

There is a method for this process setCertificate which accepts workflow client certificate, this is to actually use a certificate. By using a server method that provides the certificate, this method uses the configurations from web.config. In the web.config file, the certificate needs to be configured by providing a storeLocation, storeName, thumbPrint.



```

4 http://go.microsoft.com/fwlink/?linkid=301880
5 -->
6 <configuration>
7 <configSections>
8 <sectionGroup name="system.web.webPages.razor" type="System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup, System.Web.WebPages.Razor, Version=3.0.0.0, Cu
9 </sectionGroup>
10 <section name="pages" type="System.Web.WebPages.Razor.Configuration.RazorPagesSection, System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral, PublicKeyTo
11 </section>
12 <section name="loggingConfiguration" type="Microsoft.Practices.EnterpriseLibrary.Logging.Configuration.LoggingSettings, Microsoft.Practices.EnterpriseLibrary.Log
13 </section>
14 <section name="clientDependency" type="ClientDependency.Core.Config.ClientDependencySection, ClientDependency.Core" requirePermission="false" />
15 <section name="httpCookies" type="System.Web.Configuration.HttpCookiesSection, System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
16 <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKey
17 </section>
18 <section name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup, System, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c5619
19 </section>
20 <section name="EBS.Core.Web.MVC.Properties.Settings" type="System.Configuration.ClientSettingsSection, System, Version=4.0.0.0, Culture=neutral, PublicKeyToken
21 </sectionGroup>
22 <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?linkid=237468 -->
23 <sectionGroup name="bundleTransformer">
24 <section name="core" type="BundleTransformer.Core.Configuration.CoreSettings, BundleTransformer.Core" />
25 <section name="sassAndScss" type="BundleTransformer.SassAndScss.Configuration.SassAndScssSettings, BundleTransformer.SassAndScss" />
26 </sectionGroup>
27 </configSections>
28 <appSettings>
29 <add key="feature-development-mode" value="true" />
30 <add key="automation-client-certificate-myCert1" value="{ 'storeName': 'My', 'storeLocation': 'CurrentUser', 'thumbPrint': '40ca5cf013e50cf230ba52242d565c8d1686889' }" />
31 <add key="automation-client-certificate-FTOSWild2021" value="{ 'storeName': 'My', 'storeLocation': 'CurrentUser', 'thumbPrint': '728383137e288cb672d1679b63a50c160da5f2' }" />
32 </appSettings>
33 <!-- MVC project settings DO NOT Modify -->
34 <add key="webpages:Enabled" value="false" />

```

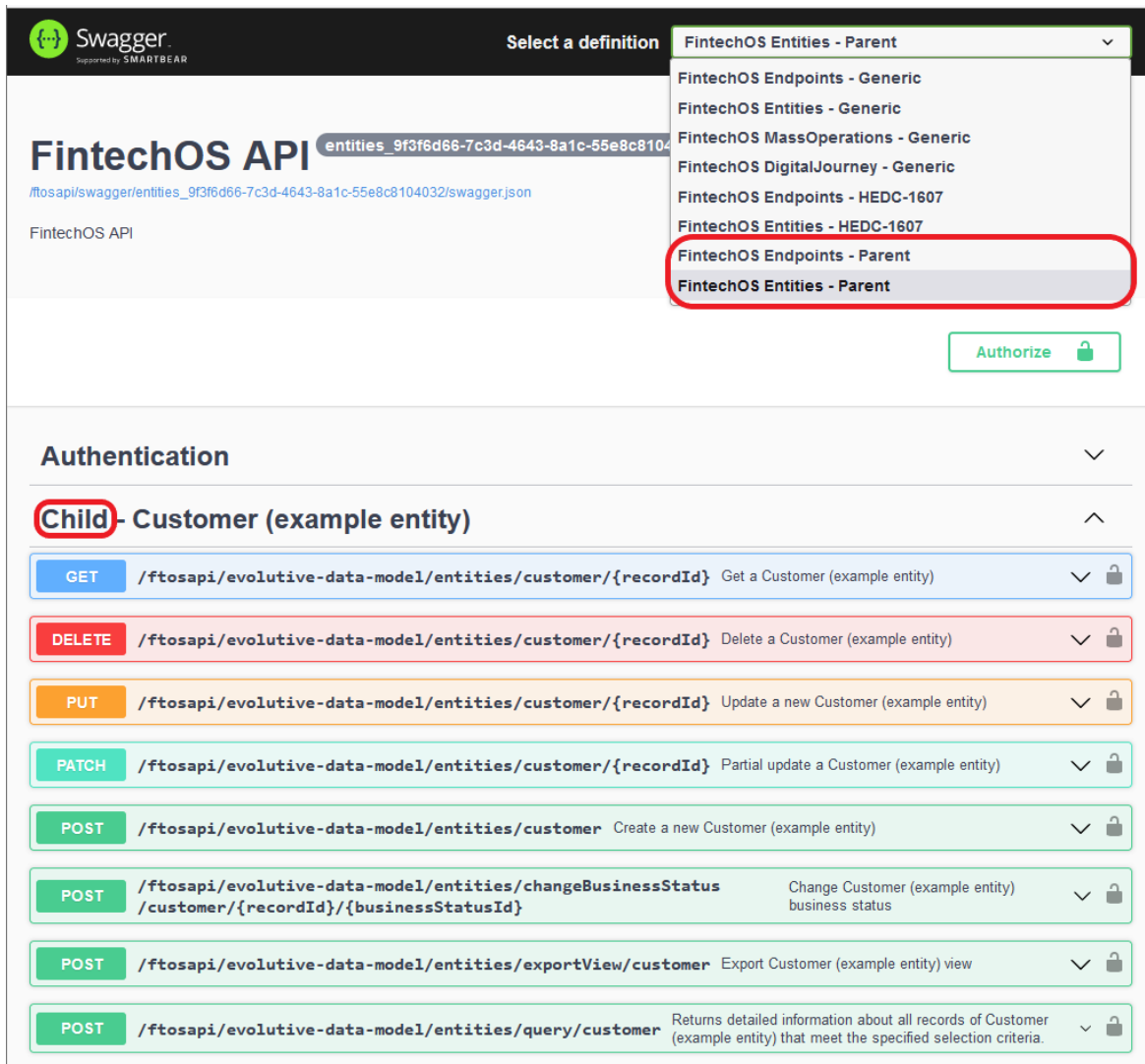
# OpenAPI Domains

OpenAPI domains allow you to organize your endpoint and entity APIs for better usability. You can assign endpoints and entities to the same domain to group them together and you can define parent - child relationships between domains to create API hierarchies that are easy to understand and navigate.

**IMPORTANT!**

A single level of children domains is supported. A child domain cannot be a parent for other domains.

In the Swagger interface (*<environment URL>/ftosapi/swagger* by default), two API definitions are created automatically for each root domain: one for the domain's endpoints and one for the domain's entities, while subdomains are prepended at the beginning of the API names.



Generic definitions include digital journeys, mass operations, and endpoints/entities that are not assigned to a domain.

## Create an OpenAPI Domain

1. In FintechOS Studio, go to **Main Menu > Ecosystem > OpenAPI Domains**.
2. Click **Insert**.

3. Fill in the domain's details:

- **Name** - Unique identifier for the domain in the FintechOS Platform.
- **Domain Name** - Domain name as displayed in the user interface.
- **Description** - Short description for the domain.
- **Parent Domain** - Select a parent domain if you wish to make this a subdomain of an existing domain.

4. Click **Save and close**.

## Assign an Entity to an OpenAPI Domain

1. In FintechOS Studio, go to **Main Menu > Ecosystem > OpenAPI Domains**.
2. In the Open API Domains list, double click the desired domain.
3. In the domain editor, scroll down to the **Open API Domain Entities** section.
4. Click **Insert**.
5. In the Add Open API Domain Entity page, use the **Entity** drop-down to select the desired entity.
6. Click **Save and close**.
7. Click **Save and close**.

## Assign an Endpoint to an OpenAPI Domain

1. In FintechOS Studio, go to **Main Menu > Ecosystem > OpenAPI Domains**.
2. In the Open API Domains list, double click the desired domain.
3. In the domain editor, scroll down to the **Open API Domain Endpoints** section.

4. Click **Insert**.
5. In the Add Open API Domain Endpoint page
  - Use the **Endpoint** drop-down to select the desired endpoint.
  - Use the **Http Verb** field to indicate the HTTP request method used to call the endpoint, such as GET, PUT, POST, etc.
6. Click **Save and close**.
7. Click **Save and close**.

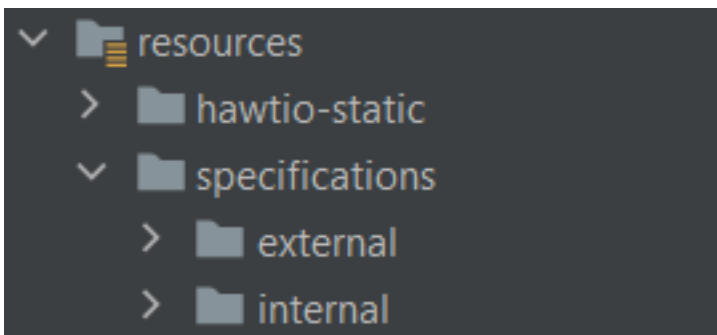
## Call External Services via Service Pipes

FintechOS Service Pipes expose integrations as OpenAPI specifications for endpoints that the FintechOS Platform can consume.

To call an external service from a server side script via Service Pipes, make sure the FintechOS Service Pipes are configured on your environment and follow the instructions below:

### Set up the endpoint specifications in your Service Pipes project

To ensure consistency, set up the project specifications for the endpoints you wish to expose or invoke based on the following structure.



- `resources/specifications/external` - Use the `external-openapi.yaml` file for specifications of endpoints exposed by the Service Pipes to external consumers.
- `resources/specifications/internal` - Use the `internal-openapi.yaml` file for specifications of Service Pipes endpoints used to invoke the FintechOS Platform and that the FintechOS Platform will consume.

For example:

- An **outbound pipe** that calls an external service will expose an endpoint based on a specification located in the **internal** directory (for the FintechOS Platform to invoke).
- An **inbound pipe** that receives a call from an external system and then invokes the FintechOS Platform will expose a Service Pipes endpoint based on a specification in the **external** directory.

## Create routes and mappings directories

In your Service Pipes project, create the following directories:

- `resources/routes` - For Service Pipes defined using XML DSL (if XML is used instead of Java DSL)
- `resources/mappings` - For your transformation files.

## Generate the data model artifacts for your Service Pipes project

Use Maven plug-ins to generate the data model based on your OpenAPI specifications. You can use the command line to run the Maven command, e.g.:

```

mvn camel-restdsl-openapi:generate-with-dto -
Dgenerator
.
openapi
.spec=${basedir}\src\main\resources\spec\external\<specname>.yaml -
Dgenerator
.
model
.package=com.fintechos.servicepipes.<projectName>.model.external -
Dgenerator
.
route
.package=com.fintechos.servicepipes.<projectName>.<packageName> -
Dgenerator.route.className=<className>

```

## Create the Service Pipes (routes)

Use one of the Domain Specific Languages (DSL) supported by [Apache Camel](#) to create your routes. The most used DSLs are Java and XML DSL.

To set up an [endpoint API](#), [entity API](#), or [digital journey API](#) as a route destination, use the following route properties:

```

.setProperty("requestType", constant("endpoint"))
.setProperty("resource", constant("MyEndpoint"))
.to("direct:callFTOSWithDefaultAccount")

```

Where:

- requestType - Is set to either `endpoint`, `entity`, or `digitalJourney`.
- resource - Is the name of the corresponding endpoint, entity, or digital journey.

## Use Cloud Configuration Manager properties in Service Pipes

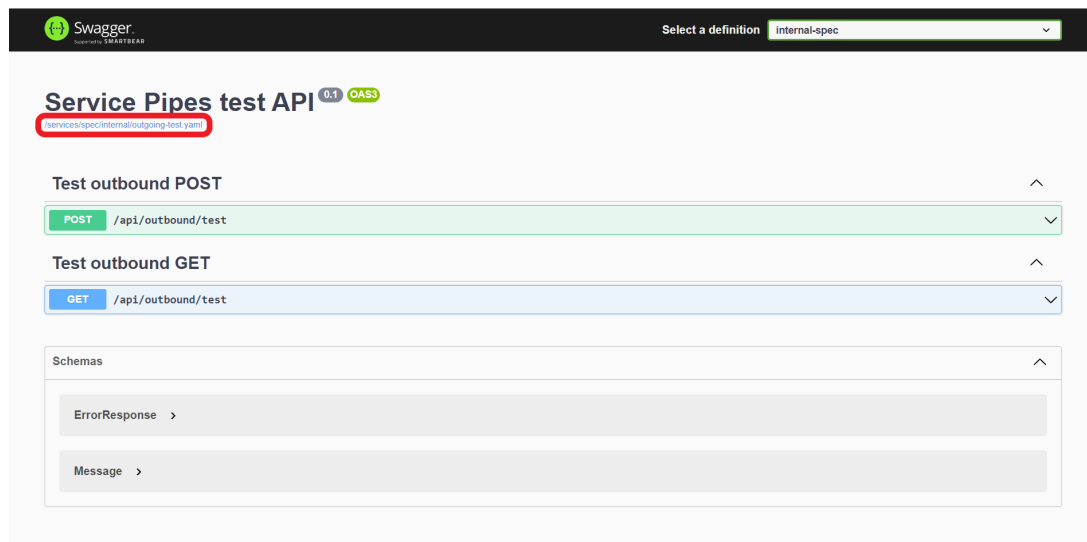
To refer endpoints, credentials, or other configurations stored in the Cloud Configuration Manager in your Service Pipes, use the `addExchangeProperty` method to retrieve them from the Service Pipes cache (`vaultCache`) add them to the `exchange` object properties.

For example, to retrieve the configuration parameter, add it to the *exchange* properties, and use it in the route code to call an endpoint with the *.toD* pattern:

```
@Override
public void configure() throws Exception {
    from(uri: "direct:pipe2")
        .bean("cachingService", "addExchangeProperty('vaultCache',
            ${exchange.getProperties()})")
        .log("exchangeProperty: ${exchangeProperty}")
        .toD("${exchangeProperty}");
}
```

## Import the Endpoint Specifications File in a Web API Client Library

1. Download the desired .yaml endpoint specifications file. You can use the Service Pipes's Swagger web app interface to download the file using a web browser.



2. Import the .yaml file in a [Web API Client Library](#).

## Create a Server Automation Script that Invokes the Web API Client Library

```
var client = importWebApiClient
(
  "ServicePipes"
  , "https://myServicePipes.azurewebsites.net/services");

// For releases prior to v22.1.2 choose either Option 1 or Option 2
// for authorization.

// Option 1 (prior to v22.1.2): use current user OIDC token
// var token = getCurrentUserAccessToken();
// client.setCustomHeader("Authorization", "Bearer " + token);

// Option2 (prior to v22.1.2): use predefined username/password
// var username = getAppSetting("username");
// var password = getAppSetting("password");
// var credentialsEncoded = server.convert.toBase64StringFromAscii
// (username+' '+password);
// client.setCustomHeader("Authorization", "Basic " +
// credentialsEncoded);

// Optional: You can add any context properties
// that are to show up in the service pipes logs
// var contextString = [{"customPropertyId": "F1AC195B-EA39-49C9-
// 8750-BD59921E64F4"}];
// client.setCustomHeader("x-context-properties", contextString);

var result = client.callTestGet();
```

For authorization, depending on your setup, you can either:

- **(v22.1.2 or later)** Select the **Use Service Account Token** option when configuring the Web API Client Library. This automatically manages the authorization. For details, see ["Create a Web API client library from an OpenAPI specification file" on page 961.](#)
- **(prior to v22.1.2)** Use the [FintechOS Identity Provider](#) authentication (in which case you can use the [getCurrentUserAccessToken](#) function to retrieve your access token).

- **(prior to v22.1.2)** Use the legacy username/password authentication (in which case you need to use the *username* and *password* system parameters to retrieve the Service Pipes credentials).

You can call the server automation script from other FTOS components using the methods available in the Client SDK. For example, to test it from the FintechOS Portal using the browser developer tools, you can use the [ebs.callActionByNameAsync](#) method:

```
ebs.callActionByNameAsync('outboundPipe', {  
  "Id": "282772",  
  "statusName": "status Name",  
  "statusReason": "status Reason",  
  "customer": "customer name"  
}).then(function(e) {  
  console.log(e)  
})
```

## Additional Information

For more information about working with Service Pipes, see the [Service Pipes tutorial](#).

## External APIs

You can take advantage of the External API on the client side and the FintechOS API endpoints on the server side to exchange data between different FintechOS Platform instances. For more information about FintechOS API resources, see the [API Reference Guide](#).

Use External APIs to access third party resources that are available through RESTful web services.

---

## Create, Configure, and Call External APIs

External APIs are predefined collections of API calls that are executed in sequence within the same context, effectively turning FintechOS Studio into an API client with advanced capabilities.

### Create External APIs

1. Open FintechOS Studio in Developer mode and navigate to the **Main Menu**.
2. Select **Ecosystem > External APIs**. This opens the list of external APIs.
3. Click **Insert** to add a new External API. This opens the Add External API page.
4. In the page that opens, enter a **Code** and, optionally, a **Name** for your External API.
5. Click **Save and Close**.

This creates an empty External API. To add API calls to the External API, you need to configure the External API.

### Configure External APIs

1. Open FintechOS Studio in Developer mode and navigate to the **Main Menu**.
2. Select **Ecosystem > External APIs**. This opens the External API List page.
3. In the External API List, double click the External API you wish to edit. This opens the Edit External API page.
4. In the External API Details grid:
  - **To add a new API call to the pipe**, click the Insert button.
  - **To edit an existing API call**, double click the API call from the grid.
5. In the page that opens, fill in the API call's details and click the **Save and Close** button at the top right corner of the page when done. For details on how to set up an API call, see the following:

## External API API Call - Settings

Setting	Description
Code	API call's code. This setting is filled automatically based on the following schema [ <i>External API code</i> ].[ <i>API call Order No.</i> ]. For details, see " <a href="#">Create, Configure, and Call External APIs</a> " on the previous page and " <a href="#">OrderNo</a> " below.
Name	Enter a descriptive name for API call.
OrderNo	The order in which the API call is executed in the pipe. This allows you to configure multiple API calls to be executed in sequence in the same External API.

Setting	Description
baseUrllsKeyInConfigFile	<p>Check to use the base URL defined in the FintechOS Studio <i>web.config</i> file for the API call's path ([base URL]+[endpoint]).</p> <div> <p><b>NOTE</b></p> <p>Checking this option disables the "BaseURL" on the next page field.</p> </div> <p>To configure a base URL in the <i>web.config</i> file, open the <i>web.config</i> file in a text editor and add the following entry in the &lt;appSettings&gt; section:</p> <div> <pre>&lt;add key="FTOS_IntegrationProcessBaseUrl_[External API Detail Code]" value=[base URL]/&gt;</pre> </div> <p>The External API's code and the API call's order number in the External API detail code are optional. For example:</p> <ul style="list-style-type: none"> <li>To use a base URL for the first API call in the P04 External API, enter: <div> <pre>&lt;add key="FTOS_IntegrationProcessBaseUrl_P04.01" value="http://www.example.com/example_Api"/&gt;</pre> </div> </li> <li>To use a base URL for all the API calls in the P04 External API, enter:</li> </ul>

Setting	Description
	<pre>&lt;add key="FTOS_ IntegrationProcessBaseUrl_ P04"  value ="http://www.example.com/example_ Api"/&gt;</pre> <ul style="list-style-type: none"> <li>To use a base URL for all the API calls of all the External API, enter:</li> </ul> <pre>&lt;add key="FTOS_ IntegrationProcessBaseUrl"  value ="http://www.example.com/example_ Api"/&gt;</pre> <p>If multiple base URL definitions in the <i>web.config</i> file are applicable for the same API call, the first one will be applied.</p>
BaseUrl	Base URL part of the API call's path ([base URL]+[endpoint]).
MethodName	Endpoint part of the API call's path ([base URL]+[endpoint]).
HttpMethod	HTTP request method used by the API call. Available options are: GET, POST, PUT, and DELETE.
hasHttpAuthentication	Check to make " <a href="#">httpAuthenticationType</a> " on the next <a href="#">page</a> mandatory.

Setting	Description
httpAuthenticationType	<p>Allows you to select <b>Basic</b> HTTP authentication based on user-id/password pairs (RFC7617) or <b>Bearer</b> tokens based authentication (RFC6750).</p> <p>To configure a user-id/password authentication scheme, open the FintechOS Studioweb.config file in a text editor and add the following entries in the &lt;appSettings&gt; section:</p> <pre>&lt;add key="FTOS_IntegrationProcessAuthUser_[External API Detail Code]" value="[user-id]"/&gt; &lt;add key="FTOS_IntegrationProcessAuthPassword_[External API Detail Code]" value="[password]"/&gt;</pre> <p>The External API's code and the API call's order number in the External API detail code are optional. For example:</p> <ul style="list-style-type: none"> <li>To set credentials for the first API call in the P04 External API, enter: <pre>&lt;add key="FTOS_IntegrationProcessAuthUser_P04.01" value="guest"/&gt; &lt;add key="FTOS_IntegrationProcessAuthPassword_P04.01" value="guest"/&gt;</pre> </li> <li>To set credentials for all the API calls in the P04 External API, enter: <pre>&lt;add key="FTOS_IntegrationProcessAuthUser_P04" value="guest"/&gt;</pre> </li> </ul>

Setting	Description
	<pre>&lt;add key="FTOS_ IntegrationProcessAuthPassword_ P04" value="guest"/&gt;</pre> <ul style="list-style-type: none"> <li>To set credentials for all the API calls of all the External API, enter:</li> </ul> <pre>&lt;add key="FTOS_ IntegrationProcessAuthUser" value="guest"/&gt; &lt;add key="FTOS_ IntegrationProcessAuthPassword" value="guest"/&gt;</pre> <p>If multiple credentials in the <i>web.config</i> file are applicable for the same API call, the first one will be applied.</p>
additionalHeaders	Code for additional configurations to include in the HTTP request header.
StopOnError	When checked, if an error occurs during the API call, the remaining API calls in the sequence will not be executed. Otherwise, the External API will attempt to run the next API call in the sequence.

Setting	Description
beforeJS	<p>Custom JavaScript code to be executed before the API call. For details, see <a href="#">External API API Call – Custom JavaScript Reference</a>.</p> <div> <p><b>IMPORTANT!</b></p> <p>When this page is saved/refreshed for the first time, the following code is automatically added in the beforeJS text box:</p> <pre> var instanceId = "\$\$instanceId\$\$"; var integrationProcessDetailId = "\$\$integrationProcessDetailI d\$\$"; var contextEntityName = "\$\$contextEntityName\$\$"; var contextUniqueId = "\$\$contextUniqueId\$\$"; var runAsync = "\$\$runAsync\$\$"; var requestParamsBeforeJs = {}; requestParamsBeforeJs = \$\$requestParamsBeforeJs\$\$;  /*set token example*/ /*requestParamsBeforeJs ["TOKEN"] = getAuthorizationTokenFromInt egrationProcessDetailId (instanceId, integrationProcessDetailI d);*/ </pre> </div>

Setting	Description
	<div>////mandatory return object return {requestParams: requestParamsBeforeJs, skippedFromBeforeJs: false };</div>

Setting	Description
afterJS	<p>Custom JavaScript code to be executed after the API call. For details, see <a href="#">External API API Call – Custom JavaScript Reference</a>.</p> <div> <p><b>IMPORTANT!</b></p> <p>When this page is saved/refreshed for the first time, the following code is automatically added in the afterJS text box:</p> <pre> var responseAsString = ""; responseAsString = \$\$responseAsString\$\$; var instanceId = "\$\$instanceId\$\$"; var instanceDetailId = "\$\$instanceDetailId\$\$"; var contextEntityName = "\$\$contextEntityName\$\$"; var contextUniqueId = "\$\$contextUniqueId\$\$"; var runAsync = "\$\$runAsync\$\$";  var responseAsJson = JSON.parse (responseAsString); log("responseAsJson:" + toJson(responseAsJson));  /*save token example*/ /* if(responseAsJson.TOKEN){ </pre> </div>

Setting	Description
	<pre> update("FTOS_ IntegrationProcessInstanceDe tail", instanceDetailId, {authorizationToken: responseAsJson.TOKEN, authorizationTokenObtainedO n: responseAsJson.TOKEN_ DATE}); }else{ resultAsjson.isSuccess = true; resultAsjson.message = "Nu a putut fi obtinut tokenul:" + (responseAsJson.errorMessage ?responseAsJson.errorMessage :null"); } */ ////mandatory return object var resultAsjson = {}; resultAsjson.isSuccess = true; resultAsjson.message = "!OK!"; return resultAsjson; </pre>
expiresInSecondsMultiplier	The number of seconds after which the authentication token expires.
secToWaitBeforeStart	Configures a delay in seconds before running the API call.
retryOnError	Check to resend the API request if the API call fails (the <code>resultAsjson</code> object has the <code>isSuccess</code> key set to <code>false</code> ). The number of retries is set by the <code>"numberOfRetries"</code> on the next page setting.

Setting	Description
numberOfRetries	Number of attempts to resend an API request if the API call fails (if the <a href="#">"retryOnError"</a> on the previous page setting is enabled).
secToWaitBeforeRetry	Number of seconds to wait before resending an API request if the API call fails (if the <a href="#">"retryOnError"</a> on the previous page setting is enabled).
Pass response in main result	Includes the API call's response in the response object of the External API. Otherwise, it will be available only within the External API's context in the responseAsJson variable in the <a href="#">"afterJS"</a> on page 981 setting.
logRequest	Logs the call's HTTP request details in the FTOS_IntegrationProcessInstanceDetailLog entity for debugging purposes.
logResponse	Logs the call's HTTP response details in the FTOS_IntegrationProcessInstanceDetailLog entity for debugging purposes.
External API Detail Parameters	Defines the parameters that will be passed to the API endpoint. For details, see <a href="#">External API API Call – Parameters</a> .

Setting	Description
External API Detail Dependencies	<p>Defines the API calls that must be completed successfully prior to running the current API call.</p> <div> <p><b>IMPORTANT!</b></p> <p>If the dependency is based on a security context where the dependee call must pass an authorization token to the current API call, check the <code>hasAuthorizationToken</code> setting and add the following code in the <a href="#">"beforeJS" on page 979</a> textbox:</p> <pre>requestParamsBeforeJs [ "TOKEN" ] = getAuthorizationTokenFromInt egrationProcessDetailId (instanceId, integrationProcessDetailId);</pre> </div>

## External API API Call - Parameters

This screen configures each parameter that will be populated and passed by the API call to the REST API web service. Parameter settings must match the specifications of the called REST API.

Parameter Setting	Description
IntegrationProcessDetail	API call's code (see <a href="#">Code</a> ).

Parameter Setting	Description
ParameterType	<ul style="list-style-type: none"> <li>• Query – Parameter is appended to the API call's path.</li> <li>• Body – Parameter is passed in the HTTP request body.</li> <li>• Formdata – Parameter is passed using the multipart/form-data media type.</li> </ul>
isMandatory	Parameter is mandatory.
isEnumList	Check if the Body type parameters are passed in the HTTP request in enumeration format (a single line JSON such as {firstName: "John", lastName: "Doe"}).
Name	Parameter name.

Parameter Setting	Description
IsAuthenticationUser	<p>Applicable for all non-Body parameters with "isEnumList" on the previous page not checked.</p> <div> <p><b>IMPORTANT!</b></p> <p>Do not set user-id static parameters in this screen.</p> <p>See <a href="#">httpAuthenticationType</a> for details on how to configure authentication credentials.</p> </div>
IsAuthenticationPassword	<p>Applicable for all non-Body parameters with "isEnumList" on the previous page not checked.</p> <div> <p><b>IMPORTANT!</b></p> <p>Do not set password static parameters in this screen.</p> <p>See <a href="#">httpAuthenticationType</a> for details on how to configure authentication credentials.</p> </div>

## External API API Call - Custom JavaScript Reference

You can use the [beforeJS](#) and [afterJS](#) settings to set up custom JavaScript code to be executed before and after an API call. The following objects are available in this context:

Object	Description
instanceId	ID of the External API instance.
integrationProcessDetailId	ID of the API call.
contextEntityName	Name of the entity for which the External API is run.
contextUniqueid	ID of the <a href="#">"contextEntityName"</a> above record.
runAsync	Either true or false depending if the External API is run asynchronously or not.
requestParamsBeforeJs	Includes static parameters and their values to be passed to the HTTP request.

Object	Description
requestParams	<p>Parameter - value pairs to be passed to the API call. You can edit this object to include "requestParamsBeforeJs" on the <a href="#">previous page</a> and/or dynamic parameters, such as an authentication token obtained in a prior step or values from the "contextEntityName" on the <a href="#">previous page</a> record.</p> <div> <b>IMPORTANT!</b>            This is a mandatory object that must be returned by the <a href="#">beforeJS</a> code.         </div>
skippedFromBeforeJs	<p>If set to true, the API call is skipped. Otherwise, it should be set to false.</p> <div> <b>IMPORTANT!</b>            This is a mandatory object that must be returned by the <a href="#">beforeJS</a> code.         </div>
responseAsString	HTTP response of the API call in string format.

Object	Description
resultAsjson	<p>Indicates if the API call was successful (resultAsjson.isSuccess = true;) and the response message (resultAsjson.message = "!OK!");).</p> <div> <p><b>IMPORTANT!</b></p> <p>This is a mandatory object that must be returned by the <code>afterJS</code> code.</p> </div>

## Call External APIs

To call a External API, use the `FTOS_IntegrationProcessLibrary` object and `callIntegrationProcess` method:

```
var integrationProcessId = FTOS_
IntegrationProcessUtils.getIdFromCode("FTOS_IntegrationProcess",
"code", restPipeCode);
var ip = new FTOS_IntegrationProcessLibrary();
ip.logEnabled = false;
var response = ip.callIntegrationProcess(integrationProcessId,
contextEntityName, contextUniqueId, requestParams, runAsync);
```

where:

Parameter	Description
integrationProcessId	External API ID based on the pipe's code (see <a href="#">"Create, Configure, and Call External APIs" on page 973</a> for details).
contextEntityName	Name of the entity for which the External API is run.
contextUniqueId	ID of the <a href="#">"contextEntityName" on page 987</a> record.

Parameter	Description
requestParams	<p>Includes static parameters and their values to be passed to each API call in the following format:</p> <pre>requestParams[External API code].[API call Order No.] = {[parameter 1]:[value 1], [parameter 2]:[value 2], ...};</pre> <p>For example:</p> <pre>requestParams["P01.01"] = {CUI: "36438401", CNP: "78787878"}; requestParams["P01.02"] = {CUI: "36438401", CNP: "78787878"};</pre> <p>These parameter values will be available in the API call's custom JavaScript code in the <code>requestParamsBeforeJs</code> object.</p>
runAsync	If set to true, runs the External API as an asynchronous process.
response	<p>The resulting response of the External API call will have the following format:</p> <pre>response:{   "ipInstanceId": "523ee20b-705c-45b3-b881-caeec1bde15e",   "isSuccess": true,   "errorMessage": null,   "mainResponse": {     "P04.02": {       "requestId": 89696.0,       "errorMessage": null,       "IsSuccess": true     }   } }</pre>

## Example

```
var ip = new FTOS_IntegrationProcessLibrary();

var integrationProcessId = FTOS_
IntegrationProcessUtils.getIdFromCode("FTOS_
IntegrationProcess", "code", "P04");
```

```

var requestParams = {};
requestParams["P01.01"] = {CUI: "36438401", CNP:
"78787878"};
requestParams["P01.02"] = {CUI: "36438401", CNP:
"78787878"};

ip.logEnabled = true;
var response = ip.callIntegrationProcess
(integrationProcessId, "P666", "31a85d94-0c9c-429b-8bc6-
8c4a5e2d91a7", requestParams, false);

```

## External API General Settings and Data Model

See below the general settings for External APIs that are made in the <appSetting> section of the *web.config* file:

- **Asynchronous Run** is performed by the FTOS\_IntegrationProcess\_JobServer\_RunAsyncInstances workflow.

Use the FTOS\_IntegrationProcess\_NumberOfAsyncInstancesToProcess key in the <appSetting> section of the *web.config* file to set up the number of asynchronous External API runs that can be queued.

- **Default Timeout**

Use the FTOSIntegrationProcessDefaultTimeOut key in the <appSetting> section of the *web.config* file to set up the default timeout for calling the web service.

## Data Model

The following entities are used to implement External APIs:

Entity	Description
FTOS_IntegrationProcess	External API definitions.
FTOS_IntegrationProcessDetail	
FTOS_IntegrationProcessDetailParameters	
FTOS_IntegrationProcessDetailDependency	
FTOS_IntegrationProcessInstance	Data about External API instance runs.
FTOS_IntegrationProcessInstanceDetail	Data about External API calls.
FTOS_IntegrationProcessInstanceDetailLog	Logs data about HTTP request and responses of the API calls.
FTOS_IntegrationProcess_AsyncProces	Data about asynchronous runs.
FTOS_IntegrationProcess_AsyncBatchProces	How many External API's are asynchronously processed in real time.

## Camel DSL Best Practices

Follow the below best practices when writing Camel DSL routing scripts, connectors helper classes, or unit tests.

### Build Camel DSL Routing Scripts

The Camel DSL is a language that allows you to configure the Camel Routing Engine behavior. When writing Camel DSL routing scripts, use the below best practices:

1. Group all inbound routes under the rest directive, with each route directed to a mini-pipeline to handle each route request.

```
rest("/codat")
    .consumes(MediaType.APPLICATION_JSON_VALUE)
    .post("/createclient").to
("direct:createClient")
    .get("/checkdatastatus/{clientId}").to
("direct:checkDataStatus")
    .get("/getbalancesheet/{clientId}").to
("direct:getBalanceSheet");
```

2. To obtain the properties from the JSON request, get the desired property from the `JSONPath` and move it in `exchangeProperties`.

To treat a required field as a bad request, 404, or some other custom error, the `JSONPath` can receive an extra `bool` parameter that stands for `SupressJsonParseException`. By default the `SupressException` is `false` and the parameter is optional. Without this parameter set to `true`, the parsing throws an exception and the execution of the DSL script exits with the HTTP status code 500.

```
.setProperty("personName").jsonpath("$.name", true)
    .choice()
    .when().simple("${exchangeProperty.name}
== null")
    .setBody(constant("name parameter is
missing from request body"))
    .to("direct:badRequest")
```

3. Camel DSL does not differentiate between a route parameter or a URL query parameter when a request call is made. All parameters are found inside `${headers}` and `log(${headers})` displays all headers.

#### NOTE

However, if you want to access a specific header, using `${headers.key}` does not work. To access an individual header you need to reference it by using `${header.key}`.

4. Starting with the latest Apache Camel version, the `ExchangeProperties` can be accessed via `${exchangeProperties}` as a collection. To access a specific key, the same mechanism from [step 3](#) applies, you need to reference `${exchangeProperty.key}`.

```
.toD(
"${exchangeProperty.baseUrl}/companies/${exchangeProperty.cl
ientId}/data/financials/balanceSheet${exchangeProperty.query
Params}")
```

5. To pass an inbound request body to the outbound request, use a Set Body component using the Simple language.

```
.setBody(simple("${body}"))
```

6. An error regarding bridge requests is returned if the headers in a helper class are not cleared. If the headers are not cleared, then all the inbound request headers are passed to the outbound request headers ( authorization headers, API tokens headers, and API keys headers).

### HINT

Enabling bridge requests as an alternative is not recommended.

7. To set a header, use the `.setHeader("key", "value"). Set Header` component.

```
.setHeader(Exchange.HTTP_METHOD, constant(HttpMethods.GET))
```

8. To execute methods from helper classes, use `.bean("class_name", "method_name")`. This invokes the method from the class in your Camel DSL Script.

```
        .bean("codatRequest",
"mapClientIdUrlSegment")
        .bean("codatRequest",
"mapBalanceSheetQueryParams")
        .bean("codatRequest",
"setupCodatBaseRequest")
```

## Create Connectors Helper Classes

Helper classes are Java objects used to share knowledge such as basic error handling, helper functions, and so on. Below are some key points to take into account when writing Camel DSL helper classes for connectors.

- Use `@Service` to make the class available for use in Camel DSL via the `.bean` directive.

```
@Service("codatRequest")
public class CodatRequest {
    ....
}
```

- `@Headers` and `@ExchangeProperties` are shared variables by Camel DSL. If you want access to headers and exchangeProperties you need to add them as parameters. Then, they are injected automatically, allowing you manipulate them.

```
public void setupCodatBaseRequest(@Headers Map<String,
String> headers, @ExchangeProperties Map<String, Object>
properties) throws VaultConfigurationException {
    // get codat auth Token from Vault
    String authToken = getVaultProperty("AUTH_TOKEN");

    // set default headers & authorization for codat
    request
    headers.clear();
    headers.put(HttpHeaders.ACCEPT,
MediaType.APPLICATION_JSON_VALUE);
    headers.put(HttpHeaders.CONTENT_TYPE,
MediaType.APPLICATION_JSON_VALUE);
    headers.put("Authorization", "Basic " + authToken);

    // set baseUrl for codat request
    properties.put("baseUrl", getVaultProperty("BASE_
URL"));
}
```

- To handle different exceptions according to your business, develop your own Exception Classes for special handling of each cause of error.

```
public class VaultConfigurationException extends Exception {
    public VaultConfigurationException(String errorMessage) {
        super(errorMessage);
    }
}
```

```
private String getVaultProperty(String property) throws
VaultConfigurationException {
    String propertyKey = String.join("_", VAULT_PREFIX,
property);
    var value = cachingService.getStringValue
(Constants.Cache.VAULT, propertyKey);
    if (value.isPresent())
        return value.get();
    else
        throw new VaultConfigurationException(property +
" not configured in Vault");
}
```

## Write Unit Tests

Below are some procedures of writing unit tests for a simple route. The following Camel DSL script, receives an inbound request containing a JSON body, checks if the body contains a property called name and then performs an outbound request to an external service to retrieve some data. It then returns this data to the original caller.

```
from("direct:createClient")
    .routeId("createClient")
    .log("Codat Create Client Route started")
    .marshal().json()
    .log("Request Body: ${body}")
    .setProperty("name").jsonpath("$.name", true)
    .log("Received New Company Name:
${exchangeProperty.name}")
    .choice()
        .when().simple("${exchangeProperty.name} ==
null")
            .log("name parameter is missing from body")
            .setBody(constant("name parameter is
missing from body"))
            .to("direct:badRequest")
        .otherwise()
            .bean("codatRequest", "setupCodatBaseRequest")
            .setBody(simple("${body}"))
            .toD(
"${exchangeProperty.baseUrl}/companies?throwExceptionOnFailure=false")
            .id("createClientEndpoint")
            .unmarshal().json()
            .log("Received Response: ${body}");
```

Add an `id` to the outbound request before writing unit tests for Camel connectors. For example:

```
.toD(
    "${exchangeProperty.baseUrl}/companies?throwExceptionOnFailure=false")
    .id("createClientEndpoint")
```

The `id` helps you mock the external request, so that your unit tests are contained within the solution. After the outbound request has been marked with an `id`. Then, add a `MockEndpoint` and a `ProducerTemplate` in your test class. For example:

```
@ActiveProfiles("local")
@SpringBootTest(webEnvironment =
    SpringBootTest.WebEnvironment.RANDOM_PORT)
@DirtiesContext(classMode = DirtiesContext.ClassMode.AFTER_EACH_
    TEST_METHOD)
@CamelSpringBootTest
@EnableAutoConfiguration
class CodatRoutesTests {

    @Autowired
    private CamelContext camelContext;

    @Autowired
    private CachingService cachingService;

    @EndpointInject("mock:createClientEndpoint")
    private MockEndpoint mockCreateClientEndpoint;

    @Produce("direct:createClient")
    private ProducerTemplate createClientTemplate;
```

### NOTE

The `MockEndpoint` is a substitute for the outbound call tagged with an `.id` directive. The `ProducerTemplate` `direct:createClient` simulates the inbound request, hence why it bears the same name as the minipipeline from the Camel DSL.

A unit test should cover a route from inbound to outbound. When testing a happy path, you are expecting a response message. However, when testing a fail path, you are expecting no message (because the route DSL Script should fail before reaching outbound).

## Happy Path Test Example

```
@Test
    void createClientSuccess_whenNameIsProvided() throws
    Exception {
        addVaultEntries();
        AdviceWith.adviceWith(camelContext, "createClient",
        routeBuilder -> routeBuilder.weaveById
        ("createClientEndpoint")
            .replace()
            .to("mock:createClientEndpoint")
            .setBody(constant
        ("resource:classpath:/createClientSuccessResponse.json")));

        var json = new JsonObject();
        json.put("name", "Test this");
        createClientTemplate.sendBody(json);
        mockCreateClientEndpoint.expectedMessageCount(1);
        mockCreateClientEndpoint.assertIsSatisfied();
    }
```

AdviceWith allows you to hook the test runtime on the Camel context, named `createClient` using the routeBuilder identification for `createClientEndpoint(outbound)`. This allows you to use `.replace()` on the existing logic and set it to the MockEndpoint declared. `setBody` for the mock simulates that the external service provides a response. The body of the mock response is in `createClientSuccessResponse.json`.

To send a JSON body from the test to the ProducerTemplate which calls the minipipeline, it is recommended that you assemble the JSON body using `JsonObject`. This is because the `sendBody()` method of the `ProducerTemplates` sends literals unless specified otherwise. If you send a string, it's sent as text, not a JSON string. If sent as a JSON string, it's considered JSON by your Camel DSL script.

Note that the message count value is (1) for `mockCreateClientEndpoint.expectedMessageCount(1)`. This is because the Camel DSL Script is expected to reach the mock endpoint (`createClientEndpoint`) and for it to return the mock JSON data defined (`createClientSuccessResponse.json`). The (1) message received indicates the test as a success.

## Fail Path Test Example

```
@Test
    void createClientFails_whenNameIsMissing() throws
    Exception {
        addVaultEntries();
        AdviceWith.adviceWith(camelContext, "createClient",
        routeBuilder -> routeBuilder.weaveById
        ("createClientEndpoint")
            .replace()
            .to("mock:createClientEndpoint")
            .setBody(constant
            ("resource:classpath:/createClientSuccessResponse.json")));

        createClientTemplate.sendBody(null);
        mockCreateClientEndpoint.expectedMessageCount(0);
        mockCreateClientEndpoint.assertIsSatisfied();
    }
```

### NOTE

Here the message count value is (0) for `mockCreateClientEndpoint.expectedMessageCount(0)`. This is because the test is expected to fail and not reach the outbound request to the external service.

The test sends null value as a body which causes an issue in the Camel DSL script.

```
        .setProperty("name").jsonpath("$.name",
true)
        .log("Received New Company Name:
${exchangeProperty.name}")
```

```

                .choice()
                .when().simple("${exchangeProperty.name}
== null")
                .log("name parameter is missing from
body")
                .setBody(constant("name parameter is
missing from body"))
                .to("direct:badRequest")

```

The `.setProperty("name").jsonpath("$.name", true)` result in a property “name” inside `exchangeProperties` and that “name” property is null. When checking if the property exists or not, `.when().simple("${exchangeProperty.name} == null")` yields true and the script ends with 400 bad request. The mock `createClientEndpoint` is not reached, therefore the mock never replies with any message; expected 0 is true.

## Create a Connector

Connectors are simple, data point connections to external SaaS providers, that are used to enrich internal customer data to help in onboarding and risk decision scenarios. API Connectors gather data from a collection of REST APIs with the purpose of merging them into a cloud-based data storage system. This process gives the possibility of filtering and transforming data into a proper format or structure for the purposes of querying and analysis.

Follow the below steps when building a new ecosystem connector.

## Prerequisites

In order to build a new connector, you need to the following:

- A code editor such as [Visual Studio Code](#) or [IntelliJ IDEA](#).
- Latest versions of [Java Development Kit](#), [Git](#), and [Apache Maven](#) installed.
- Submit a request to [FintechOS ITSupport](#) to obtain access to *FintechOS-ManagedServices Azure Devops Artifactory* ([service-pipes - Azure Artifacts](#)) and the GitHub template.

## Use the GitHub Template

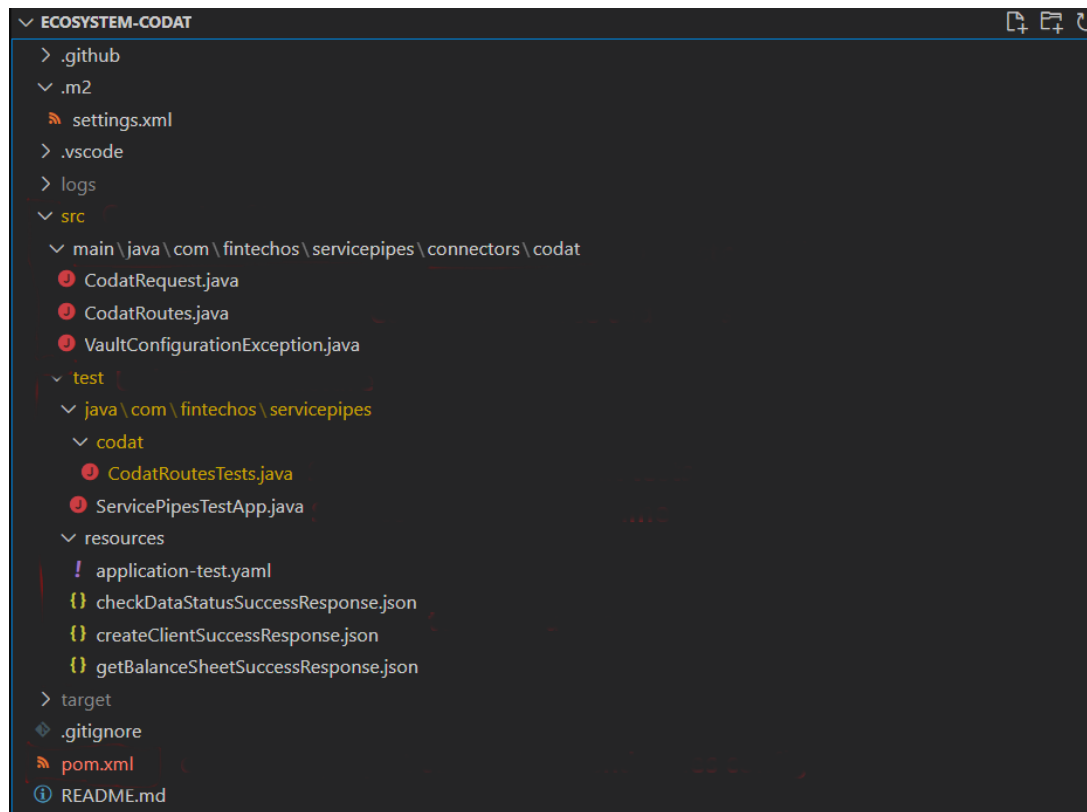
Ecosystem connectors cannot run independently and must be included in a [Service Pipes](#) project. Use the instructions from the [Service Pipes template](#) to set up and configure your Service Pipes Applications projects.

## Create the Project Structure

To define the project structure, you need to:

1. Create a repository in Git and name it **ecosystem-project**.
2. Clone the created repository locally.

3. Build the project structure as per the below image:



### HINT

As a template, you can use the [Codat Apache Camel Connector Project](#).

## Project Configurations and Settings

### Project Object Model (POM) File

The `pom.xml` file contains project and dependencies configurations and has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>com.fintechos.servicepipes</groupId>
        <artifactId>service-pipes-dependencies</artifactId>
        <version>1.0.0</version>
    </parent>
    <groupId>com.fintechos.servicepipes.connectors</groupId>
    <artifactId>service-pipes-codat</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <name>Codat connector</name>
    <properties>
        <service.pipes.version>1.1.0</service.pipes.version>
    </properties>
    <scm>
        <tag>HEAD</tag>
    </scm>
    <dependencies>
        <dependency>
            <groupId>com.fintechos.servicepipes</groupId>
            <artifactId>service-pipes-core</artifactId>
            <version>${service.pipes.version}</version>
            <scope>provided</scope>
        </dependency>
        <!--Test-->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.apache.camel</groupId>
            <artifactId>camel-test-spring-junit5</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <repositories>
        <repository>
            <id>service-pipes</id>
            <name>FintechOS-ManagedServices</name>
            <url>https://pkgs.dev.azure.com/FintechOS-
ManagedServices/_packaging/service-pipes/maven/v1</url>
            <releases>
                <enabled>true</enabled>
            </releases>
            <snapshots>

```

```

        <enabled>true</enabled>
      </snapshots>
    </repository>
  </repositories>
</project>

```

## Parent

Gets the service pipes dependencies from [Azure Artifactory](#) and the specified version on build.

```

<parent>
  <groupId>com.fintechos.servicepipes</groupId>
  <artifactId>service-pipes-dependencies</artifactId>
  <version>1.0.0</version>
</parent>

```

## Project Properties

Sets the information for the connector project that you're currently working on.

```

<groupId>com.fintechos.servicepipes.connectors</groupId>
<artifactId>service-pipes-codat</artifactId>
<version>1.0.0-SNAPSHOT</version>
<name>Codat connector</name>

```

Parameter	Description
name	Project name.
groupId	Specifies where the artifact resides in the Artifactory.
artifactId	Artifact name.
version	The version the project. You can increment it manually or via GitHub when creating a release.

## Properties

Here you can define property variables to be used inside `pom.xml`.

```
<properties>
  <service.pipes.version>1.1.0</service.pipes.version>
</properties>
```

## Dependencies

The main dependency is: `service-pipes-core`. The version is used from the `<properties>` tag.

```
<dependencies>
  <dependency>
    <groupId>com.fintechos.servicepipes</groupId>
    <artifactId>service-pipes-core</artifactId>
    <version>${service.pipes.version}</version>
    <scope>provided</scope>
  </dependency>
  <!--Test-->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-
test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.camel</groupId>
    <artifactId>camel-test-spring-
junit5</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```

## Repositories

Where the custom dependency artifacts are located. The repository id and name get tied with the information present in the [Maven settings.xml](#) file as pulling packages are needed for building the application.

```
<repositories>
  <repository>
```

```

        <id>service-pipes</id>
        <name>FintechOS-ManagedServices</name>
        <url>https://pkgs.dev.azure.com/FintechOS-
ManagedServices/_packaging/service-pipes/maven/v1</url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</repositories>

```

### Maven setting.xml File

There are two Maven settings.xml files:

- A global file that resides in the Maven installation folder that serves some configurations,
- A local file that can take or overwrite the configurations from the global file when necessary. In this case, the local file has priority.

```

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://maven.apache.org/SETTINGS/1.0.0
      https://maven.apache.org/xsd/settings-1.0.0.xsd">
  <profiles>
    <profile>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <id>service-pipes</id>
      <repositories>
        <repository>
          <id>service-pipes</id>
          <url>https://pkgs.dev.azure.com/FintechOS-
ManagedServices/_packaging/service-pipes/maven/v1</url>
          <releases>

```

```

        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </repository>
  </repositories>
</profile>
<profile>
  <id>sonar</id>
  <activation>
    <activeByDefault>true</activeByDefault>
  </activation>
  <properties>
    <sonar.host.url>
      ${env.SONARQUBE_URL}
    </sonar.host.url>
  </properties>
</profile>
</profiles>
<servers>
  <server>
    <id>service-pipes</id>
    <!--
      <username>${env.MAVEN_USERNAME}</username>
      <password>${env.MAVEN_CENTRAL_TOKEN}</password>-->
      <username>FintechOS-ManagedServices</username>
      <password>glndomqjwnjx2emjasegmfrizukiqooe2zl4jhhsbcggtt
olr2awa</password>
    </server>
    <server>
      <id>github</id>
      <username>${env.GITHUB_ACTOR}</username>
      <password>${env.GITHUB_TOKEN}</password>
    </server>
  </servers>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
</settings>

```

## Build a Connector

To build the connector, execute in cmd.exe one of the following commands:

- `mvn package / mvn clean package`: build / rebuild command.

You can also use this command with the clean parameters to also clean the project before the build.

- `mvn install / mvn clean install`: build and publish in your local repository, located in the [Maven installation folder](#).

If you have other applications that are dependent on the connector, the command uses the local version of the build package, instead of pulling it from the Azure DevOps repository.

[Unit test](#) are executed by default as they initialize the service pipes application container to start for every single test. To avoid spending too much time running unnecessary tests, execute the following command to run only certain unit tests:

- `mvn clean install -DskipTests=true`

### IMPORTANT!

The connector cannot run independently and needs to be integrated in a [Service Pipes project](#).

## Debug a Connector

### NOTE

To debug a connector, make sure you first enable debugging in the Service Pipes App.

Follow the below steps to add the debug configuration on your connector project in in [Visual Studio Code](#):

1. Navigate to **Run > Add Configuration**.
2. Replace the `launch.json` contents with the following JSON code:

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
```

```
// For more information, visit:  
https://go.microsoft.com/fwlink/?linkid=830387  
"version": "0.2.0",  
"configurations": [  
  {  
    "type": "java",  
    "name": "Debug (Attach)",  
    "projectName": "MyApplication",  
    "request": "attach",  
    "hostName": "localhost",  
    "port": 5005  
  }  
]
```

**NOTE**

Port 5005 is the same as the configuration section in the Service Pipes App `pom.xml`.

3. Run the Service Pipes App on your local machine and attach to it the Connector Project Debugger.

## Use Service Pipes Container Apps

Connectors cannot run independently unless integrated in a Service Pipes container application. This chapter shows how to create a Service Pipes Project and run your own Service Pipes container application.

### Prerequisites

In order to create a Service Pipes Project, you need to the following:

- A code editor such as [Visual Studio Code](#) or [IntelliJ IDEA](#).
- Latest versions of [Java Development Kit](#), [Git](#), and [Apache Maven](#) installed.
- Submit a request to the [FintechOS ITSupport](#) to obtain access to *FintechOS-ManagedServices Azure Devops Artifactory* ([service-pipes - Azure Artifacts](#)) and the GitHub template.

## Create the Service Pipes Project and start it locally

To define create the project, you need to:

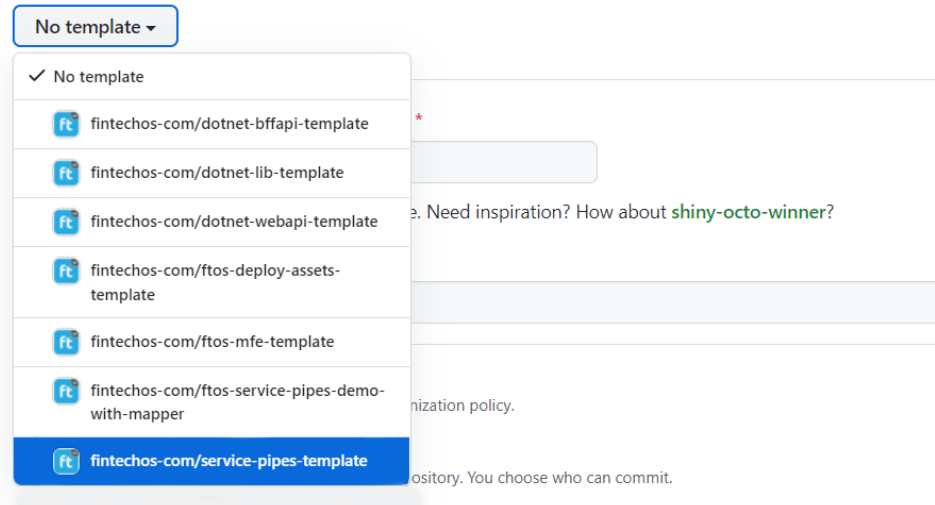
1. Create a new repository in Git and use **service-pipes-template**.
2. Use the steps in the Readme to create and configure your project.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

#### Repository template

Start your repository with a template repository's contents.



The Service Pipes Container created has a `pom.xml` file that has the following structure:

```
<parent>
```

```

        <groupId>com.fintechos.servicepipes</groupId>
        <artifactId>service-pipes-dependencies</artifactId>
        <version>1.0.2</version>
    </parent>
    <artifactId>service-pipes-app</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <name>Service Pipes Template</name>
    <scm>
        <tag>head</tag>
    </scm>
    <properties>
        <service.pipes.version>1.1.0</service.pipes.version>
        <docker-repository>acrcloudservices.azurecr.io</docker-
repository>
        <service-pipes-codat.version>1.0.0-SNAPSHOT</service-pipes-
codat.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>com.fintechos.servicepipes</groupId>
            <artifactId>service-pipes-core</artifactId>
            <version>${service.pipes.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.camel.springboot</groupId>
            <artifactId>camel-spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>com.fintechos.servicepipes.connectors</groupI
d>
            <artifactId>service-pipes-codat</artifactId>
            <version>${service-pipes-codat.version}</version>
        </dependency>
    </dependencies>

```

## Parent Dependency

A parent dependency on **service-pipes-dependencies**.

```

<parent>
    <groupId>com.fintechos.servicepipes</groupId>
    <artifactId>service-pipes-dependencies</artifactId>
    <version>1.0.2</version>
</parent>

```

## Application Name and Version

The Service Pipes application name and version information.

```
<artifactId>service-pipes-app</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <name>Service Pipes Template</name>
```

## Properties

Here, you can add the connectors versions that you are developing and put them in the Service Pipes application container.

```
<properties>
  <service.pipes.version>1.1.0</service.pipes.version>
</properties>
```

## Dependencies

Here, you can add the connector references of the connectors you want to include in this service pipes container.

### IMPORTANT!

This structure also needs to contain the service-pipes-core dependency.

```
<dependencies>
  <dependency>
    <groupId>com.fintechos.servicepipes</groupId>
    <artifactId>service-pipes-core</artifactId>
    <version>${service.pipes.version}</version>
  </dependency>
  <dependency>
    <groupId>org.apache.camel.springboot</groupId>
    <artifactId>camel-spring-boot-
starter</artifactId>
  </dependency>
  <dependency>
```

```

groupId      <groupId>com.fintechos.servicepipes.connectors</
groupId>
              <artifactId>service-pipes-codat</artifactId>
              <version>${service-pipes-
codat.version}</version>
              </dependency>
            </dependencies>

```

## Run the Service Pipes Container Application

Use the below steps when running the Service Pipes Container Application.

1. Add all your connectors as reference/ dependencies in the `pom.xml`.
2. Open the `application-local.yaml`.
3. Populate existing and add the needed keys, so that the app container can mock them and serve them to your application.

Once the connectors are referenced inside the container's `pom.xml` and the configurations are inside `application-local.yaml`, open `cmd.exe` and run the following command inside the repository root folder:

- `mvn spring-boot:run "-Dspring-boot.run.profiles=local"`

### IMPORTANT!

The connectors inside the Service Pipes container application need to be already published:

- locally (installed in your local `.m2` repository)
- available in the Azure DevOps repository

The port on which the Service Pipes container application runs is configurable in the `application.yaml` files. The default values are:

- port: **8080**
- base path: `http://localhost:8080/services/api/XXXXXXX`. where `XXXXXXX` is a placeholder for the routes defined in your integrated connectors.

## Enable Debug Mode

To enable the debug mode, you need to uncomment or add the following section in the service pipes app' pom.xml:

```
[...]
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <!-- uncomment this to enable debug -->
      <!--
      <configuration>
        <jvmArguments>-Xdebug -Xrunjdwp:transport=dt_
socket,server=y,suspend=n,address=5005</jvmArguments>
      </configuration>
      -->
    </plugin>
  </plugins>
</build>
<profiles>
  <profile>
[...]
```

The <configuration> tag with <jvmArguments> for -Xdebug enables the debug port on the java virtual machine

### NOTE

The port is specified in the section, for the above example it is **5005**.

With the section added / uncommented, [run the application](#). The debug mode is a prerequisite when [debugging connectors](#).

# Analytics

Analytics allow you to examine aggregated data from the database by extracting, filtering, and finally displaying results in a graphical or tabular form. For instance, you can use analytics to create a report that lists all delinquent debtors, along with their outstanding payments and days past due.

There are two stages in setting up your analytics: collecting your data and setting up how you wish to present it.

1. To extract the required data, you can run advanced queries based on fetch objects or database stored procedures.
2. For presentation, you have the option to display the results in the form of simple grid reports, SSRS compliant reports that meet financial regulations, interactive Power BI reports, or charts.

## Data Source

Data sources are used for generating reports.

## Reports

Use the Reports section to generate simple grid reports that offer a tabular view of the data, or custom reports that are used for statutory reporting to legal authorities.

Reports can be accessed based on previously defined security roles, or automatically generated on predefined milestones and dates. You can export the reports to multiple file formats, or deliver them to subscribers by e-mail or to a shared file. In addition, reports can be generated from the application menu.

## Advanced Analytics

Advanced analytics bring data examination using business intelligence (BI), in order to process large amounts of unstructured data, discover insights, make predictions, and generate recommendations.

Integrated into FintechOS, Microsoft Power BI helps analyze data, determine metrics, and share insights across the organization.

## Charts

Charts are visual presentations of data that help you convey information, and understand data in a visual way. Use charts when a tabular report won't adequately show relationships between data points.

This section covers the following topics:

---

## Data Sources

You can use an SQL stored procedure or fetch data as data sources for your reports.

## Add a Data Source

1. From the Main Menu, click **Analytics > Data Sources**. The **Data Sources List** page appears.
2. Click **Insert**. The **Add Data Source** page appears.
3. Enter the **Name** of the data source that is used by the system.
4. Enter the **Display Name** of the data source that is displayed in the UI.

5. If your data source is a stored SQL procedure, tick the **Use Stored Procedure** checkbox, and in the **Stored Procedure** field, enter the name of the SQL procedure following this convention: `procedure_name_as_stored_in_DB @Id @EntityName @UserId`. For more information, see ["Use Stored Procedures" below](#).
6. If your data source is fetch data, tick clear the **Use Stored Procedure** checkbox and fetch the data. For information on how to fetch data, see ["Use Fetch Data" below](#).

## Use Stored Procedures

**Prerequisite:** Create the SQL procedure that you want to use.

The following parameters of the stored SQL procedure are automatically mapped to specific values, as described in the table below:

Parameter	Value mapped to
@userId	The ID of the user that runs the <b>Report</b> linked to the <b>Report Document</b> .
@businessUnitId	The ID of the business unit that the user belongs to.

If you want to use other parameters declared within the stored SQL procedure, you have to save and reload the **Add Data Source** page, and add the parameters in the **PARAMS** section first. For information on how to add parameters, see [Add Report Parameters](#).

## Use Fetch Data

**Prerequisites:**

- Make sure there are at least two entities in the system. For information on how to add entities, see ["Create Entities" on page 60](#).
- Create relationships between the entities on which you do the fetch. For more information on relationships, see [Entity Relationships](#).

- Add custom attributes to each entity for which you do the fetch; you can use the attributes when defining the fetch. For information on how to add attributes, see [Adding Attributes](#).

To fetch the data, you can write the fetch directly into the **Fetch Object Expression** field.

You can also use the **Fetch Designer**:

1. Click **Show Fetch Designer**.
2. Choose the criteria and conditions for clustering the database.
3. Choose the data to be included in the report.

Basic fetch:

```
return {
  "entity": {
    "alias": "base",
    "name": "entity",
    "attributelist": null
  }
}
```

Change the values that you want to be dynamically replaced by the parameter values by using `getParamValue`.

Dynamically change property values with the parameter values when generating the report.

In the example above, we are going to change the value of the **name** property as follows:

```
return {
  "entity": {
    "alias": "base",
    "name": "entity",
    "attributelist": null
  },
  where: {
    type: "and",
    conditionlist: [{
      first: "base.defaultEntityStatusId",
      type: "equals",

```

```

                second:"val(<span style="background-color:
#fff0f0">getParamValue(entityStatus</span>))"
            }]
        }
    }
}

```

When generating the report, the system uses the value of the `entityStatus` parameter for the **name** property.

The platform supports multiple parameters at once, therefore you can use `getParamValue` to dynamically change properties value with the parameters value as many times as you need.

**NOTE** You have to save and reload the **Add Data Source** page and in the **PARAMS** section, add the parameters whose values replace the values of properties as defined in the fetch. For information on how to add parameters, see [Add Report Parameters](#).

## Charts

Prior to creating charts, think about the outcome that you want to achieve, or the specific data that you want to show. Make your charts simple to keep your audience focused on relevant information and to avoid confusion.

### Add a chart

1. From the main menu, click **Analytics > Charts**. The **Charts List** page appears.
2. Click **Insert**. The **Add Chart** page appears.
3. Configure the chart:

Chart Attribute	Description
Name	The chart name used by the system.

Chart Attribute	Description
Chart Title	The chart name displayed in the Digital Experience Portal.
Chart Base Type	<p>Two options are available:</p> <ul style="list-style-type: none"> <li>• <b>Standard</b> - represents a Name/Value series on the X/Y axes. For example, income= 20k, education=high school.</li> <li>• <b>Series</b> - represents more series from a given fetch. For example, client Paul Mathew: income=15k, education=undergraduate. When you have a third variable it is suitable to use series.</li> </ul>




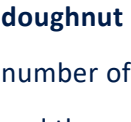
Chart Attribute	Description
Chart Type	<p>Select one of the following options from the drop-down:</p> <ul style="list-style-type: none"> <li> <b>bar</b> - displays rectangular bars. Use bar charts to present or compare data from the same category, e.g., the sales value or product volume over a period of time.            </li> <li> <b>line</b> - displays a two-dimensional scatter-plot of ordered values connected by lines following their order. Use lines charts to display the relationship between multiple sets of data over a period of time.            </li> <li> <b>Pie</b> - displays percentage values as a slice of a pie. Use pie charts to focus on the big picture, drawing attention to important information.            </li> <li> <b>doughnut</b> - similar to pie charts, this type of charts have a number of elements including the division of segments and the meaning of arc for an individual segment. It is useful to present the relationship between proportions of different data groups.            </li> </ul>


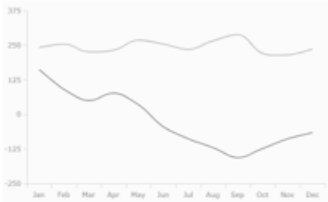


Chart Attribute	Description
	<div></div> <ul style="list-style-type: none"><li><b>spline</b> - similar to line charts, the spline chart has data points connected with smooth curves instead of straight lines. It usually displays the change of a variable over time (in Cartesian charts) or simply along the X-axis (for Scatter charts). Spline charts are the basis for spline area charts.</li></ul> <div></div> <ul style="list-style-type: none"><li><b>scatter</b> - allow analyzing how different goals have been achieved around a main topic and their different dimensions over a time period. For example, compare types of products based on budgets and selling prices.</li></ul> <div></div> <ul style="list-style-type: none"><li><b>area</b> - similar to line charts, they have solid plot lines. They're useful for displaying trends over a period for single or several categories, or the change between several data groups.</li></ul> <div></div>

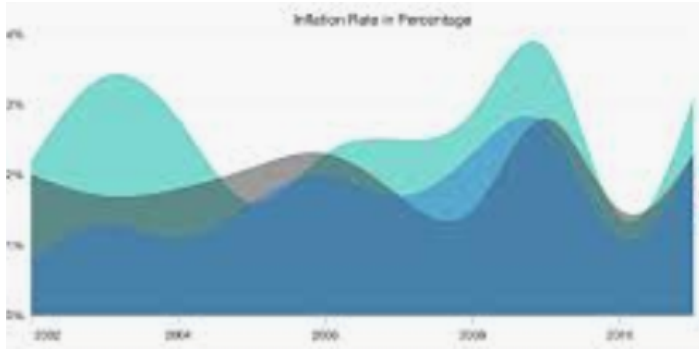
Chart Attribute	Description
	<ul style="list-style-type: none"> <li>• <b>spine area</b> - an area chart in which data points are connected by smooth curves. The area between the line segments and the X-axis is colored to emphasize the magnitude of change over time.</li> </ul>  <p><b>NOTE</b> For base charts of type <b>series</b>, the following chart types are not available: <b>pie</b> and <b>doughnut</b>.</p>
<b>Name field</b>	Alias used in the fetch which maps the name on X axis.
<b>Value field</b>	Alias used in the fetch which maps the value on the Y axis.
<b>Series argument field</b>	The pairing of an argument and its value are represented on a diagram's axes as their X and Y coordinates. This field holds the third. For example, this would be the client variable.
<b>Show legend</b>	Tick to display data about the datasets that appear on the chart. You can choose where you want the legend to be shown on the chart: <b>Legend horizontal alignment</b> (left, right or center) and <b>Legend vertical alignment</b> (top or bottom).
<b>Show labels</b>	Tick to add data labels to the data points of the chart. Labels help you identify data series in a chart.

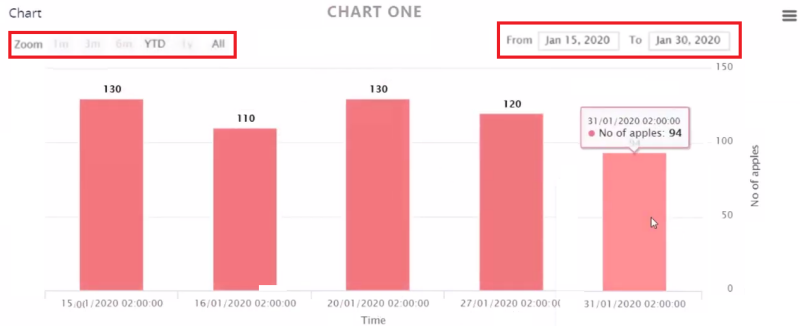
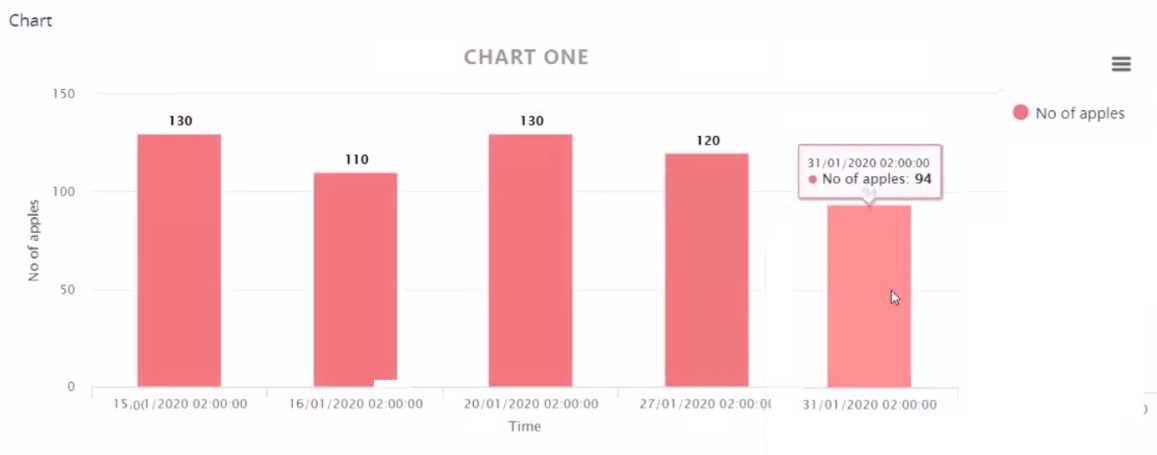
Chart Attribute	Description
<b>Override Color</b>	Change the default rendering color of the chart.
<b>Axis X Title</b>	The title to be displayed on the chart's X axis.
<b>Axis Y Title</b>	The title to be displayed on the chart's Y axis.
<b>Render Type</b>	The chart rendering type: <b>Chart</b> or <b>Finchart</b> .
<b>Render As Time Series</b>	<p>Tick to render chart values over a period of time. It enables you to zoom in/out chart data per time intervals (1 month, 3 months, 6 months, year, or all) and also filter the data within specific from - to dates.</p> <p>The figure below presents a standard bar <b>Finchart</b> rendered as time series:</p> 
<b>Container CSS Class</b>	The name of the CSS class to add to the graphic container.
<b>Container CSS Inline Style</b>	The CSS changes you want to apply to the chart.
<b>After Generate Js</b>	JavaScript code to be executed after the chart is rendered.

Chart Attribute	Description
Fetch expression return object	Returns the fetch data for the chart.

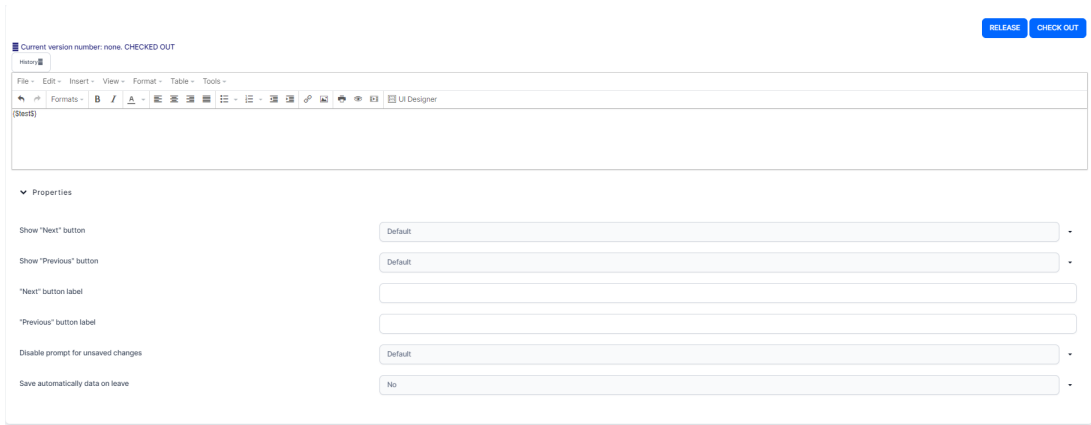
- Click **Save and close** . If you want to see how the chart looks like, click **Save and Reload** and scroll down to the Chart section which displays the chart.

The figure below shows an example of a standard bar Finchart which is not rendered as time series:

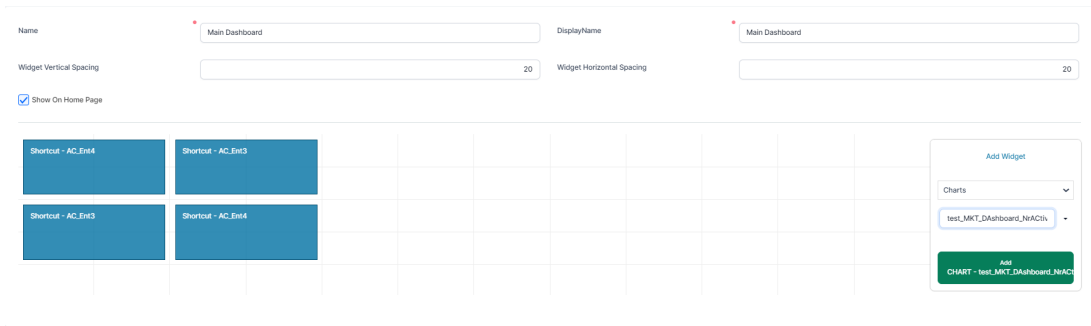


There are various ways in which you can render charts:

- On forms by using the token **{ChartName\$}**. By creating a form driven flow and in the **Steps** tab, create the step you wish to insert the chart in e.g. Step 4 and in the **UI** tab insert the token with the name of the chart created. Click **Save and close**. Display the shortcut for the Digital Journey in a dashboard and navigate to Step 4 to see the chart.



- Add charts to **Digital Experience Portal** dashboards. Go to the desired dashboard and select the **charts widget** select the name of the chart and click **Add** chart as shown below.



- Programmatic by using the **ebs.generateChart** method. For details, see [ebs.generateChart](#) in the [Client SDK user guide](#).

# Advanced Analytics

Microsoft Power BI integrated into FintechOS Studio enables you to embed interactive reports and visuals into your app. With the use of BI systems, it helps identify data patterns. A Power BI report is a multi-perspective view into a dataset with different findings and insights from that dataset.

**NOTE**

FintechOS Platform integrates with Microsoft Power BI, however FintechOS does not provide user accounts with the latter. Contact your organization for a Microsoft Power BI account.

To use a Microsoft PowerBI Report in FintechOS Studio, follow these steps:

1. [Register App for Power BI](#)
2. [Embed the PowerBI Report](#)
3. [Add the PowerBI report to the Dashboard](#)
4. [Add the PowerBI report to Digital Journeys](#)

## Register App for Power BI

### Prerequisites

- You should know the ID of your company's **Azure Active Directory** domain (also known as tenant URL).
- You should have a **Power BI** account provided by your organization.

**NOTE**

FintechOS Platform integrates with Microsoft Power BI, however FintechOS does not provide user accounts for the latter. Contact your organization for a Microsoft Power BI account.

### How to find the tenant URL

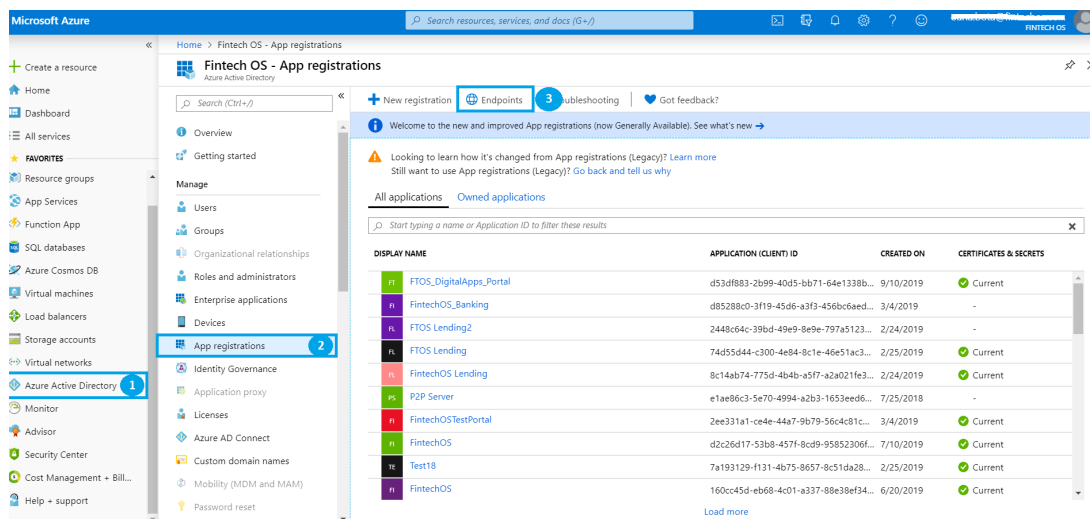
**NOTE**

To find the tenant URL, you should have a **Microsoft Azure** account provided by your

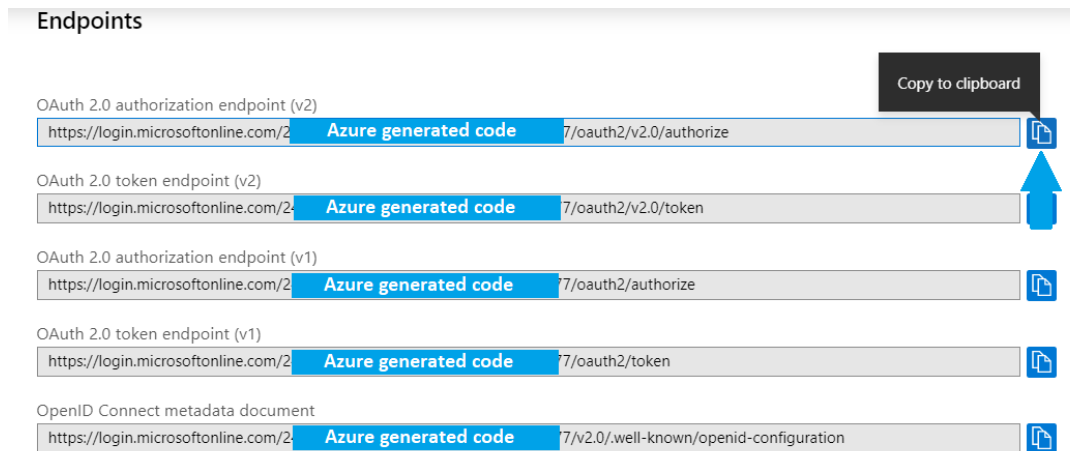
organization.

Follow these steps to find the tenant URL:

1. Go to <https://portal.azure.com/>. You are automatically logged in with your **Microsoft** account.
2. From the main menu, click **Azure Active Directory**. Your company's **Azure Active Directory** overview appears.
3. From **Manage** menu, click **App registrations**. The list of all apps registered by your company in **Microsoft Azure** appears.



4. On the toolbar, click **Endpoints** to open the list of available endpoints.



5. In the list, search for the **OAUTH 2.0 authorization endpoint** regardless the version, and click the **Copy to clipboard** icon corresponding to the endpoint.

You are going to need the tenant URL after you register the app for **Power BI** in **Azure**.

## Register app for Power BI

Once you configure the **Power BI** reports in the user interface, you need to register your application for **Power BI**:

1. Access the **Power BI** [registration link](#). Follow on-screen instructions.

### NOTE

You need to log in to your **Power BI** account to be able to register your app for **Power BI**.

2. In the **App Name** field type a descriptive name for your app as it is displayed on the login page. The **App Type** field is set by default to **Server-side Web App**.
3. In the **Redirect URL** field, type the application URL and add the suffix **Azure/Redirect.aspx**.

E.g., [https://188.210.90.229/EBSCore\\_CRM/Azure/Redirect.aspx](https://188.210.90.229/EBSCore_CRM/Azure/Redirect.aspx)

**NOTE**

Provide a secure redirect URL, otherwise the app registration fails.

4. In the **Home Page URL** field, type the application URL and add the suffix `/Main`.

E.g., [https://188.210.90.229/EBSCore\\_CRM/Main](https://188.210.90.229/EBSCore_CRM/Main)

**NOTE**

Provide a secure application URL, otherwise the app registration fails.

5. Select the **APIs** and the level of access your app needs:

Step 3 Choose APIs to access

Select the APIs and the level of access your app needs.

Dataset APIs	Report and Dashboard APIs	Other APIs
<input checked="" type="checkbox"/> Read All Datasets	<input checked="" type="checkbox"/> Read All Dashboards	<input checked="" type="checkbox"/> Read All Groups
<input checked="" type="checkbox"/> Read and Write All Datasets	<input checked="" type="checkbox"/> Read All Reports	<input checked="" type="checkbox"/> Create Content
	<input checked="" type="checkbox"/> Read and Write All Reports	

6. Click the **Register App** button. **Power BI** generates unique keys (**Client ID** and **Client Secret**) for your application.

Step 4 Register your app

Once you've set everything the way you want it, click the button below and we'll register your app. Your client ID and secret (for web apps only) will appear below. Be sure to copy the values into your app. By clicking the Register App button, you have accepted the [terms of use](#).

**Register App**

Client ID:

28cc7d53-b7e6-4665-b14f-52d07287094c

Client Secret:

drjLbC7oBeAodMmc3bukW3NmFDcd+YgmFf0m5qFG2N8=

7. Add your Azure client ID and tenant URL settings to the Configuration Manager:

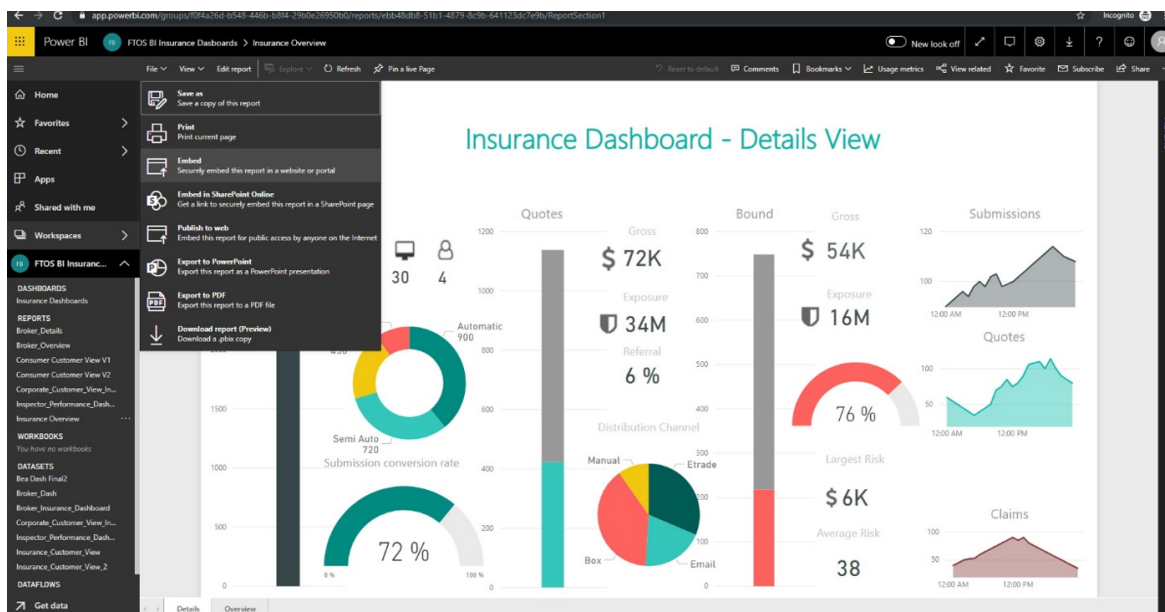
Key	Value
kv/<environment name>/<Portal name>/app-settings/Azure_ClientID	Your Azure Client ID
kv/<environment name>/<Portal name>/app-settings/Azure_TenantUrl	Your Azure Tenant URL

## Embed Power BI Report

### Get the Power BI report ID

Go to <http://app.powerbi.com> and log in using the credentials given by your organization. Go to the Power BI report and get the report ID.

This is how a power BI report might look like in the **Microsoft Power BI** app:



## Embed the Power BI report in FintechOS Studio

To embed and display a Power BI report in FintechOS Studio, follow these steps:

1. From the menu, click **Analytics > Advanced Analytics**. The **Advanced Analytics List** page is displayed.
2. Click **Insert**. The **Add Advanced Analytics** page is displayed.
3. In the Name field, type the name of the Power BI report (displayed in the user interface).
4. In the Power BI Id field, type the ID given by Power BI to your report, that is the ID displayed in the Power BI URL (not the ID of the Power BI Dashboard).
5. Select the **Authentication mode**. Two options are available:

**Requires Sign-In (legacy)** - User needs to authenticate with Azure AD in order to view the report.

#### NOTE

Supported only in HTML Widgets of type Report, this option does not support passing any parameters.

**Embedding** - An authorization token is generated by the server, and the user can view the report without authentication in **Azure AD**. This mode supports a richer API to interact with report parameters with markup components and the **Client SDK**.

6. In the **Workspace ID** field, type the unique identifier of the Power Bi workspace holding the report. This parameter is required when the authentication mode is **Embedding**.

The screenshot shows a form titled 'ADVANCED ANALYTICS' with the following fields:

- name**: A text input field containing 'myDemo'.
- Power BI Id**: A text input field containing '12aa34b5-6cde-7efg-8ghj-4i9mn12o34'.
- Authentication mode**: A dropdown menu with 'Embedding' selected.
- Workspace ID**: A text input field containing '88aa34b5-6cde-9efg-8ghj-4i9mn12o34'.

7. If you selected the authentication mode **Embedding**, click **Save and reload** to continue to add report parameters, following the steps explained in the next section. Otherwise, click **Save and close**.

## Add report parameters (Embedding authentication mode only)

Power BI report parameters define mappings to remote Azure Power BI report parameters. The table below describes the Power BI report parameters.

Parameter	Description
<b>Name</b>	The name of the parameter that will be used by the system to reference the parameter. The field is mandatory.
<b>Description</b>	The description of the parameter. Can be localized.
<b>Parameter Type</b>	<p>Mandatory field for choosing the report parameter type. Two options are available:</p> <ul style="list-style-type: none"> <li>• <b>Context binding</b> - At runtime, the parameter is bound automatically to a context property (the current entity property). If selected, in the <b>Parameter Value</b> field, provide the property name.</li> <li>• <b>Constant</b> - The parameter value attribute contains the actual value.</li> </ul>
<b>Parameter Value</b>	Available only if the <b>Context binding</b> parameter type was chosen. This field holds the property name.
<b>Operator</b>	The comparison operator used to compose the filter expression. It is comprised of the parameter, and the value.
<b>Parameter Value Data Type</b>	The data type mapping for the parameter.
<b>Power BI TableName</b>	The name of the Power BI table containing the field to be mapped.
<b>Power BI FieldName</b>	The name of the Power BI field that will be mapped to the specified parameter.

Add PowerBI parameter

PowerBI parameter

Name

pStartDate

Description

Campaign start date

Parameter Type

Context binding

- /

Parameter Value

InvStartDate

Operator

GreaterThanOrEqual

- /

Parameter Value Data Type

Invariant Date

- /

PowerBI TableName

ebs\_FTOS\_Mkt\_Campaign

PowerBI FieldName

After you finish providing the report parameter details, click **Save and Close** to save the parameter. After you finish mapping to remote Azure Power BI report parameters (adding all needed report parameters), at the top-right corner of the **Edit Advanced Analytics** page, click **Save and Close** to save the Power BI report settings.

Now you can add and use the Power BI report in your digital journeys. For more information, see ["Add Power BI Reports to Digital Journeys" on the next page](#).

## Add Power BI Reports to Dashboards

### Prerequisites

- You should have created a dashboard.
- You should have registered **Power BI** for apps.
- You should have embedded the Power BI report into FintechOS Studio

### Add a Power BI report to a dashboard

1. From the **Main Menu**, go to **Digital Experience > Digital Frontends > Dashboards**. The **Dashboards List** page is displayed.
2. Double-click on the desired dashboard. The dashboard configuration page is displayed.
3. In the **Add Widget** area, from the first drop-down, select **Power BI Report**.
4. From the second drop-down, select the Power BI report you want to add to the current dashboard. The name of the Power BI report appears in a rectangle below the toolbar.
5. Click **Add**. The report widget is added to the dashboard, and the **Add Widget** area is replaced by **Edit Widget**.
6. Optionally, you can resize the widget by placing the cursor on the bottom-right corner of the report rectangle. A resize icon is displayed. Click to drag and drop, and resize as preferred.
7. Click **Save and close**. The Power BI report is added to the dashboard.

## Add Power BI Reports to Digital Journeys

You can use the [powerbi-client](#) library to embed Power BI reports in your forms. E.g.:

Use the **ebs.msal.renderMsalAuthButton** and **ebs.msal.getPowerBiToken** functions to render a Power BI authentication button and retrieve the Power BI authentication token respectively.

```
ebs.importClientScript("PowerBiScript");
// This a a client script library that stores the
https://github.com/microsoft/PowerBI-JavaScript/tree/master/dist
Library

/* Render the Power BI authentication button */
const msalEl = document.querySelectorAll('.msal-dialog-btn__app')[0];
console.log(msalEl);
ebs.msal.renderMsalAuthButton(msalEl);

/* Click event for the refresh-msal-token button */
$('#refresh-msal-token').on('click', async function (event) {
  console.log("refresh-msal-token");
  const ttok = await ebs.msal.getPowerBiToken();
  const msalElTok = document.querySelectorAll('.msal-token')[0];
  msalElTok.textContent = ttok;
});

/* Click event for the gen-report button */
$('#gen-report').on('click', function (event) {
  console.log("The button gen-report was clicked");
  embedReport()
});

/**      */
async function embedReport() {
  const reportId = "xxxxxxxx-yyyy-zzzz-www-xxxxxxxxxxxx";
  const embedUrl = "https://app.powerbi.com/reportEmbed?reportId="
+ reportId;
  const accessToken = await ebs.msal.getPowerBiToken();
  const msalElTok = document.querySelectorAll('.msal-token')[0];
  msalElTok.textContent = accessToken;
}
```

```

    embed(reportId, embedUrl, accessToken);
  }
  function embed(reportId, embedUrl, token) {
    // 1 - Get DOM object for div that is report container
    let reportContainer = document.querySelectorAll
    ('.reportContainer')[0];
    // 3 - Embed report using the Power BI JavaScript API.
    /*    let models = window['powerbi-client '].models;*/
    let config = {
      type: "report",
      id: reportId,
      embedUrl: embedUrl,
      accessToken: token,
      //permissions: models.Permissions.All,
      //tokenType: models.TokenType.Aad,
      /*viewMode: models.ViewMode.View,*/
      settings: {
        panes: {
          filters: { expanded: false, visible: true },
          pageNavigation: { visible: false },
        },
      },
    };
    let report = powerbi.embed(reportContainer, config);
  }

```

## Simple Grid Reports

Simple grid reports offer a tabular view of data; each column representing a field and each row representing a record.

**Prerequisite:** Create a [data source](#) for your report.

### 1 Add Report Parameters

Report parameters are used for inputting data into document reports, used to filter the data when generating the report.

To add a report parameter, follow these steps:

1. In the **Edit Data Source** page, scroll down to the **PARAMS** section and click **Insert** . The **Add Data Source Param** page appears.
2. Provide the **Name** of the parameter matching the name that used in the custom fetch, or in the stored procedure.  
  
When generating the report, the value of the "Name" property is replaced with the value of the "entityName" parameter provided in the fetch, we will add the "entityName" parameter.
3. In the **Display Name** field, enter the name of the parameter as it is displayed in the UI.
4. If your license includes the parallel datastore option, you can select the **Use parallel datastore** checkmark to store the report in the dedicated database.
5. Select the **Attribute** type. For more information on the types of attributes available in the platform, see [Types of Attributes](#).
6. If you want to add multiple parameters, click **Save and reload** and add the parameters. Otherwise, click **Save and close**.

## 2 Add a Simple Grid Report

1. From the Main Menu, click **Analytics > Reports**. The **Reports List** page appears.
2. Click **Insert**. The **Add Report** page appears.
3. Fill in the fields, as follows:
  - **Name** - Enter the report name that is used by the system.
  - **Display Name** - The name of the report which is displayed in the Portal. This field is mandatory.
  - **Scope** - Select General.
  - **Type** - Select Simple Grid Report.

- **Report Render Format** - Select the format type from the drop-down: XLSX, CSV or OnScreen.

4. Click **Save and reload** . The **Edit Report** page appears.

### IMPORTANT!

You have to add a report item, otherwise the report cannot be generated.

## 3 Add Report Items

A report item represents a configuration for the report that is added to the report, and that gathers data within the specified dates. You can have many report items, but only the one set as default is used upon the report generation.

To add items to a simple grid report, follow these steps:

1. In the **Edit Report** page, scroll down to the **Report Items** section and on top of the section, click **Insert**. The **Add Report Item** page appears.
2. Select the **Start Date** and **End Date**. These represent the validity period of the report item.
3. Select the **Data Source** for the report.
4. Select the **Is Default** checkbox if this is the item that you want to be used when generating the report. Only one report item can be set as default.
5. Click **Save and reload** if you want to add another report item, otherwise, click **Save and close** .

If needed, you can restrict users' access to the report by adding security roles to the report. For more information on security roles, see the [Define Report Access Privileges](#) section.

## 4 Define Report Access Privileges

If your business case requires that the simple grid report is available to designated roles within your organization, in the **Edit Report** page, scroll down to the **Report Security Roles** section, click **Insertexisting**, and select the security roles that should have access to report. If no security roles are added here, all users are able to view the report.

Once you finish adding "Security Roles" on page 1245, click **Save and close**.

## Custom Reports

For custom reports, FintechOS integrates with SQL Server Reporting Services (SSRS) in order to generate custom reports that comply with financial regulations. Custom reports can be generated on demand based on the selected time frame, and the inclusion/exclusion criteria.

### Prerequisites:

- A SSRS user account with credentials at hand (username and password).
- A report in SSRS.

## 1 Add a report

1. From the main menu, click **Analytics > Reports**. The **Reports List** page appears.
2. Click **Insert**. The **Add Report** page appears.

## 3. Fill in the fields:

- **Name** - Enter the report name that will be used by the system.
- **Display Name** - The name of the report that will be displayed in the Portal. This field is mandatory.
- **Entity Menu Section** - Select the entity menu section where users will be able to generate the report.
- **Show In Menu** - Select the check box only if you selected the entity menu section where users will be able to generate the report.
- **Scope** - Select General or Entity.
- **Type** - Select Document, Custom Report or Simple Grid Report. Document is available only for the scope Entity. Simple Grid Report is available only for scope General.

For scope General or Entity, and type Custom Report:

- **Server Url** - Enter the URL of the SQL Server Reporting Services (SSRS).
- **Domain** - Enter the domain name where you host the SSRS.
- **Username** - Enter the username associated to your SSRS user account.
- **Password** - Enter the password of your SSRS user account.
- **Always Return File** - Tick the checkbox to return the file. Only available for scope Entity.

For scope Entity and type Document or Custom Report

- **Output Method** - Select the output method: Attach to entity or Download file.
- **Destination Field** - Set the destination field for your report. Only available for Attach to entity output method.
- **Destination File Name** - Add the destination file name.
- **Report Render Format** - Only available for type Document. Select a document type from the drop-down: DOCX, PDF, XLSX.

4. Click **Save and reload** . The **Edit Report** page appears.

**IMPORTANT!** You have to add a report item; otherwise the report cannot be generated.

## 2 Add report items

This section is used for setting the time interval from which the report will select the data. Each insertion of a report item is sort of a template for that report. Hence, it is possible to add more templates the same report with different dates.

The screenshot shows the 'Edit Report Item' form with the following fields and values:

Field	Value	Icon
Name	FirstItem	
StartDate	01/01/2017	Calendar
EndDate	14/02/2022	Calendar
Data Source	TestReportDataSource	Dropdown
IsDefault	<input checked="" type="checkbox"/>	
Report	TestReport	Dropdown

1. In the **Edit Report** page, scroll-down to the **Report Items** section. At the top of the section, click **Insert**. The **Add Report Item** page will be displayed.
2. The **Name** field is automatically filled with the name of the report inserted in Step 1.3.
3. Select **Start Date** and **End Date**. Upon the report generation, it will gather data within the specified time interval (between the start date and the end date).
4. In the **Report Path** set the path for where this report will be stored.
5. For the bool **IsDefault**, tick if you wish this template to be the default for the report. If not, leave empty.
6. The **Report** field is automatically filled in with the name of the report.

7. Click **Save and reload** if you want to add another report item, otherwise, click **Save and close**.

**NOTE**

For the Document report type, the Add a Report Item page also displays the Report Document field.

Here you can use a Digital Document to have XLSX and DOCX reports generated based on a template.

For more details on how to create your template and how to add a digital document, check the [Digital Documents Processor User Guide](#).

### 3 Insert Report Parameters

This section allows setting one or more attributes for the report. This is available only if the scope of the report was set to **Entity**. Fill in the following fields:

Field	Data type	Description
<b>Name</b>	Text	Insert a name for the parameter.
<b>Attribute</b>	Option set	Select an attribute for the list.
<b>Report</b>	Text	It is automatically filled in with the name of the report inserted at Step 1.3.

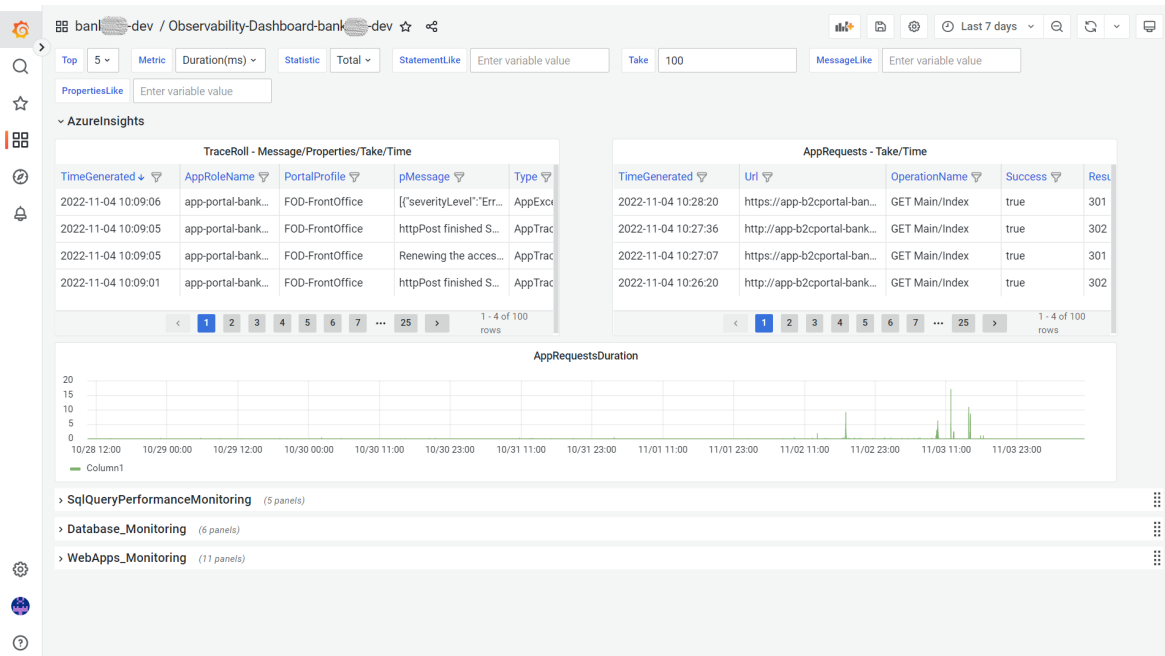
Click **Save and close** . Repeat as many times as needed.

### 4 Define who has access to the custom report

If your business case requires that the custom report is available to designated roles within your organization, in the **Edit Report** page, scroll-down to the **Report Security Roles** section. Click **Insert existing** , and select the security roles that should have access to report. If no security roles are added here, all users will be able to view the report. For details, see "[Security Roles](#)" on page 1245.

# Observability & Telemetry

FintechOS incorporates Grafana into the observability solution. Grafana is an open-source analytics & monitoring platform, which allows users to see their data in charts and graphs that are aggregated into one (or multiple) dashboard(s) for easy interpretation and understanding. Grafana includes built-in support for Azure Monitor data source, useful when monitoring the FintechOS - Azure infrastructure services. Data is gathered from all Azure resources and FintechOS applications, and the entire set of Azure Monitor and Grafana features enable you to monitor application performance.



## 1 Authentication & Roles

Grafana inherits the FintechOS Identity Provider for authentication and is by default configured to reside in a subdomain of your FintechOS environment, so once you are

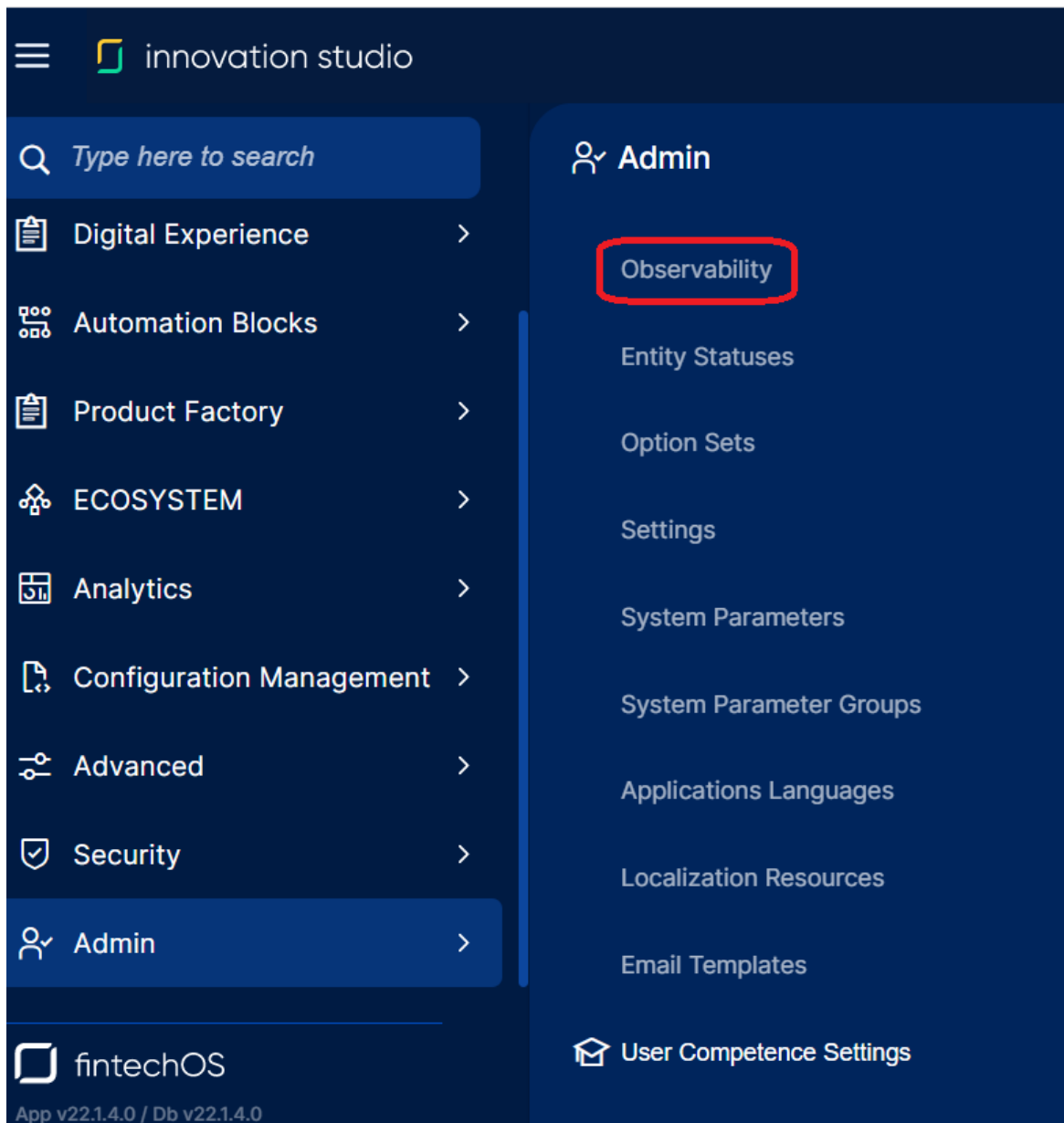
logged in FintechOS Studio or Portal, login is done automatically. Grafana roles need to be created in FintechOS Studio and are then included through scripting in the IDP token at deployment.

There are three types of permissions associated with Grafana roles:

- **Viewer** - allows the user to authenticate in Grafana, but they cannot see any dashboards. To grant read-only access, the role viewer must be allocated per dashboard in Grafana.
- **Editor** - provides read-write access to the dashboards in the instance.
- **Admin** - provides full control of the instance including viewing, editing, and configuring data sources.

Also, in FintechOS Studio, you need to use the "Observability" security role to map a connection between your established roles and the Observability menu item displayed in the UI. For more details on security roles, see the [Manage Security Roles page](#).

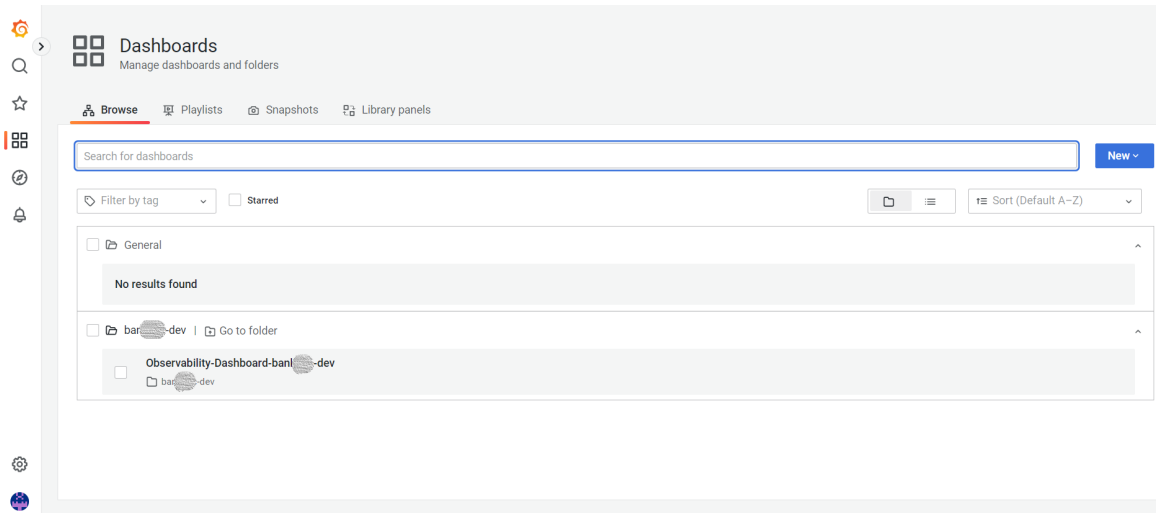
To open Grafana, from the FintechOS Studio main menu, navigate to **Admin** and select **Observability**. This opens the Grafana web application in a new tab.



**NOTE** You need to assign the 'Observability' security role to users, for the 'Observability' menu item to be visible to them in the menu UI.

## 2 Dashboards

A dashboard is a set of one or more panels organized and arranged into one or more rows. Each panel provides data from a specific FintechOS data source. Dashboards can be organized and grouped into folders. You reach the dashboard folder page by hovering over a result in the dashboard search, or on the Manage dashboards page.



During environment deployment, two dashboards are created and configured automatically: a default dashboard and a business dashboard.

### Default dashboard

This dashboard contains the default setup of metrics for monitoring FintechOS - Azure services and application performance, and is structured on the following panels:

#### Monitoring Metrics

#### WebApps (for FintechOS components and Business Components)

- CPU\_Percentage\_Portal
- CPU\_Percentage\_MFE\_CAMEL\_OpenAPI
- Requests\_Portal\_Studio\_B2CPortal
- CPU\_Percentage\_B2CPortal

- CPU\_Percentage\_Envoy
- Requests\_JobServer
- CPU\_Percentage\_Studio\_JobServer
- HTTP\_5xx\_Portal\_Studio\_B2CPortal
- ResponseTime\_Portal\_Studio\_B2CPortal
- ResponseTime\_JobServer
- ResponseTime\_Camel\_Envoy

### AzureInsights

- TraceRoll (for FintechOS components)
- AppRequests
- AppRequestsDuration

These metrics are described here: [Monitor apps - Azure App Service](#).

### Database

- Database\_DTU
- Database\_CPU
- SqlDB\_Deadlocks
- SqlDB\_Data\_IO
- SqlDB\_Log\_IO
- Database\_Failed\_Connections

### SqlQueryPerformance

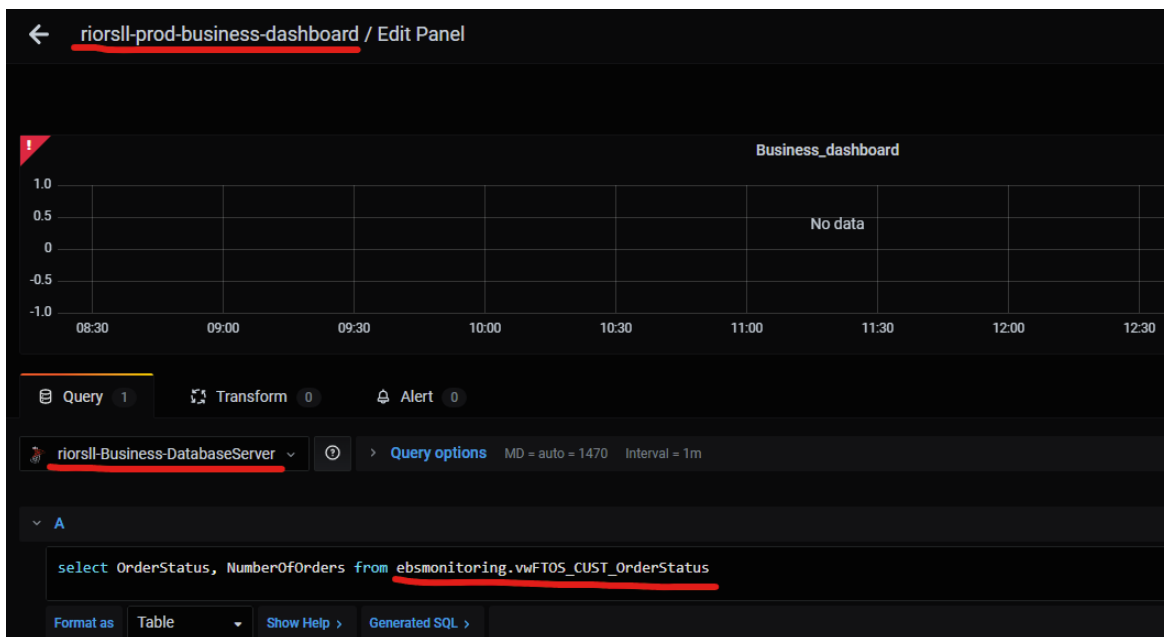
- TopResourcesConsumer
- MissingIndex

- Big Tables
- Deadlocks

## Business dashboard

The business dashboard works like a frame you can use during the project implementation phase, where you can create different monitoring panels based on your specific business needs. For this, a special data source was created with the following naming: ***production\_client\_name-Business-DatabaseServer***. The only permission of the database user that this data source uses is the 'Read' permission on the **ebsmonitoring** schema.

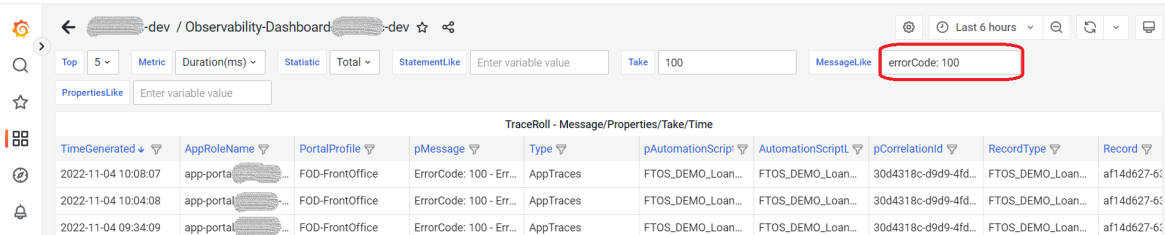
**NOTE** You need to create SQL views that have the *ebsmonitoring* schema - this should be used in the queries that get the data in the business dashboards.



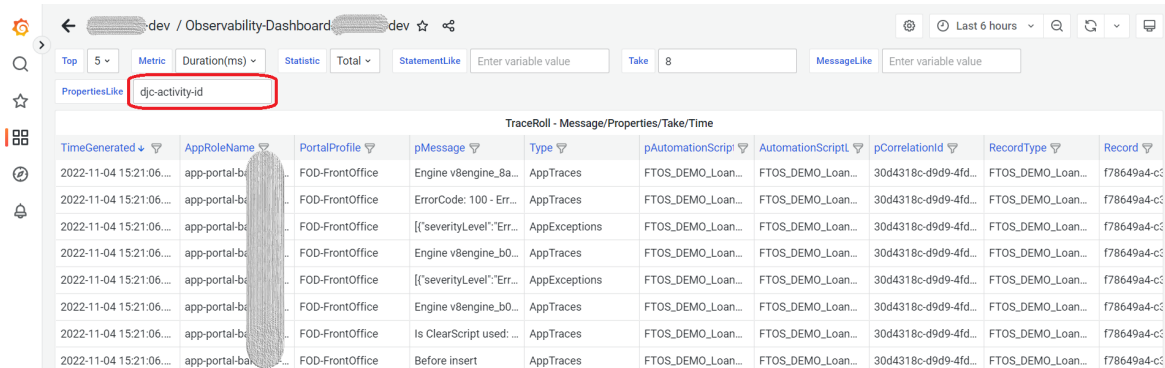
## 3 Search & Filtering

Grafana allows for extensive search and filtering options so you can customize your queries and get specific data. For example, you can type a variable value in the **MessageLike** field, and you will get only the TraceRoll logs that contain that specific

value:



You can also limit the number of rows displayed in the results with the **Take** field, filter columns by value, or refine your query with the **PropertiesLike** field:



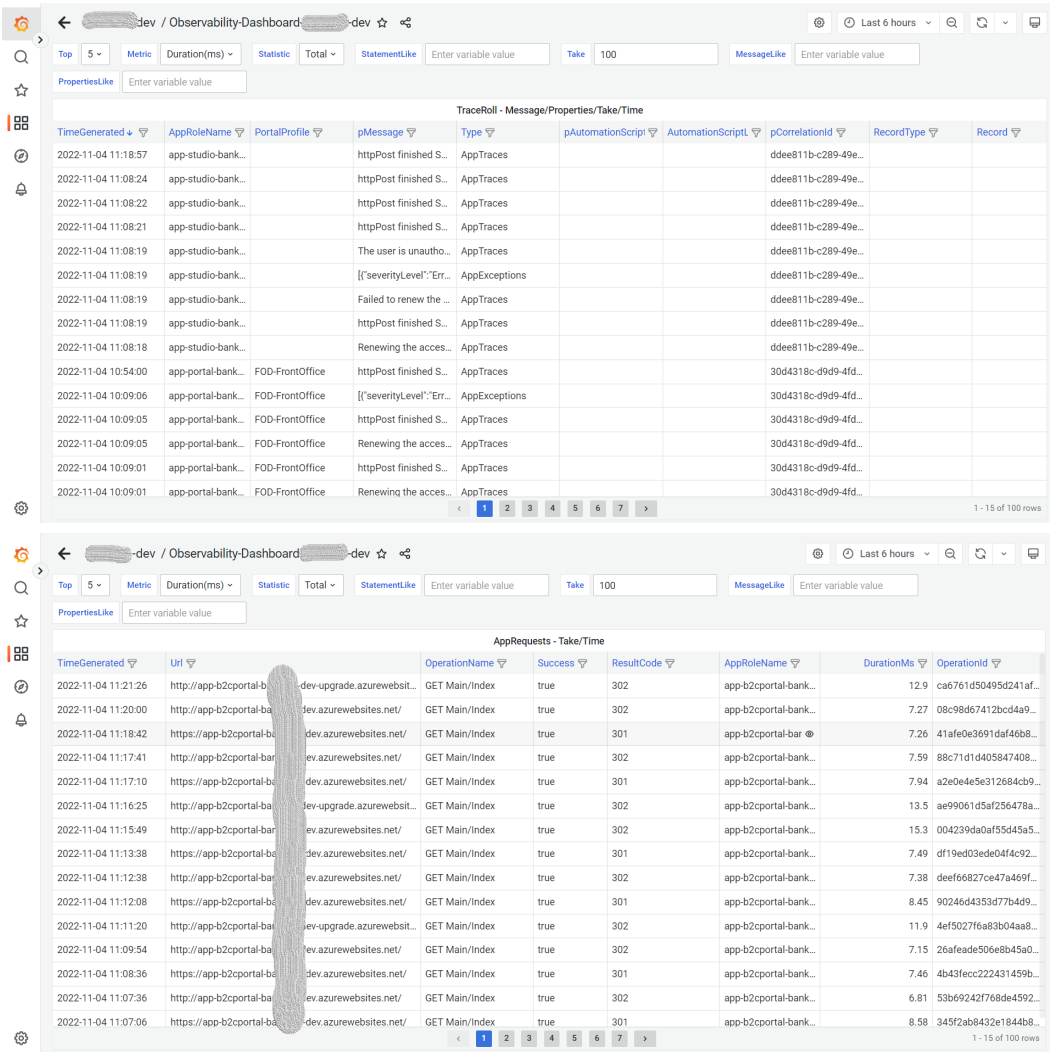
## 4 Examples

### Panels

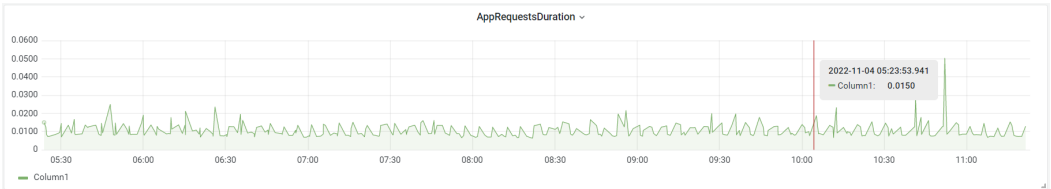
Below are the various panels displayed in Grafana for a production environment, with time intervals ranging from hours to the last three months.

#### AzureInsights

**TraceRoll - Message/Properties/Take/Time , AppRequests - Take/Time**



AppRequestsDuration



SqlQueryPerformanceMonitoring

TopResourcesConsumer - Metric/Statistic/Top/Time

← dev / Observability-Dashboard dev ↻

🔍

Top ▾ Metric ▾ Duration(ms) ▾ Statistic ▾ Total ▾ StatementLike  Take 100 MessageLike

PropertiesLike

Big Tables - Top

SchemaTableName	Table_Name ▾	rows ▾	reserved_MB ▾	reserved_GB ▾	data_MB ▾	data_GB ▾	index_size_MB ▾	index_size_GB ▾	unused_KB ▾
EbsMetadata	ProjectDeployment...	954	490	0.478	489	0.477	0.150	0.000100	56
EbsMetadata	CustomizationSetL...	401991	432	0.422	392	0.383	37.7	0.0368	229
EbsMetadata	VersionData	30401	354	0.346	343	0.335	8.46	0.00830	268
EbsMetadata	CustomizationSetL...	40455	130	0.127	39.6	0.0387	72.3	0.0706	1834
ebs	FTOS (Prici...	7116	123	0.120	105	0.103	13.4	0.0131	436

## Big Tables - Top

	query_id	est_logical_reads	sum_executions	avg_avg_logical_io_reads	query_text	query_plan
	11497	7025045	12225	575	((messageStatusInP uniqueIdentifier,@utc...	<ShowPlanXML xmlns='http://schemas.mic...
	10139	380874	4883	78	(@cultureName nvarchar(4000))SELECT * F...	<ShowPlanXML xmlns='http://schemas.mic...
	1113619	12314	1	12314	(@p1 nvarchar(50))SELECT * FROM ( SELE...	<ShowPlanXML xmlns='http://schemas.mic...
	1125086	6672	8	834	(@p1 nvarchar(5))SELECT * FROM ( SELEC...	<ShowPlanXML xmlns='http://schemas.mic...
	40	2505	1	2505	select * + S.name + "" + O.name "" as Tab...	<ShowPlanXML xmlns='http://schemas.mic...

## Big Tables - Top

← dev / Observability-Dashboard dev ↻

🔍

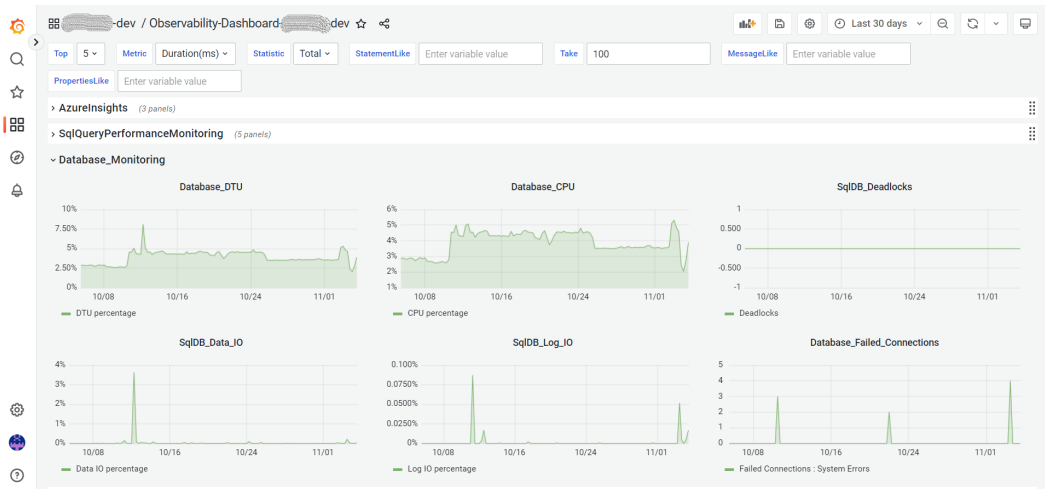
Top ▾ Metric ▾ Duration(ms) ▾ Statistic ▾ Total ▾ StatementLike  Take 100 MessageLike

PropertiesLike

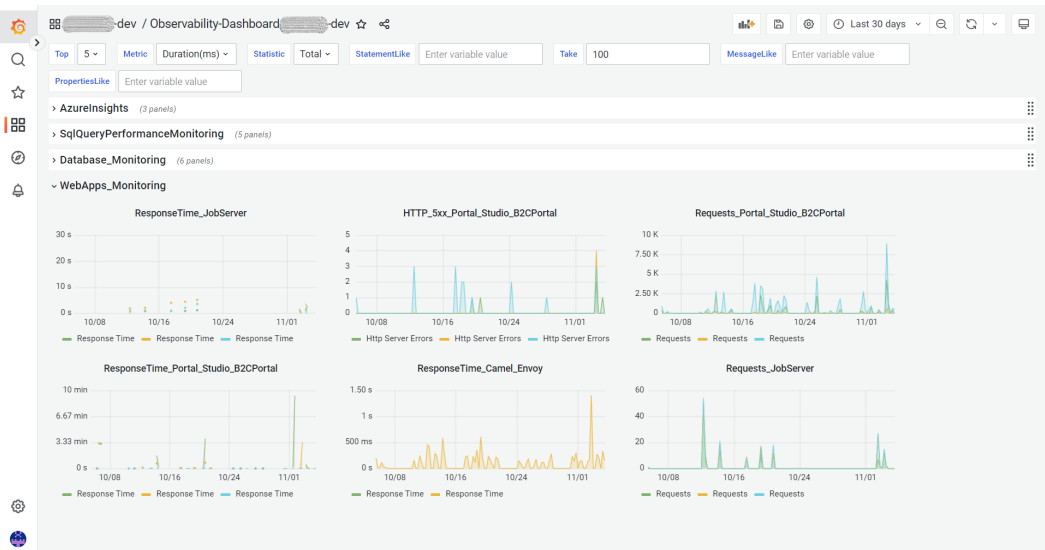
Big Tables - Top

SchemaTableName	Table_Name ▾	rows ▾	reserved_MB ▾	reserved_GB ▾	data_MB ▾	data_GB ▾	index_size_MB ▾	index_size_GB ▾	unused_KB ▾
EbsMetadata	ProjectDeployment...	954	490	0.478	489	0.477	0.150	0.000100	56
EbsMetadata	CustomizationSetL...	401991	432	0.422	392	0.383	37.7	0.0368	229
EbsMetadata	VersionData	30401	354	0.346	343	0.335	8.46	0.00830	268
EbsMetadata	CustomizationSetL...	40455	130	0.127	39.6	0.0387	72.3	0.0706	1834
ebs	FTOS (Prici...	7116	123	0.120	105	0.103	13.4	0.0131	436

Database\_Monitoring



WebApps\_Monitoring



# Configuration Management

Configuration management allows you to control the ownership, versioning, deployment, and import/export of all components that fulfill a specific business need (such as a digital journey) by organizing them into digital assets. Digital assets group together configuration items (entities, attributes, libraries, digital journeys, etc.) that belong to a common context. This allows you to:

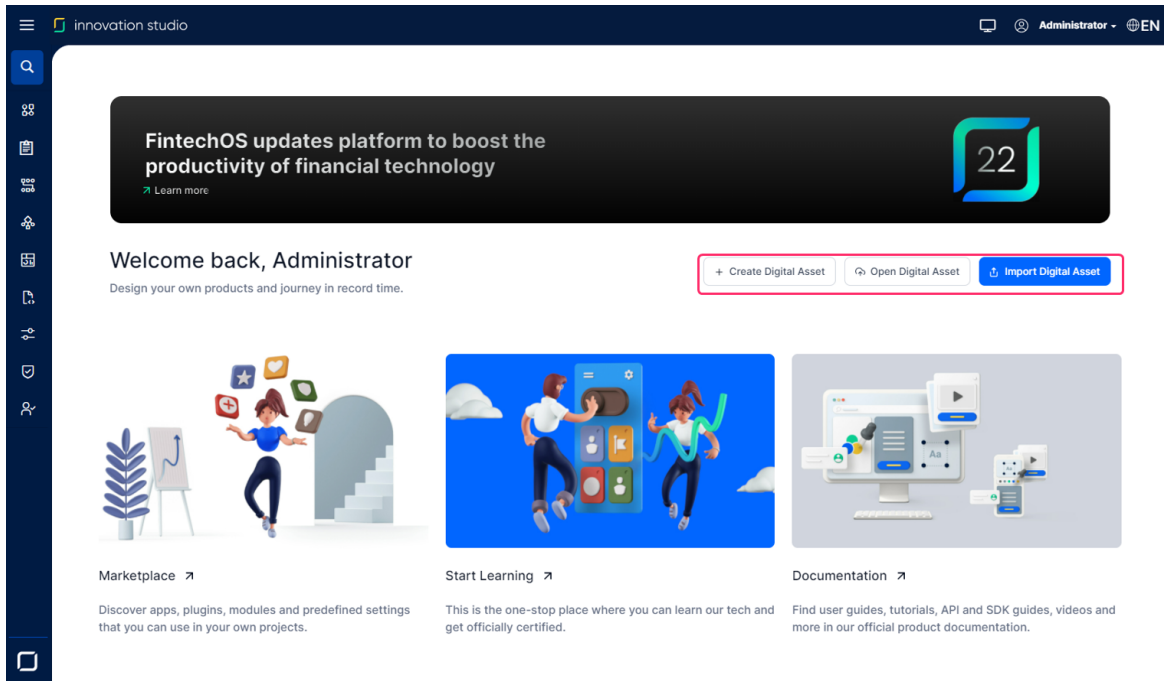
- Prevent situations where different digital solutions update the same configuration item, and corrupt each other's data.
- Eliminate the manual tracking of all the components required to run a specific functionality when you need to export it and deploy it on another environment.
- Lock an entire application in a specific state, for instance a stable version used for distribution, and prevent further modifications of its configuration items.

## The Context Digital Asset

### **IMPORTANT!**

In order to edit digital assets, your user account must have the Developer security role assigned. Also, some operations, such as locking or unlocking digital assets, are only available to users with the Release Manager [security role](#).

When you open FintechOS Studio, the first thing you should do is to set the context digital asset (the digital asset that is currently active). You can do so by either opening, importing, or creating a new digital asset.



Within the context of a digital asset, all configuration items you create are automatically assigned to that digital asset. Configuration items that belong to other digital assets are available only in read-only mode. For more information about managing digital assets, see ["Digital Assets" on the next page](#).

### NOTE

After setting the context of a digital asset, the FintechOS Studio menu sections are filtered depending on the digital asset type.

Once you select your context digital asset, a persistent header is displayed at the top of the screen. This header contains information about the context digital asset's name, version number, status (active/obsolete), state (locked/unlocked), as well as options to navigate the asset's configuration items or to close the digital asset and return to the FintechOS Studio's home page.



### NOTE

Although it is possible to work with unallocated configuration items (entities, attributes, libraries, digital journeys, etc.), it is recommended to always use digital

assets in your workflow to enforce rigorous ownership rules over each of your configuration items.

For more information about configuration management, follow the links below:

---

## Digital Assets

Digital assets group together configuration items (entities, attributes, libraries, digital journeys, etc.) that belong to a common context, such as a data model, a digital journey, or an app data form.

### IMPORTANT!

You need to have the **Developer** security role to set a digital asset as context. Without this role, you can create a digital asset, but cannot use a digital asset as context. For details, see ["The Context Digital Asset" on page 1053](#).

## Create Digital Assets

1. To start creating a digital asset, you can either:
  - Click **Create new Digital Asset** in the FintechOS Studio home page.
  - Go to **Main Menu > Configuration Management > Digital Assets** click **Insert**.
2. In the **General** tab, fill in the following fields:

Field	Description
Type	<ul style="list-style-type: none"> <li>• Data Configuration - Used for defining metadata relationships and packaging business records which comply with a broad range of use cases, such as promoting banking or insurance products from one environment to another.</li> <li>• Data Model - Used for creating and configuring generic data models that can be used by one or more Digital Assets.</li> <li>• Digital App (App Data Form) - Used for building solutions published in employee portals for mid- and back-office applications.</li> <li>• Digital Journey - Used for building customer facing journeys for a broad range of scenarios.</li> <li>• Product - For distributing products and offers across platform instances.</li> <li>• Resources - Used for bundling generic resources such as endpoints, libraries, or style sheets. These can be referenced in more than one Digital Solution by one or more digital assets.</li> </ul>
Code	Identification code for the digital asset.
Name	Insert a suggestive name.
Description	Insert a text that describes the digital asset.

Field	Description
Version	<p>Tracks the digital asset versions based on when you <a href="#">"Lock and Unlock Digital Assets"</a> on page 1074.</p> <p>To enable/disable automatic versioning, in the <a href="#">Configuration Manager</a>, at the <code>kv/&lt;environment name&gt;/&lt;Studio instance&gt;/app-settings</code> path, set the <b>feature-da-version-type</b> key to:</p> <ul style="list-style-type: none"> <li>• <i>none</i> - The version number can be edited manually. You can enter integer values or numeric values in the x.y.z format.</li> <li>• <i>autoincrement</i> - The field is readonly and incremented automatically on asset unlock.</li> </ul> <p>When switching the key from <i>none</i> to <i>autoincrement</i>, the asset versions will be incremented automatically from the previous manually set values. On versions using the x.y.z format, the patch number (z) will be incremented.</p> <p>If the <b>feature-da-version-type</b> key doesn't exist, the setting defaults to autoincrement.</p>
Status	This field is read-only. The status is automatically Unlocked.

**NOTE**

For each digital asset type, check the configuration item types compatibility in the Configuration items section.

3. Click the **Dependencies** tab. From the drop-down list, select the **Minimum Platform Version** with which the digital asset is compatible with. Once selected, the digital asset cannot be imported in a prior version of the FintechOS platform.

#### 4. Click **Save and reload**.

The digital asset is created. The second dependency available could be a dependency on another digital asset. For details, see ["Edit Digital Assets" below](#).

## Edit Digital Assets

When you open a digital asset, several configuration sections are available

### General

When editing a digital asset, the fields in the General tab that you used to create the asset (see ["Create Digital Assets" on page 1055](#)) are visible in read-only mode, with the exception of the Description field which you can still edit.

### Set the Context Digital Asset

If you click **Set as context digital asset**, the name of the digital asset will be displayed next to the FintechOS Studio logo.

The screenshot displays the 'General' tab of the FintechOS Studio interface. At the top, there is a navigation bar with eight tabs: 1. General, 2. Configuration Items, 3. Related Digital Solutions, 4. Data Import Files, 5. Custom Files, 6. Dependencies, 7. Configuration Items Migration, and 8. Advanced. Below the navigation bar, there is a 'Lock Digital Asset' button. The main form contains several fields: 'Type' (Digital Journey), 'Code' (000), 'Name' (CoreInsurance1), 'Description' (CoreInsurance1), 'Version' (1), 'Status' (Unlocked), 'Obsolete' (checkbox), and 'Creation Mode' (Primary). At the bottom left, there is a 'Set as context digital asset' button, and at the bottom right, there is a 'Set Owner Secret Key' button.

All the configuration items created while the context is active are assigned automatically to the digital asset.

To close the context digital asset, click **Close Digital Asset**.

Unlock Sealed Digital Assets

Some official FintechOS digital assets have controlled distribution and are locked by FintechOS to ensure their integrity. In this case, a **Set Customization Key** button is displayed in the interface. This button allows you to insert an unlock code provided by FintechOS to unlock the asset if you need to edit it.

Type

Digital App

Code

1

Name

1

Description

--

Version

1

Status

Unlocked

Obsolete

☐

Creation Mode

Primary

\*Sealed by Owner (Non-Customizable)

Set Customization Key

**NOTE**  
Sealed digital assets cannot be set as context digital assets and do not support the regular lock/unlock mechanism (see ["Set the Context Digital Asset"](#) on the previous page and ["Lock and Unlock Digital Assets"](#) on page 1074).

Configuration Items

This tab typically includes all the components required for the digital asset to work, such as entities, form driven flows, entity views, scripts, configurations for the automation processors, and any other configurations made while the digital asset was set as context.

You can also use configuration items to distribute ownership over different areas of a business context. For instance, you can manage an entity form in a digital asset that is different from the one that owns the entity, or you can manage a form step in a digital asset that is different from the one that owns the entity form. This is useful when delivering modular business solutions for back office scenarios.

The Configuration Items tab also shows when an item was linked to the digital asset, and by whom. This is useful for keeping track of the modifications brought to the configuration items. They are shown in the last two columns of the Configuration Items grid: **Modified On** and **User Name**.

1 General2 Configuration Items3 Related Digital Solutions4 Data Import Files5 Custom Files6 Dependencies7 Configuration Items Migration8 Advanced

+ AddBranch Digital AssetRemove

Entity Name	Record Name	Parent Name	Modified On	User Name
workflow	BD_getBusinessStatus_011		04/11/2022 17:20	Administrator
action	BD_getBusinessStatus_011		04/11/2022 17:20	Administrator
entity	BD_getBusinessStatus_01		04/11/2022 15:35	Administrator
entityform	default	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	BD_getBusinessStatus_01id	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	businessUnitId	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	createdByUserId	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	createdOn	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	entityStatusId	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator
attribute	modifiedByUserId	BD_getBusinessStatus_01	04/11/2022 15:35	Administrator

51020Page 1 of 2 (18 items)12

The section tab has three buttons above the grid. The **Branch Digital Asset** button merges two digital assets by transferring all the configuration items from one asset to the other. By clicking the button from the current asset, the configuration items will be moved to another asset and deleted from the currently displayed asset. For details, see ["Branch Digital Assets" on page 1075](#).

Manually Add Configuration Items to the Digital Asset

To add configuration items to the digital asset, click **Add**.

Add Digital Asset Configuration Item

Type

Select...

Action GroupAdvanced AnalyticsAttributeBusiness Data ModelBusiness EntityBusiness Entity ExtensionBusiness WorkflowBusiness Workflow ConfigurationChartClient Script LibraryCode BlockConfiguration Data DefinitionCustom Form

The Add Digital Asset Configuration Item page is displayed. The **Type** field allows you to select the type of configuration item.

The configuration item types compatibility depends on the Digital Asset type:

Configuration item types	Digital asset types					
	Data Model	Digital Journey	Digital App	Data Configuration	Products	Resources
Action Group		✓	✓			
Authorization Operation Type		✓	✓			✓
Advanced Analytics		✓	✓			✓
Anonymous Frontends		✓	✓			✓
Attribute	✓	✓	✓			
Back-Office Flow			✓			
Business Data Model	✓	✓	✓			
Business Entity	✓	✓	✓			
Business Entity extension	✓	✓	✓			
Business Service Component						✓
Business Workflow	✓	✓	✓			
Business Workflow Configuration	✓	✓	✓			
Chart		✓	✓			
Client Script Library		✓	✓			✓
Code Block		✓	✓			✓

Configuration item types	Digital asset types					
	Data Model	Digital Journey	Digital App	Data Configuration	Products	Resources
Configuration Data Definition	✓	✓	✓	✓	✓	
Custom Form		✓	✓			✓
Dashboard		✓	✓			
Data Import Template	✓	✓	✓	✓	✓	
Data Set		✓	✓		✓	✓
DB Task		✓	✓			✓
Digital Document		✓	✓		✓	✓
Digital Journey		✓	✓			
Email Template	✓	✓	✓	✓		✓
Endpoint		✓	✓			✓
Entity Form Step	✓		✓			
Entity Form	✓ (default form only)	✓ (if wizard mode)	✓			
Entity Unique Constraint	✓	✓	✓			
EntityView	✓ (default view only)	✓	✓			✓
Form UI Customization		✓	✓			
Formula		✓	✓		✓	✓
Formula parameter mapping		✓	✓		✓	✓

Configuration item types	Digital asset types					
	Data Model	Digital Journey	Digital App	Data Configuration	Products	Resources
Menu Item		✓	✓			
Option Set	✓	✓	✓	✓		✓
Portal Profile		✓	✓			✓
Report		✓	✓		✓	✓
Schedule Job		✓	✓			✓
Schedule Service		✓	✓			✓
Sequencer		✓	✓			✓
Server Automation Script		✓	✓			✓
Server Automation Script Library		✓	✓			✓
Style Sheet		✓	✓			✓
System Parameter	✓	✓	✓	✓		✓
System User			✓			✓
User Competence	✓	✓	✓	✓		
Web Api Client Library		✓	✓			✓
Widget		✓	✓			

where ✓ = compatible

Once the configuration item type is selected, the system will automatically display the grid with the available configuration items.

### Manually Remove Configuration Items from the Digital Asset

To remove configuration items from the digital asset, click **Remove** and confirm. This action also removes child dependencies, such as when removing an entity, it removes the entity default view, attributes, default form, or relations. Once removed, the configuration item becomes unallocated and can be used by other digital assets.

## Related Digital Solutions

This section tab displays the "Digital Solutions" on page 1077 where the digital asset is included.

The screenshot shows the 'Related Digital Solutions' tab selected in the top navigation bar. The main content area is titled 'Digital Solution - Digital Assets'. It features a search bar with a magnifying glass icon and a 'Code' label. Below the search bar, there are two input fields: 'AP\_DSCode' and 'AP\_DSName'. To the right of the search bar, there are two buttons: 'Export' and 'Refresh'.

## Data Import Files

This section tab displays the "Data Import Templates" on page 165 and the "Configuration Data Definitions" on page 1079 which allow you to populate the digital asset's metadata and data. From here, you can set the configuration items' templates and the records' data models that you will use for your imports, then you can perform the actual imports.

The screenshot shows the 'Data Import Files' tab selected in the top navigation bar. The main content area is titled 'Digital Asset Data Import Files'. It features a search bar with a magnifying glass icon and a 'Data Import Template' label. Below the search bar, there are two input fields: 'Data Config Definition' and 'Type'. To the right of the search bar, there are four buttons: '+ Insert', 'X Delete', 'Export', and 'Refresh'. Below the input fields, there are two more input fields: 'Order Index' and 'File'. The text 'No data' is displayed below the input fields.

1. To add a new package, click **Insert**. Select the **Type**:
  - Based on Data Import Templates
  - Based on Data Config Definitions
2. Depending on what was selected for the type, two new fields are displayed, Data Import Templates/Data Config Definitions. Select the name of one of the templates/definitions.

3. Click **Select file**. The local File Explorer opens. Select the desired file and add it to the template/definition.
4. Click **Save and reload**.

From this section tab, you can download the templates which were previously displayed:

1. Navigate to the Application Data Import Files grid.
2. Select from the grid the template you wish to consult.
3. Click the desired record in the grid. The Edit Application Data Import File page opens.
4. To download a file already imported into the digital asset, click the file and it will download on the local drive.

## DB objects

Use the database objects tab to easily add SQL objects such as views, stored procedures, or functions to your digital asset. Database objects added to a digital asset are executed before import when importing the digital solution package to a new environment.

Views, user defined data types and table types can be added as database objects starting with v24.1.1. However, user defined data types and table types objects can only be imported once on another environment, at creation. Packages with updates to user defined data types and table types are not supported and might cause missing reference errors. This is not the case for views, which support updates.

### **IMPORTANT!**

Digital Assets containing SQL objects can only be imported using the FtosSysPackDeployer. For more details about importing digital solution packages, check [Import and Export Packages](#).

From this section you can attach SQL objects to a digital asset:

1. Use the available search bar to look for a specific SQL object.
2. Check the box for the function, procedure, or view that needs to be attached. Keep in mind that only **stored procedures** with the name starting with "ebs." are displayed in this list. Objects that start with "FTOS\_DEV", "vwResources" , "aspnet\_" will not appear in this view.

1 General2 Configuration Items3 Related Digital Solutions4 Data Import Files5 DB Objects6 Custom Files7 Dependencies8 Advanced

Available SQL Objects

Search

Function

☐ fnFTOS\_DCO\_GetDayOfWeekOccurrenceInMonth

Procedure

☒ FTOS\_ACC\_CB\_UpdateMemberAddGroup

☒ FTOS\_ACC\_EmployeeUpdateMonthsAtJob

☐ FTOS\_CMB\_ActivateProductCategory

☐ FTOS\_CMB\_AddProductBusinessType

☐ FTOS\_CMB\_AddProductCategory

☐ FTOS\_CMB\_AddProductClass

☐ FTOS\_CMB\_AddProductStatus

☒ FTOS\_CMB\_AddProductSubClass

Selected SQL Objects

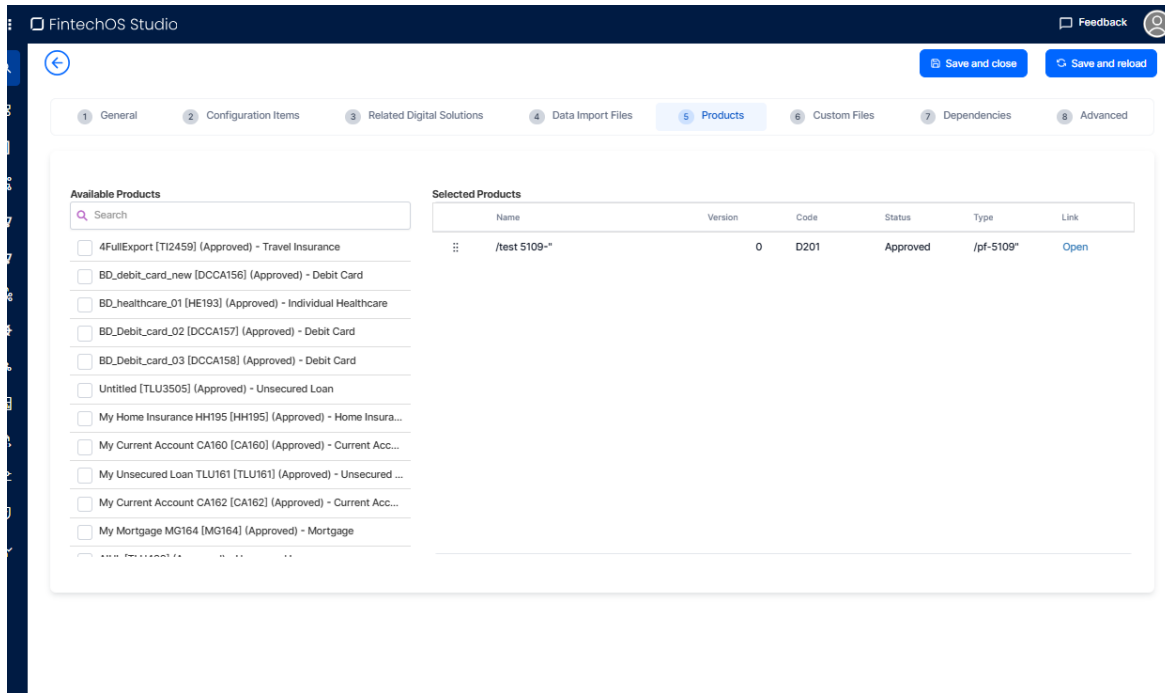
	Name	Type	Description	Save to proceed
<div></div>	FTOS_CMB_AddProductSubClass	Procedure		
<div></div>	FTOS_ACC_EmployeeUpdateMont...	Procedure		
<div></div>	FTOS_ACC_CB_UpdateMemberAdd...	Procedure		

HINT

Save the objects configuration to successfully add it to the digital asset.

Products

Use the **Available Products** search tab to pull up previously configured products that you can include in the digital asset. Click one product in the list to display its full name, version, code, status, type, and link.



## Custom Files

Custom files are used to customize [portal profiles](#), allowing you to change the appearance of the FintechOS Portal instances. They contain resources such as stylesheets, JavaScript files, or images that determine the look-and-feel of the FintechOS Portal. These resources are grouped into two main folders: custom and custom-on-demand. They are hosted using Azure Blob storage containers.

### NOTE

Before adding custom files into a digital asset, you must configure the storage location in Azure as detailed in the [Storage for Custom Files Folders](#) page.

Depending on your preferences, custom files can contain:

- style sheets (CSS) to style your application layout (change fonts, colors, margins, etc.)
- different types of images that you want displayed on your application,
- PDF files (Terms and Conditions, Loan Agreement, GDPR),
- Java script files,

- FintechOS Client SDKs;
- Font files.

### Custom and Custom-on-demand Folders Structure

- A **custom folder** contains the video custom components (css, images, and javaScripts: `dcs-sdk.js` and `onfido.min.js`). The items in this folder are loaded by default for every journey or form-driven form. To optimize performance, keep the folder size as small as possible. Use images and custom CSS here only for universal elements, such as the bank theme and other shared styles. This folder has the below structure:

```
├── {Digital Asset Name}
│   ├── js (deprecated)
│   ├── img
│   └── css
```

- A **custom-on-demand folder** contains the liveness component resources and the items it contains must be loaded manually (on demand). For example, in the case of scripts, you need to use `InjectScriptAsync` to load a library from custom-on-demand. This folder has the below structure:

```
├── {Digital Asset Name}
│   ├── common
│   ├── liveness
│   ├── usermedia
│   ├── video
│   └── zoom
```

To add custom or custom-on-demand files click the **Add file** button to upload the **Content For Custom Folder** or the **Content For Custom On Demand Folder**.

Custom Files

---

This section is used to upload .zip archives of the Custom and CustomOnDemand files needed by your solution. It is advisable you archive the files in a folder with the same name as your Digital Asset using the following structures:

Example of custom folder structure

```

(Digital Asset Name)
├── css
├── img
└── js

```

Example of custom-on-demand folder structure

```

(Digital Asset Name)
├── common
├── liveness
├── usermedia
├── video
└── zoom

```

Content For Custom Folder

Add file or Drop file here

Content For Custom On Demand Folder

Add file or Drop file here

Report Fonts Folder

Add file or Drop file here

**\*Make sure that files with the same name are not used on different Portals, as only the most recently copied file will be kept.**

**\*Preview for Custom/Custom-on-Demand is available only for development environments.**

## Custom Fonts for Document Templates

Use the **Report Fonts Folder** control to add or remove font files for your [document templates](#) to the digital asset.

## Recommended Workflow

To find problems early, test your changes separately, and move them through each stage without trouble, follow this approach:

- **Develop and attach:** create custom files and link them to the digital asset in your Dev environment;
- **Test locally:** export and re-import the package to validate functionality within Dev;
- **Promote to QA:** move the tested package to QA for formal testing;
- **Validate in UAT:** import to UAT for business approval before deploying to production.

Afterward, you can add custom files to a digital asset, export the package, and import to another environment. When you import a digital solution package, custom files are refreshed and published on portals.

## Packaging and Import Guidelines

It is advisable to pack your custom files in a digital asset and use the "[Import Digital Solutions Packages](#)" on [page 1090](#) wizard to publish the custom files on the application server. Through this wizard, you can specify which portal/application you want to deploy the custom files on.

A handy trick for troubleshooting allows you can open the files directly in your browser by navigating to a URL like this: `https://your-environment/portal/platform/custom/your_file`. Keep in mind, the exact URL may vary depending on how your custom folders are named, for example:

- `https://[your-environment]/portal/platform/custom/img.jpg`
- `https://[your-environment]/portal/platform/custom-on-demand/abs.css`

Here both `img.jpg` and `abs.css` are the names of custom files uploaded in Azure for that instance of Studio.

The import is influenced by the value set for [feature-development-mode](#) in [App Settings](#):

- if the `feature-development-mode` is set to `true (1)` then the import populates selected portals immediately;
- if the `feature-development-mode` is set to `false (0)` then you need to restart the affected FintechOS Portal instance(s) after the import, so that custom files and folders are mapped correctly.

## File Format and Structure

The files should be in .zip archive format and it is advisable to archive them in a folder with the same name as your Digital Asset. If your custom files contain a resource with the same name as an existing one, the existing resource file will be overwritten.

If you wish to include the digital solution package inside a Syspack type format (to be imported via import package pipelines), then an additional

`<digital solution package name>.config` file must be added inside the "01 DeploymentPackages" folder at the same level with the `<digital solution package name>.zip`.

The .config file should contain information similar to DeploymentSettings / ProjectSummary.xml. Read more about this on the [Importing and Exporting Package](#) page.

## Error Handling

The "File not found" exception is triggered when orphaned file references exist within the database, entries pointing to non-existent or unlinked files. Upon encountering such discrepancies, the import process stops execution and surfaces the error.

The workaround is to manually query the database for the invalid file references flagged in the error payload. These entries must be purged to restore import continuity. Note that multiple iterations may be required, as not all orphaned references are surfaced in a single pass.

## Dependencies

Dependencies allow you to share resources between digital assets by providing the current digital asset with read-only access to configuration items that belong other digital assets.

1 General 2 Configuration Items 3 Related Digital Solutions 4 Data Import Files 5 Custom Files 6 Dependencies 7 Configuration Items Migration 8 Advanced

Minimum Platform Version v20.2.0.0

Digital Asset Dependencies

+ Insert X Delete Export Refresh

Digital Asset	Code	Type	Referenced Application Version
AC_DA3	AC_DA3	Digital App	1

To add more dependencies for the digital asset:

1. In the grid labeled Digital Asset Dependencies, click **Insert**.
2. The Add Digital Asset Dependency opens with the field labeled **Referenced Digital Asset**. Click the drop-down arrow and select from the list the digital assets.
3. Click **Save and close**.

Repeat as many times as needed.

To delete a dependency, in the grid, select the digital asset by clicking once, then click **Delete**.

## Configuration Items Migration

Migration is the process of importing a deployment package (including deployment packages from prior releases that don't support configuration management) into an existing digital asset. This allows you to easily upgrade your legacy applications to take advantage of the configuration management capabilities available starting with release 21.2 (digital assets, digital solutions, digital solutions packages, etc.).

To migrate a deployment package, follow the steps:

1. Expand the **Choose deployment package for migration** field and select the deployment package to migrate to.

### IMPORTANT!

You cannot migrate an obsolete digital asset. If you attempt to migrate an obsolete digital asset, a toast message is displayed to inform the user.

2. Click **Migrate**. If the migration is successful, a grid will be displayed below the deployment package. From the grid, use the **Export** button to export records and see all deployment packages that were migrated.

## Run server scripts in the Advanced tab

In order to re-initialize sequencers, troubleshoot data entry errors, or turn formulas from draft to actives, create a digital asset that contains automation scripts that can either call an endpoint, database tasks, or server side methods.

In the **Advanced** tab, add the automation script:

- In the **Before Import** tab if the script should be executed before the import of the digital solution package on the destination environment.
- In the **After Import** tab if the script should be executed after the import of the digital solution package on the destination environment.

Click **Open in Editor** to edit your scripts in the **Code Editor**. The `beforeImport` and `afterImport` javascript files are available in the `_importOrchestration` subfolder.

#### NOTE

For the scripts to import and run successfully, the resources used by it must be available on the destination environment.

## Delete Digital Assets

To delete a digital asset, you must first unallocate all its configuration items:

1. Go to **Main Menu > Configuration Management > Digital Assets**.
2. In the digital assets list, double click the digital asset you wish to delete.
3. Open the Configuration Items tab, select all the configuration items and click **Remove**.
4. Once all the configuration items have been removed, click **Save and close** to return to the Digital Assets List.
5. From the list, select the digital asset and click **Delete**.
6. Confirm the deletion by clicking the **Yes** button when the pop-up question appears with the question *Delete one record?*

## Import Digital Assets

Sometimes, you may need to import a digital asset on your environment. This is often the case when distributing applications that are based on a digital asset.

**IMPORTANT!**

Imports based on deployment packages are deprecated and provided only for backward compatibility. The recommended method for importing digital assets is to use ["Digital Solutions Packages" on page 1086](#).

## Prerequisites

- Check the minimum required version of the package and the version of the FintechOS Studio.
- Check the dependencies of other digital assets.
- Make sure the digital asset status is not obsolete.

## Lock and Unlock Digital Assets

**IMPORTANT!**

To lock or unlock digital assets, you need to have the Release Manager security role assigned to your user account. For more information, see ["Security Roles" on page 1245](#).

Locking a digital asset prevents further modifications to its configuration items. The configuration items are still functional, i.e. you can still use the asset's digital journeys to create new records for instance, but you can no longer edit them in FintechOS Studio. This allows you to ensure compliance with your organization's application implementation workflow, for example when you wish to mark a stable version to be used for release.

When you lock a digital asset, its locked state is indicated in red in the FintechOS Studio header (when the digital asset is currently set as the context digital asset):



If you wish to resume development on a locked digital asset (for instance to start working on the next release of your application), you have to unlock it. This reopens

the asset's configuration items for editing and increments the digital asset's version number. The changes in the asset's state and version number are indicated in the FintechOS Studio header when the asset is currently set as the context digital asset:



## How to Lock a Digital Asset

1. Log in to FintechOS Studio using a user account with the Release Manager security role.
2. Go to **Main Menu > Configuration Management > Digital Assets**.
3. From the Digital Assets List, open the digital asset you wish to lock.
4. In the General tab, click **Lock Digital Asset**.

## How to Unlock a Digital Asset

1. Log in to FintechOS Studio using a user account with the Release Manager security role.
2. Go to **Main Menu > Configuration Management > Digital Assets**.
3. From the Digital Assets List, open the digital asset you wish to unlock.
4. In the General tab, click **Unlock Digital Asset**.

## Branch Digital Assets

Branching allows you to merge applications by transferring all the configuration items from one digital asset to another. This empties the source digital asset and tags it as Obsolete. You cannot reactivate a digital asset or use it in any other way once it is obsolete.

To branch a digital asset:

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Assets**.
2. From the Digital Assets List, open the source digital asset (the digital asset whose configuration items you wish to transfer).

3. Open the **Configuration Items** tab, and click **Branch Digital Asset**.

The screenshot shows the 'Configuration Items' tab selected in a multi-tabbed interface. The tabs are: 1 General, 2 Configuration Items (active), 3 Related Digital Solutions, 4 Data Import Files, 5 Custom Files, 6 Dependencies, 7 Configuration Items Migration, and 8 Advanced. Below the tabs, there are three buttons: '+ Add' (blue), 'Branch Digital Asset' (blue with a dropdown arrow), and 'X Remove' (red). Below these buttons is a table with five columns: Entity Name, Record Name, Parent Name, Modified On, and User Name. Each column has a search input field with a magnifying glass icon. The table contains one row of data: attribute, code, styleSheet, 07/11/2022 16:49, and Administrator.

4. In the Branch the current Digital Asset window, select the destination digital asset from the dropdown box. Make sure that the target digital asset is not locked or obsolete.

The screenshot shows the 'Branch Digital Asset' window. At the top right is a close button with an 'X' icon. Below it is a section titled 'Digital assets'. Under this title is a dropdown menu showing 'BD\_Digital\_Asset'. At the bottom of the window are two buttons: 'Save' (blue) and 'Cancel' (white with a grey border).

5. Click **Save**.

6. The configuration items are transferred to the destination digital asset and the source asset is emptied and marked as obsolete.

The screenshot shows the 'Lock Digital Asset' window. At the top left is a button labeled 'Lock Digital Asset'. Below it are several form fields: 'Type' (Digital Journey), 'Code' (BD\_custom\_code), 'Name' (BD\_custom\_name), 'Description' (des), 'Version' (1), 'Status' (Unlocked), 'Obsolete' (checked), and 'Creation Mode' (Primary). There is a red asterisk next to the 'Version' field. At the bottom right is a button labeled 'Set Owner Secret Key'.

# Digital Solutions

Digital solutions bundle your digital assets for easy distribution and deployment. You can use digital solutions to create complex solutions that contain multiple assets, each purposed for a specific use case. For example, if you are building a solution that contains two flows, one for the customer and another for the internal operator, then you can bundle the corresponding digital journey asset and app data form asset, as well as their shared Data Model asset under the same Digital Solution .

By bundling digital assets into digital solutions, you can leverage the "[Digital Solutions Packages](#)" on [page 1086](#) capabilities to easily export and deploy complex deliverables.

**IMPORTANT!**

Digital solutions containing "[Digital Assets](#)" on [page 1055](#) which have SQL objects attached can only be imported using the FtosSysPackage Deployer. For more details on how to import packages using the FtosSysPackage deployer, check the Import and Export Packages page.

## Create Digital Solutions

- 1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions** to display the Digital Solutions list.

Digital Solutions list		
<input type="checkbox"/>	Code	Name
Description		
<input type="text"/>	<input type="text"/>	<input type="text"/>
	01 FTOS DFP Common	01 FTOS DFP Common
	02 FTOS Content Templates	02 FTOS Content Templates
	02 FTOS Foundation	02 FTOS Foundation
	02 FTOS Project HyperPersonalization	02 FTOS Project HyperPersonalization
	02 FTOS Versioning	02 FTOS Versioning
	03 FTOS Project Campaign	03 FTOS Project Campaign
	API Demo	API Demo
	BD_Data_Model_Order_Index_001	BD_Data_Model_Order_Index_001
	BD_Data_proj_solution_02	BD_Data_proj_solution_02
	BD_Export_Solution_01	BD_Export_Solution_01
		dec
5	10	20

- 2. Click **Insert**.

3. Fill in the **Code**, **Name**, **Description** and **Version** fields accordingly.
4. Click **Save and reload**. The digital solution is now created, and the Edit Digital Solution page is displayed.

## Edit Digital Solutions

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions**.
2. In the Digital Solutions page, locate the desired digital solution and double click it.
3. In the Edit Digital Solution page, you can:
  - Edit the digital solution description.
  - Insert, delete, export digital assets that are part of the digital solution.
  - Refresh the list of digital assets.
4. Once all editing is complete, click **Save and reload** to save the changes.

## Delete Digital Solutions

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions**.
2. Locate the target digital solution, select it, and click **Delete**. A confirmation dialog is displayed.
3. Click **Yes** to confirm deleting the digital solution. The selected digital solution is now deleted.

# Configuration Data Definitions

Configuration data definitions allow advanced data exports, by defining the relevant entity attributes along with rules to mitigate duplication, referential integrity, versioning, business statuses, etc. on the destination environment at import. Thus, the data can be shared between different environments quickly and easily.

After creating a configuration data definition, you can include it in "[Configuration Data Packages](#)" on [page 1099](#) where you can select which records matching your configuration data definition you wish to export.

In general, configuration data definitions and configuration data packages are used by digital developers to transfer business configurations from one environment to another (e.g. a new version of a banking product, a new sales campaign, etc.).

To create a configuration data definition:

1. In FintechOS Studio, go to **Main Menu > Configuration Management> Configuration Data Definitions** to open the Configuration Data Definitions list.
2. Click **Insert** to open the General configuration page.

3. Fill in the following fields:

Field	Description
Name	Name of the Configuration Data Definitions settings.

Field	Description
<b>Display Name</b>	Name shown in the user interface.
<b>Master Entity</b>	<p>The entity on which the data export is built. This entity must have a unique constraint set (for details, see <a href="#">"Entity Unique Constraints" on page 94</a>).</p> <p>The unique constraint is used to identify the records and determine imported records that overlap with existing records on the target environment.</p>
<b>Mirror Collection</b>	<p>This option replaces data on the target environment with an exact copy of the export.</p> <p>For example, if the checkmark is not ticked, and in the source entity there is an attribute called <i>cash</i> and in the target entity there is an attribute called <i>card</i>, after the import, the target entity will have both attributes <i>cash</i> and <i>card</i>.</p> <p>If the checkmark is ticked, the <i>cash</i> attribute is imported the <i>card</i> attribute is removed.</p>
<b>Include Business Unit</b>	Includes the security elements set for the data.
<b>Include Business Status</b>	<p>When using <a href="#">"Business Workflows" on page 418</a> to assign statuses to entity records, the record statuses will be also exported.</p> <div> <p><b>NOTE</b></p> <p>This setting is mutually exclusive with <b>Is Versionable</b>.</p> </div>

Field	Description
<b>Is Versionable</b>	<p>If the master entity is versionable, you can mark this in the configuration data definition (see "<a href="#">Entity Cloning and Versioning</a>" on page 1261 for details).</p> <p>If checked, on import, depending on the matching records found on the destination environment (records with the same unique constraints), the following apply:</p> <ul style="list-style-type: none"> <li>• If no matching record is found, a new record in the Draft state is created.</li> <li>• If a matching record in the Draft or Version Draft state is found, the import fails.</li> <li>• If a matching record in the Approved state is found, a new record in the Version Draft state is created.</li> </ul> <div> <p><b>NOTE</b></p> <p>This setting is mutually exclusive with <b>Include Business Status</b>.</p> </div>

Field	Description
Endpoint Name	<p>Allows you to trigger a server side script with the specified endpoint name on the destination environment on import. The IDs of the inserted master entity records will be passed to the script through the <code>context.Data.packageVersionIds</code> object, for example:</p> <pre>var packageVersionIds = context.Data.packageVersionIds; for (var i = 0; i &lt; packageVersionIds.Count; i++) {     var id = packageVersionIds[i]; }</pre>
Description	Insert details about the configuration data definition here.

- Click **Save and Reload** and open the **Definition** tab to configure the master entity's relationships.

←

Save and close

Save and reload

1 General

2 Definition

Regenerate

Name	Type	Entity Name	Include	Is Hierarchy	Identification Constraint Name	
Version Settings	Root	FTOS_VersionSettings	<input type="checkbox"/>	<input type="checkbox"/>	unique_name [ name ]	Edit

Name: \*

BWstatus

Type:

Lookup Attribute

Entity Name:

BWstatus

Include:

☒

Internal Name:

statusApproved

Update Only Reference:

☐

Is Hierarchy:

☐

Hierarchy Levels:

Identification Constraint Name:

Select...

Save

Cancel

BWstatus	Lookup Attribute	BWstatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
BWstatus	Lookup Attribute	BWstatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
BWstatus	Lookup Attribute	BWstatus	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
entity	Lookup Attribute	entity	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
Version Settings Item	Details Collection	FTOS_VersionSettingsItem	<input checked="" type="checkbox"/>	<input type="checkbox"/>	unique_name_versionSettings [ name, versionSettings ]	Edit
attribute	Lookup Attribute	attribute	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
entity	Lookup Attribute	entity	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit
entity	Lookup Attribute	entity	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Edit

5. If the master entity's data model has changed since the last time you edited the configuration data definition, click **Regenerate** to update the grid.
6. Click the **Edit** button at the right side of each attribute, and fill in the following information:

Field	Description
Name	This is the name of the attribute.

Field	Description
<b>Type</b>	<p>This is the type of attribute:</p> <ul style="list-style-type: none"> <li>• Root - Master entity.</li> <li>• Lookup Attribute - Lookup attribute with a Dictionary lookup relationship type.</li> <li>• Details Collection - Lookup attribute with a IsChildOf lookup relationship type.</li> <li>• N-to-N - Many-to-many relationship.</li> <li>• Parent - Entity that references the master entity.</li> </ul>
<b>Entity name</b>	Name of the referenced entity (including the master entity).
<b>Include</b>	Tick if you wish to include the attribute in the export.
<b>Internal name</b>	It is made up of the Name + "v"+Version + exportDate.
<b>Update only reference</b>	This field is mandatory for parent-child relationship between entities.
<b>Is Hierarchy</b>	When referencing an entity with an IsChildOf relationship type, if the referenced entity includes a self-referencing lookup attribute, this box is automatically checked.
<b>Hierarchy Levels</b>	When the Is Hierarchy box is checked, use this field to set up how many levels of self-referencing are allowed for the referenced entity. Default: 4.
<b>Identification constraint name</b>	Select the constraint created earlier. For more information, see <a href="#">"Entity Unique Constraints" on page 94</a> .

**NOTE**

If the entity has lookup attributes to or is a child of a different entity, the

referenced attributes must be included in the import. Otherwise those attributes will appear in the file, but on import will be empty (reference as none).

- 7. Click **Save** to add the attribute or click **Cancel** to cancel the process.  
Repeat for as many attributes as needed.

You can now use ["Configuration Data Packages" on page 1099](#) to export the records.

# Deployment

The Deployment submenu allows you to exchange data between FintechOS Platform environments via deployment packages (deprecated), configuration data packages, and digital solutions packages.

**IMPORTANT!**  
Starting with version 22.1, deployment packages are deprecated. Instead, use ["Configuration Data Packages" on page 1099](#) or ["Digital Solutions Packages" on the next page](#).

- Configuration data packages are aimed at importing records for a specific entity. They allow you to define advanced rules to mitigate duplication, referential integrity, versioning, business statuses, etc. on the destination environment, as well as the ability to filter the import only to specific records in the data set.
- Digital solutions packages allow you to export and import entire ["Digital Solutions" on page 1077](#) in one go, enabling easy deployment of complex products. A digital solution deployment package is a deployment package associated with a specific digital solution and contains all the digital assets that are part of that digital solution.

<b>Digital Solutions Packages</b> .....	<b>1086</b>
---	-------------

Configuration Data Packages .....	1099
Deployment Package Logs .....	1102

## Digital Solutions Packages

**IMPORTANT!**

To manage digital solutions packages, you need the Release Manager security role. For more information, see ["Security Roles" on page 1245](#).

Digital solutions packages allow you to export and import entire ["Digital Solutions" on page 1077](#) in one go, enabling easy deployment of complex solutions. A digital solutions package is a deployment package associated with a specific digital solution and contains all the digital assets that are part of that solution.

On digital solutions imports, package integrity validations check for incompatible platform versions or digital assets, to ensure that the local data is not corrupted.

### Create Digital Solutions Packages

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions Packages**.
2. Click **Insert**.
3. Fill in the **Name** field, and select the source **Digital Solution** from the corresponding field.
4. Click **Save and Reload**. The new digital solution package is now created.
5. The Digital Solution Package Items grid is populated, allowing you to check the digital assets that you want to be included in the package.

#### Add security configuration to the digital solution package

1. In **Settings** tab, tick the box for **Include Security Configuration** to export the security configuration of the digital assets included in the package.

2. From the **Deployment Scenario** drop-down, choose the action that occurs when importing the digital solution package in another environment:
  - **Append**: the security configuration is added on top of the destination configuration.
  - **Overwrite**: the destination configuration is replaced with the one on the digital solution package.
3. Check the **Include Extended Security Configuration** box to include security configurations that relate to items external to your package but are linked to the same security roles.

**NOTE**

Items attached to the following default security roles are not exported: Debugger, Developer, Release Manager, Observability, JobServer, User Management. Default security roles must not be used for securing business solutions.

4. When done, click **Save and reload**.

**Configure the custom files mappings**

To configure the mappings between digital assets files and [portal profiles](#) for [custom](#) and [custom on-demand files](#), follow the below steps:

1. In the **Setting** tab, click the **Configure** button to open the **Configure Custom Files Deployment** page.

2. In the **Configure Custom Files Deployment**, select the portal profiles where you want to deploy the custom files. Perform this step for each digital asset.

### Configure Custom Files Deployment

Please select the Portals for each Digital Asset where you would like to deploy the custom files.

Select All

da1

☒ PP1

☒ PP2

☐ PP3

☐ On all Portals without Profile

da1

☐ PP1

☐ PP2

☐ PP3

Selected: 2

Cancel

Save

Please take into consideration:

- This configuration when exporting a digital solution package is merely a suggestion, it's one way of doing things. When the digital solution package is imported into the target environment, the link between the custom files in the digital solution package and the portals in that environment is prepopulated (provided that the target environment has the same portals as the source environment). However, you can choose any combination before importing.

- The meaning of the **Apply to All Portals Without Profile** checkbox is that, during import, the respective files are copied to all environments that do not have a portal profile defined (“anonymous” environments). This means they will also be copied to the default Portal and B2C Portal (provided that no portal profile has been assigned to them). If, at any point, other “anonymous” portals are created, these files will be copied there as well.
3. Click **Save** to keep your selected choices.
  4. When finished, click **Save and reload**.

Once you configure the mappings, you can [export the package](#).

#### Include identity provider UI themes

The **Settings** tab of a digital solution package comes with two options in the **Identity Provider UI Themes section**:

- **Include UI Themes**: tick to include the UI theme associated with the Portal Profile in the package. The UI theme stored in FintechOS IDP must have the same name as the Portal Profile.
- **Stop Package Generation on Themes Retrieval Error**: tick to stop the UI theme export if the environment cannot communicate with the FintechOS IDP.

Click **Save and Reload** after ticking the above boxes for the changes to take effect.

The UI theme should fit the branding or specifications of the project that you are working on. At import, the theme is automatically deployed on the environment based on the portal profile name.

If the package contains 3 portal profiles ("profile1", "profile2", "profile3") and the FintechOS IDP has 2 custom themes ("profile1", "profile2"), when the package is generated it will automatically include the themes named "portal1" and "portal2". The portal profile named "profile3" will not bring a theme into the package as there is no theme named "profile3".

## Export Digital Solutions Packages

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions Packages**.

2. Select the desired package.
3. Click **Export Digital Solution Package**.
4. Choose the save location of the package. The digital solution package is exported in a .zip file.

**NOTE**

The export includes only the security configuration of the digital assets added in the package.

## Import Digital Solutions Packages

**NOTE**

Starting with v24.3.2, the size of digital solution packages is restricted at import to the default of 50MB. Navigate to **Configuration Manager > Studio > app-settings > DigitalSolutionPackageMaxSizeInMB** key to change the default to another size.

Follow the below steps for an automated import process at digital asset level.

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions Packages**.
2. Click **Import digital solution package**. A window is displayed.
3. Locate the .zip file containing the package, select it, and click **Import**. A wizard window with three tabs displays:
  - **Solution Validation**: shows a summary of the digital assets included in the package.

1

Solution Validation

2

Digital Assets

3

Import Setting

DocsDA, DocsDA

Status of import rules

Digital Asset	Included	Version	Version On Environment	Minimum Platform Version	Environment Platform version	Validation Message	Validation Status
DocsDA	Y	1		22.1.4.1	24.1.1.0	Existing platform version is ok	✓ Passed
DocsDA	Y	1	1	22.1.4.1	24.1.1.0	Digital Asset will be updated	⚠ Warning

Cancel

Next →

- **Digital Assets:** specify the destination of the digital assets. Add them in an existing Digital Solution or in a new one. Click **Next** once done.

1

Solution Validation

2

Digital Assets

3

Import Setting

Choose where to import the Digital Assets

☒ In an existing Digital Solution

Select an application

☐ In a new Digital Solution

Name

Code

← Previous

Cancel

Next →

- **Import Setting:** specify settings:
  - tick the **Deploy audit configurations** box if needed.
  - if the **Include UI theme** option is ticked, then all themes present in the package are uploaded to the IDP by overwriting existing themes (if there are theme sharing same name) or new ones are created.
  - use the **Stop solution import on themes save error option** to stop the package import if the platform cannot communicate with the IDP.

**NOTE**

In order to successfully import Scheduled Service (v22.1.1 or later) "[Configuration Items](#)" on [page 1059](#), make sure the Scheduled Job and workflow linked to the Schedule Service exist on the destination environment.

The package is imported and can also be viewed in the Digital Solution Packages List.

**IMPORTANT!**

1. Manual changes of Digital Solution Packages are not allowed and the platform will reject any corrupted packages.

2. There is a set of import rules which a package must comply with, and which are displayed as a summary before import:

- a) Minimum platform version set on Digital Asset (DA) level in the package must be  $\leq$  platform version of destination environment. The version doesn't necessarily need to be specified in the destination environment.
- b) If DA Status on destination = Unlocked and DA version in package  $\geq$  DA version on destination, then Digital Asset will be updated.
- c) If DA Status on destination = Unlocked and DA version in package  $<$  DA version on destination, then Import cannot be performed.
- d) If DA Status on destination = Locked and DA version in package  $>$  DA version on destination, then Digital Asset will be updated.
- e) If DA Status on destination = Locked and DA version in package  $\leq$  DA version on destination, then Import cannot be performed.

Considerations at package import:

- Because of the differential deploy feature in FintechOS Studio, when reimporting a package with changes, files are marked accordingly with the Update or New status, making it easy to see which files have been changed since the last import.
- When importing a package that includes an updated entity form into a target environment where an older version of that form already exists, the system checks its child files—such as entity extensions, filtered fields, allowed attributes, and virtual attributes. If these child files haven't changed and match those in the imported package, they won't be re-imported.
- An error may be triggered after importing a package with a renamed extended model. This is a known issue in the following scenario:
  - on a source environment, create two entities with lookup attribute between them. Create a new extended model of type Related and add the virtual attributes to this extension. Export the package and import it to a destination environment. The import should work.
  - return to the source environment and modify the name of the extended model. Export the package with an updated name and reimport it on the destination environment. The import will fail with an error message.

The **workaround** implies using a before import script that will map the entity extension before package import:

```
var ee = ftos.data.query.getAlias("entityExtension")

var rows = ftos.data.query.from('entityExtension', ee)
  .where(ee.Name.eq('ext_rel'))
  .executeAndMapComplex({entityExtensionId : ee.EntityExtensionId})

if(rows.length > 0){
  setMessage(rows[0].entityExtensionId)
  setMessage("Record found. Will update");
  update('entityExtension', rows[0].entityExtensionId,
    {"name":"ext_rel_update"})
}
else{
  setMessage("Record not found.");
}
```

## Delete a Digital Solution Package

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions Packages**.
2. Select the desired package and click **Delete**. A confirmation dialog is displayed.
3. Click **Yes** to remove the Digital Solution Package.

The package is deleted and can no longer be viewed in the Digital Solution Packages List.

## Delete a Digital Solution Package

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Digital Solutions Packages**.
2. Select the desired package and click **Delete**. A confirmation dialog is displayed.
3. Click **Yes** to remove the Digital Solution Package.

The package is deleted and can no longer be viewed in the Digital Solution Packages List.

## Deployment Scenarios

When exporting digital solution packages from one environment and importing them into another, several use cases may arise, especially when considering the security roles and security role items involved in each environment, entities, as well as the available package export settings in the FintechOS Platform.

In the scenario below, we start from a scenario where the solution package has certain security roles and security role items on certain entities. When importing the package to the destination environment, several things must be taken into consideration:

- the security role item (SRI) is on the entity 3 (E3) on the source environment, but on the E1 on the destination environment;
- several SRIs are brought on to existing entities on the destination environment;
- SRI-35 is on E3 on both environments.

Find below a visual representation of this scenario:



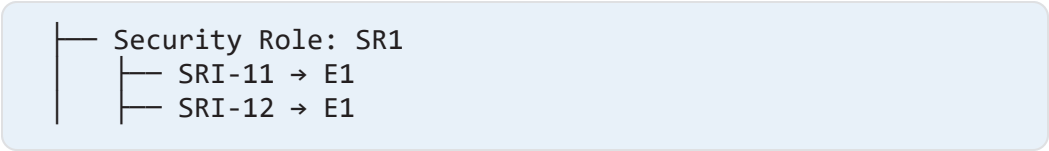
Depending on existing settings in the FintechOS Platform, several use cases emerge:

## Use case 1

When creating the digital solution package, if the following options are used:

- **Include Security Configuration** box is ticked;
- **Include Extended Security Configuration** is not ticked.

Then the following are included in the export package:

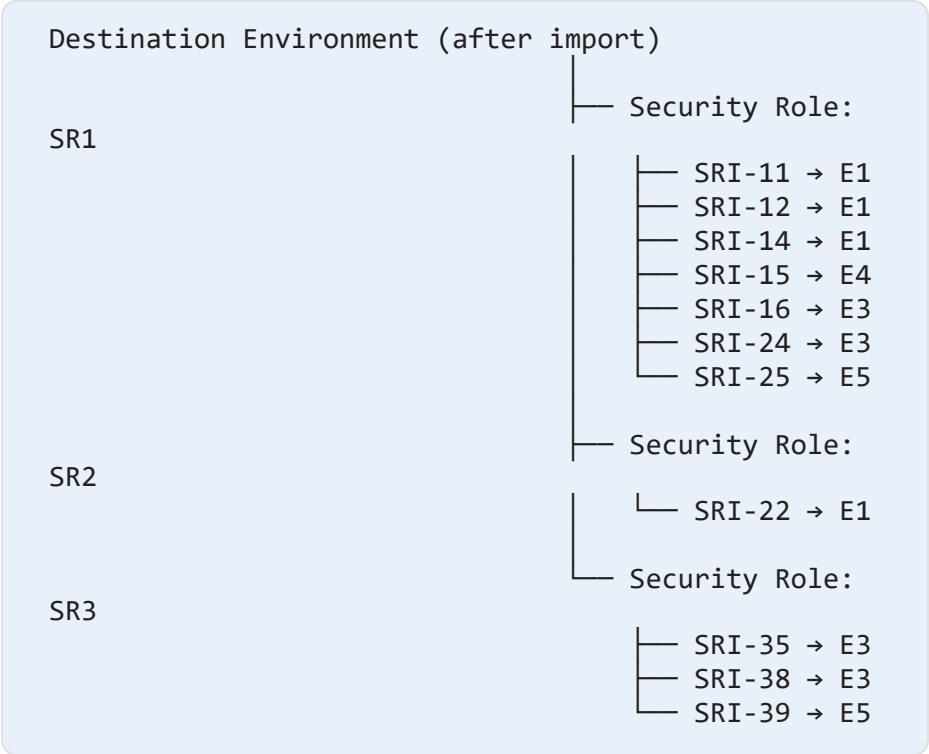


```
└─ SRI-14 → E3
└─ Security Role: SR3
    └─ SRI-35 → E3
```

## Use case 1.1

If, additionally, we select **Append**, then when importing to the destination environment, the roles that don't already exist are added, and existing ones are not modified.

The result would be:



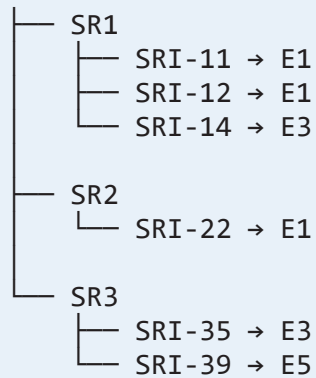
## Use case 1.2

Configure the following settings:

- **Overwrite** is selected;
- **Include Extended Security Configuration** box is not ticked.

When importing, existing items are deleted and overwritten with those in the package at the entity level. E4 and E5 are not deleted because they are not included in the package.

The result would be:



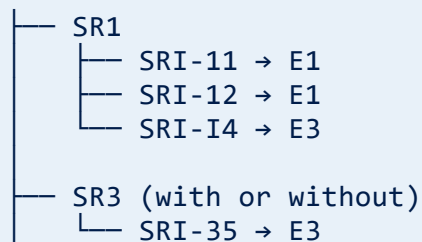
The two items on E3 are deleted and replaced with the item imported for E3. SRI-39 is imported because the SR3 from the package does not have an E5.

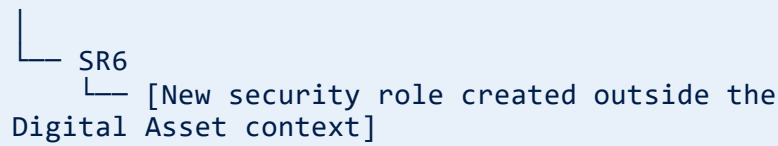
SRI - 22 on E1 is not deleted because the E1 already exists on the imported package. The overwrite acts on entity level.

## Use case 2

In this use case, the conditions are the following:

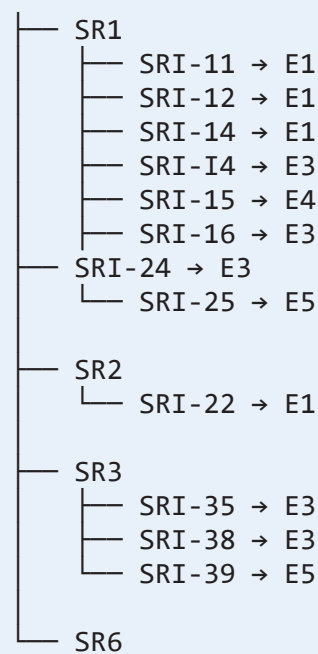
- **Include Extended Security Configuration** box is ticked;
- The digital solution is slightly changed:





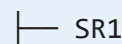
## Usecase 2.1

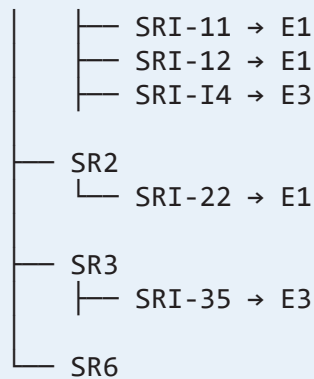
If we select **Append** and keep the **Include Extended Security Configuration** selected, then when importing to the destination environment, the security configuration is added on top of the destination configuration:



## Usecase 2.2

If we select **Overwrite** and keep the **Include Extended Security Configuration** selected, then when importing to the destination environment, the destination configuration is replaced with the one on the digital solution package:





- For SR1 (entities E1 and E3): the security role items associated with these entities will be replaced with those from the exported package. An overwrite is performed using the new information from the package.
- For SR2 and SR6: no changes will be made, as the exported package does not contain these roles. The definitions from the source package will be retained.
- For SR3: an overwrite will be performed only for the packages associated with entity E3, since the exported package does not contain security role items defined for E5.

In summary, an overwrite will be performed only for those roles and security role items that are associated with entities present in the exported package.

## Configuration Data Packages

Configuration Data Packages allow you to export and import entity records that meet specific ["Configuration Data Definitions" on page 1079](#) between different environments.

## Export a Configuration Data Package

1. In FintechOS Studio, go to **Main Menu > Configuration Management > Config Data Packages**.
2. Click **Insert**.
3. Fill in the following fields:

Field	Description
<b>Name</b>	Name of the package.
<b>Version</b>	Name + "v"+Version + exportDate.
<b>Display Name</b>	Name displayed in the user interface.
<b>Configuration Data Definition</b>	The data definition for the export (for details, see <a href="#">"Configuration Data Definitions" on page 1079</a> ). Once the configuration data package is saved, this field cannot be edited.
<b>Description</b>	Insert the details relevant to the package here.

4. Click **Save and reload**.
5. Click **Insert** in the grid at the bottom of the page and select the items (records) to be included in the package.



Add Data Config Deployment Package Item

Master entity    app001entity

+ Insert
Refresh

-
Name

☐ Georgescu
☒ Ionescu
☒ Popescu

- Click on **Export Data Config Data** to export the package.

← Save and close Save and reload Export Data Config data

Edit Configuration Data Deployment Package

Name: app001entityPackage Version: 001

Display Name: App 001 Data Deployment Package

Configuration Data Definition: app001configDataDefinitions Was Imported

Description: Data set for app001entity.

+ Insert × Delete Export Refresh

Name	Master Entity Type Id	Data	Record Id
Ionescu	app001entity	{ "name": "Ionescu" }	6931c90e-476f-4297-97d1-8dccc4...
Popescu	app001entity	{ "name": "Popescu" }	99f46b02-7edf-4030-b85f-f485aab4...

## Import a Configuration Data Package

- In FintechOS Studio, go to **Main Menu > Configuration Management > Config Data Packages**.
- Click **Import Data Config Data**.
- The local Files menu of the device opens. Select the package file to import.
- The system warns you about the source database version. Select **Yes** to continue.
- The system asks **"Do you want to deploy audit configurations (you should normally choose 'Yes')?"**. Select **Yes** if you wish to deploy it. If you do not wish to deploy it, click **No**.
- The package is successfully imported if it meets the requirements. These requirements must be met when the package is created, i.e. exported.

To check the package import:

1. Search for it by its name (to see only the imported packages, you can filter the **was imported** column to **yes**).

Configuration Data Deployment Packages list

Name	Display Name	Configuration Data Definition	Was Imported	Version
Q	Q	Q	(All)	Q
Demo	Demo	FlowSettings	(All)	22.1
FlowSettings_CognitiveClient_Example 1/1...	FlowSettings_CognitiveClient_Example	FlowSettings	Yes	2.0
FlowSettings_CognitiveClient_Example 1/1...	FlowSettings_CognitiveClient_Example	FlowSettings	No	2.0
FlowSettings_CognitiveClient_Example 1/1...	FlowSettings_CognitiveClient_Example	FlowSettings	✓	2.0
FlowSettings_CognitiveClient_Example 1/1...	FlowSettings_CognitiveClient_Example	FlowSettings	✓	2.0
FtosFaceRecAndLiveness_Examples 1/19/...	FaceRec and Liveness Examples	FlowSettings	✓	1.0

5 10 20 1 2 3

2. To open the package, double-click on its name. The fields are read-only. If the boolean **Was imported** is true, i.e. ticked, then the package was imported successfully.

Edit Configuration Data Deployment Package

Name	FtosVideoOperator_Example 1/19/2022 8:25:56 PM	Version	1.0
Display Name	FtosVideoOperator_Example		
Configuration Data Definition	FlowSettings	Was Imported	✓
Description			

Export Refresh

Name	Data	Record Id	Master Entry Type Id
Q	Q	Q	Q
Video Operator Example	{ "name": "Video Operator Example" }	396300ea-7a4e-4b4c-88ce-7793a7121982	FTOS_DFP_FlowSettings

## Deployment Package Logs

When importing a deployment package into an environment, the output of the checks performed during the import is saved into a log specific to the deployment package.

To view the log of a deployment package, in FintechOS Studio, go to **Main Menu > Configuration Management > Logs** to open the Deployment Package Logs list page.

Deployment Package Logs list	
Name	Created on
01 FTOS DFP Common.zip - Import	19/01/2022 20:03
01 Pack_LookupRel_SystemEntities - 1.0 - Copy.xml - Import	25/01/2022 11:10
02 FTOS Content Templates.zip - Import	19/01/2022 20:03
02 FTOS Foundation.zip - Import	19/01/2022 20:05
02 FTOS Project HyperPersonalization.zip - Import	19/01/2022 20:11
02 FTOS Versioning.zip - Import	19/01/2022 20:12
03 FTOS Project Campaign.zip - Import	19/01/2022 20:13
Af_ent.zip - Import	20/01/2022 09:43
BD_Data_solution_02.zip - Import	26/01/2022 10:27
BD_Digital_Solution.zip - Import	31/01/2022 17:29

The Deployment Package Logs list contains a table with two columns: the name of the deployment package (the name of the xml file), and the date when it was imported (or when a user tried to import the file).

Double-click on the desired deployment package name to open the Edit Deployment Package Log page opens.

**NOTE** You cannot edit fields in this page.

Edit Deployment package log

CUSTOMIZATIONSETLOG

Name
01 FTOS DFP Common.zip - Import

Message

LIST OF CUSTOMIZATIONSETLOGITEMS

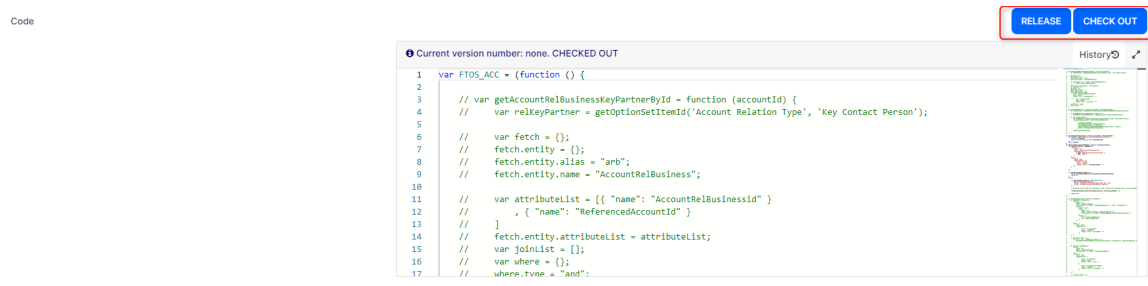
+ Insert X Delete Export Refresh

Element	Operation	Status	Message
<Entity srcXmlLnNr="13" name="FTOS_DFP_FlowSettings" displayName="Digital Flow S...	IMPORT	OK	
<Entity srcXmlLnNr="160" name="FTOS_DFP_ProcessorSettings" displayName="Proce...	IMPORT	OK	

The List of CustomizationSetLogItems section displays the status of all the deployment package components imported. If the import for a component has failed, the reason for the import failure is displayed in the Message column.

# Manage Locked Files

Some of the FintechOS Studio configuration items, such as fetch object expressions, JavaScript code, etc. are protected using a Check Out - Release mechanism to prevent concurrent access.



To view all the currently checked out configuration items, in FintechOS Studio, go to **Main Menu > Configuration Management > Manage Locked Files**.

External Id	User Name	Created On
entityformsection_b41d9e8-fa52-415f-ad72-fd5f90571bd_template	host	1/31/2022, 4:33 PM
entityformsection_fe269aca-e2ac-4fd6-9ff5-e436774ae304_beforeSaveJs	host	1/31/2022, 3:32 PM
entityform_b7afac26-c766-4d45-a2b0-1f8b4326e8cf_afterGenerateJs	host	1/31/2022, 3:24 PM
HtmlWidget_Sccba9e5-8936-4f36-adb7-e3d0fa0e8e7_Javascript	host	1/31/2022, 2:54 PM
entityform_0ad43a15-962a-490c-81d6-24fbd5c440_afterGenerateJs	host	1/31/2022, 2:45 PM
entityformsection_296f25ab-8f90-4b82-80e6-8e2381e67c89_template	host	1/27/2022, 2:11 PM
entityview_a8001f67-90ad-415f-b008-a0ce246b8b8e_fetchObjectExpression	host	1/26/2022, 10:14 PM
entityview_366eb0e0-1e76-4ef1-8a49-48157e019516_fetchObjectExpression	host	1/26/2022, 10:00 PM
entityformsection_fe269aca-e2ac-4fd6-9ff5-e436774ae304_afterGenerateJs	host	1/26/2022, 3:38 PM
entityformsection_fe269aca-e2ac-4fd6-9ff5-e436774ae304_afterSaveJs	host	1/26/2022, 3:38 PM

5 10 20

Page 1 of 3 (26 items) 1 2 3

You can use the **Unlock** button at the top right corner of the Locked Files list to force release a configuration item that has been checked out by another user.

# Advanced Developer Tools

FintechOS Studio provides developers with advanced capabilities to:

- manage HTML, CSS, and JavaScript in forms at source code level
- create and use scripts in customer journeys and forms
- create script libraries to avoid writing the same lines of codes repeatedly, resulting in increased productivity
- work with supporting assets like sequencers, code blocks, or stored procedures
- set up multi-factor authorization for specific operations
- set up B2C environments

This section covers the following topics:

---

<b>Code Editor</b>	<b>1108</b>
Open the Code Editor	1109
Code Editor Layout	1109
Manage and Edit Files	1112
Source Control	1113
Intellisense for SDK Methods	1114
Code Editor in Sandbox	1114
<b>Code Editor Development Guide</b>	<b>1115</b>
Initial Preparation	1116
Add a Change	1128
Integrate changes	1137
Build and deploy	1141
<b>Advanced Code Editor (deprecated)</b>	<b>1142</b>
Features	1143
Access the Advanced Code Editor	1143

---

General Layout .....	1143
Code Editing Supporting Features .....	1148
HTML Syntax Guidelines .....	1148
Debug Files using File History .....	1149
<b>Business Service Components .....</b>	<b>1149</b>
Create Business Service Components .....	1152
Debug Components .....	1154
Generate Endpoints .....	1155
Import or Export Components .....	1156
<b>Client Script Libraries .....</b>	<b>1156</b>
Create a Client Script Library .....	1157
Clone a Client Script Library .....	1158
Import Client Script Libraries in your Client Side Code .....	1159
<b>Server Automation Scripts .....</b>	<b>1162</b>
Event Triggered Automation Scripts .....	1163
On-Demand Automation Scripts .....	1163
Create Event Triggered Automation Scripts .....	1163
Create On-demand Server Automation Scripts .....	1168
Call Actions .....	1171
XML Support .....	1173
Fluent Queries .....	1177
Create Asynchronous Processing Flows .....	1188
Debugging .....	1190
JavaScript Engine Jint V3 .....	1201
JavaScript ES6 Support .....	1203
<b>Server Automation Script Libraries .....</b>	<b>1206</b>
Create Automation Script Libraries .....	1207
<b>Style Sheets .....</b>	<b>1209</b>
Create a New Style Sheet .....	1209
Clone a Style Sheet .....	1209

---

Apply Style Sheets to Form Driven Flows (No-Code) .....	1211
Apply Style Sheets Using Code .....	1212
Limit Style Impact to Current Form .....	1212
Overwrite Variables .....	1213
<b>Endpoints .....</b>	<b>1213</b>
Create an endpoint .....	1213
Attach a security role to an endpoint .....	1215
Customize the Input Parameters .....	1216
Customize the Output Structure .....	1219
<b>Plugin Assemblies (Deprecated) .....</b>	<b>1221</b>
1 Add the Plugin Assembly .....	1221
2 Add the IEbsPlugin Plugin .....	1222
3 Add the UI Processor .....	1222
<b>Sequencers .....</b>	<b>1222</b>
Roll the Sequencer Back or Forward .....	1225
Retrieve a Sequence Number from the Sequencer .....	1226
<b>Code Blocks .....</b>	<b>1227</b>
Define Code Blocks Categories .....	1227
Add Code Blocks .....	1228
Use Code Blocks .....	1229
<b>DB Tasks .....</b>	<b>1231</b>
1 Add DB Tasks .....	1231
2 Add Security Roles to DB Tasks .....	1232
3 Execute DB Tasks .....	1232
<b>Operation Authorization .....</b>	<b>1233</b>
1 Prerequisites .....	1234
2 Details Automation Script .....	1234
3 Pass-Code Message and Authorization Page Templates .....	1235
4 Authorization and Rejection Automation Scripts .....	1237
5 Action Authorization Type .....	1237

---

Initiate an Operation Authorization Process .....	1240
---	------

## Code Editor

The Code Editor is a modern code editor built using one of the most renowned IDEs in the industry: Visual Studio Code. This editor allows you to visualize digital solutions and edit code in isolation through a private workspace where you easily review and manage changes through remote code Github repositories.

Starting with v24.3.2, use the `feature-external-editor-allow-form-html-template-editing` key in **Configuration Manager > studio > app-features**, to restrict saving Entity forms, App data forms (tabs), or Form driven flows (wizard), in the Code Editor. Enable the key to allow editing with the Code Editor.

Configure the editor in the following way:

1. Go to Configuration Manager and in the `kv/<environment name>/<studio>/<app-features>` path, set the `feature-use-external-dev-environment` key value to:
  - 0 - edit in Studio/Monaco
  - 1 - edit in FintechOS Editor in the browser
  - 2 - edit in FintechOS Editor on desktop.
2. Create a ticket to Cloud Support to run the dedicated pipeline.

You can disable the code editor on one environment by logging into Studio with a user with **Developer** security role, navigating to the user profile and ticking the **Disable External Editor** box.

### NOTE

Check out the [Code Editor Development Tutorial](#) for steps on how to set up the editor for your environments, how to collaborate with your team, as well as guidelines for setting up your GitHub branches and way of working.

## Open the Code Editor

The Code Editor is used for editing items that are part of a Digital Asset. Therefore, it is advised that you first organize all the items you need in Digital Assets, Digital Solutions, and Digital Solution Packages.

1. In the General tab of your digital asset, click the **Open in Editor** button.
2. Select the corresponding digital solution and the digital solution package. If the digital asset is not included in any digital solution, one is automatically created.

### HINT

To open the editor successfully, enable pop-up windows in your browser.

## Code Editor Layout

### IMPORTANT!

The edit function is available only for CSS, HTML and JavaScript files (not metadata). Starting with v24.1.1, the Code Editor comes with SCSS support. Simply open your digital asset in the Code Editor and edit files with the .scss extension.

## Explorer Structure

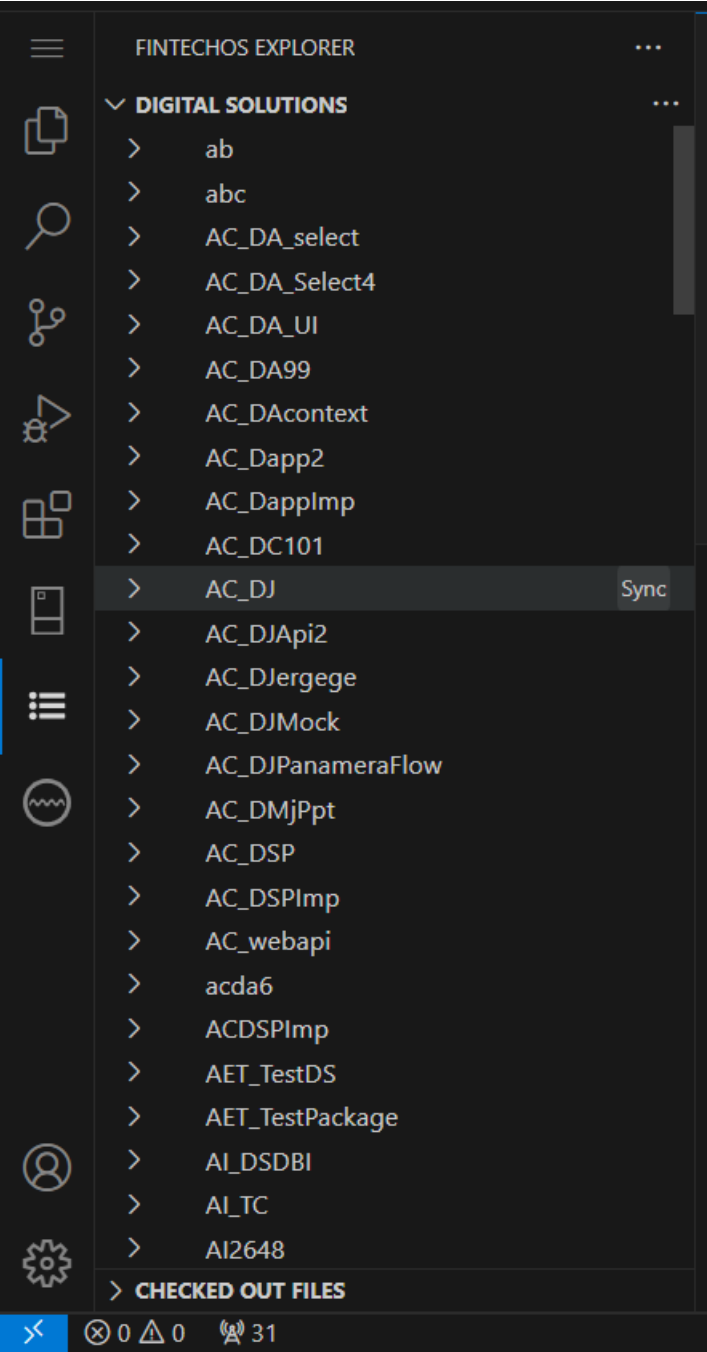
Section	Description
Workspace	<p>Section where the digital asset and digital solution files are centralized. There are two main folders:</p> <ul style="list-style-type: none"> <li>Assets where the selected <a href="#">digital asset</a> from your environment is listed. Each digital asset folder contains different folders depending on its structure: all configuration items are listed in their corresponding folder whereas the items attached to it, i.e database objects, custom and custom on demand files, data import templates, and data configuration packages are listed under the resources folder. <code>beforeImport</code> and <code>afterImport</code> javascript files are listed under the <code>_importOrchestration</code> subfolder.</li> <li>Solutions where the selected <a href="#">digital solution</a> configuration details and deployment settings are listed.</li> </ul>
Outline	Allows easy navigation between file sections through a short overview of the file tree.
Timeline	Local version control system that lets you keep track of the changes you make to your files
SonarLint Issue Location	Shows issues in your code reported by SonarLint.

### HINT

In case the business solution you are shipping is split in more than one Digital Solution, use the FintechOS Explorer to bring additional sources.

## FintechOS Explorer

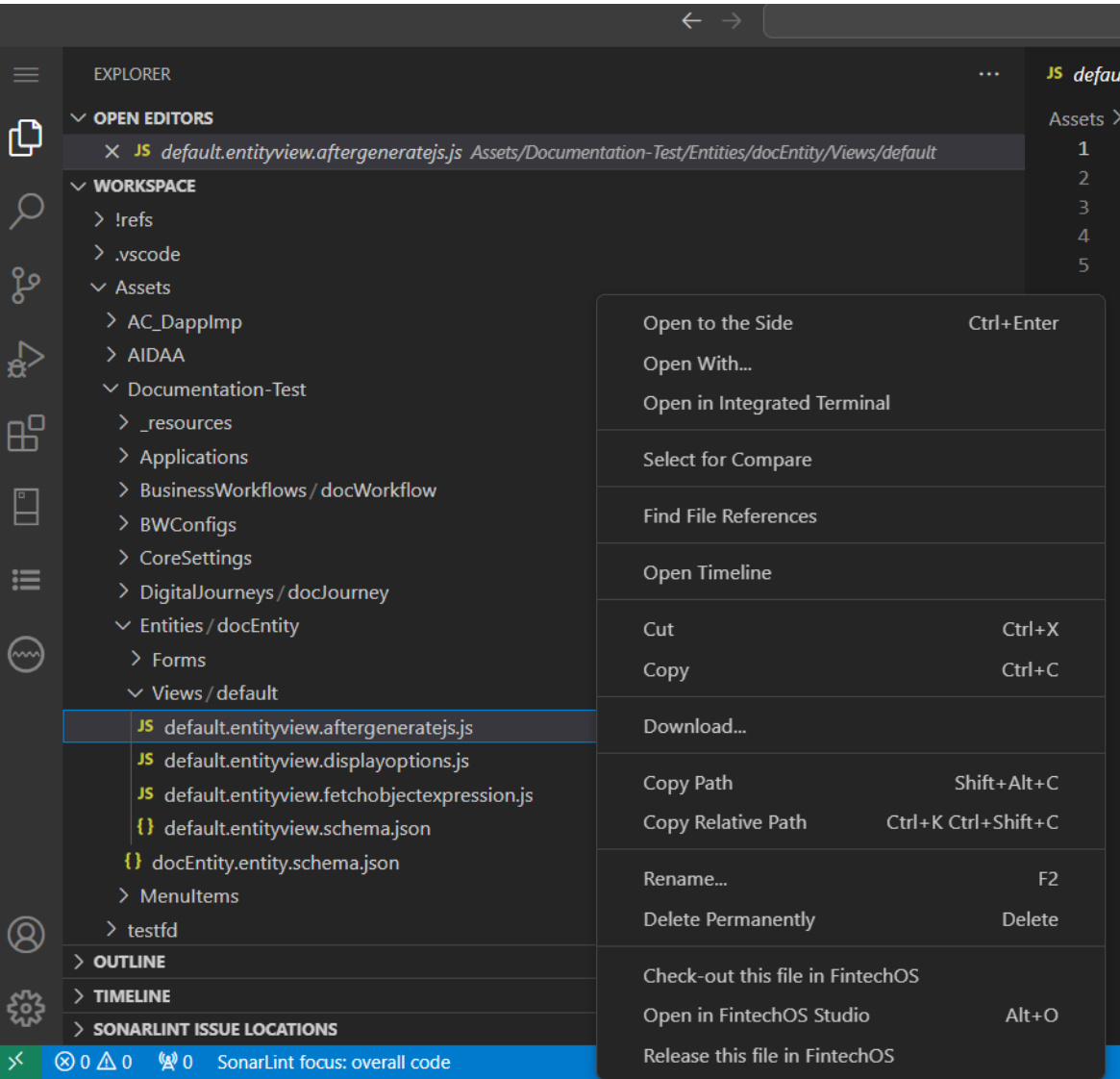
Use the FintechOS Explorer menu item to add new digital solution packages to your workspace or update existing ones. Click the **Sync** button next to the desired digital solution package to bring it up to the latest version or add other digital solution package from your environment to the Code Editor instance.



## Manage and Edit Files

The editing mechanism of Code Editor allows you to edit all Javascript, HTML and CSS files while keeping the check-out and release features present in the FintechOS Platform embedded code editor.

Right-click the item in the workspace to open the following controls:



Control	Description
Check out this file in FintechOS	Use this option to lock the file and enable editing.
Release this file in FintechOS	Use this option to release the file. You cannot edit the file after release.
Open in FintechOS Studio	Use this option to open the element in FintechOS Studio.

## Source Control

Manage digital solution package versions and track changes to the digital asset components to ensure all users can easily work on simultaneously on items and trigger an easy code review process through branching and version control. The following steps describe creating a remote repo and committing your local digital package.

1. Create a remote code Github repository for your digital solution package.
2. Initialize a local repository by running the following Git commands in the terminal of Code Editor: `git init` and `git remote add origin <remote repo URL>`.
3. Use `run git fetch origin` to fetch the remote repository and then `run git pull origin main --allow-unrelated-histories` to keep the existing files in the local repository.
4. Fix merge issues, then use the `git add` command, `git commit -m "Merged local folder with GitHub repository"`, then `git push origin main`. Alternatively, use the **Commit** button from the Source Control tab to perform an initial commit of the digital solution to the newly created repository.

### Steps:

```
git init
git remote add origin <repository_url>
git fetch origin
git pull origin main --allow-unrelated-histories
fix merge issues
git add
git commit -m "Merged local folder with GitHub repository"
git push origin main
```

For more in-depth details on how to set up a Git repository, branching, and source control, check the [Visual Studio Code user guide](#).

## Intellisense for SDK Methods

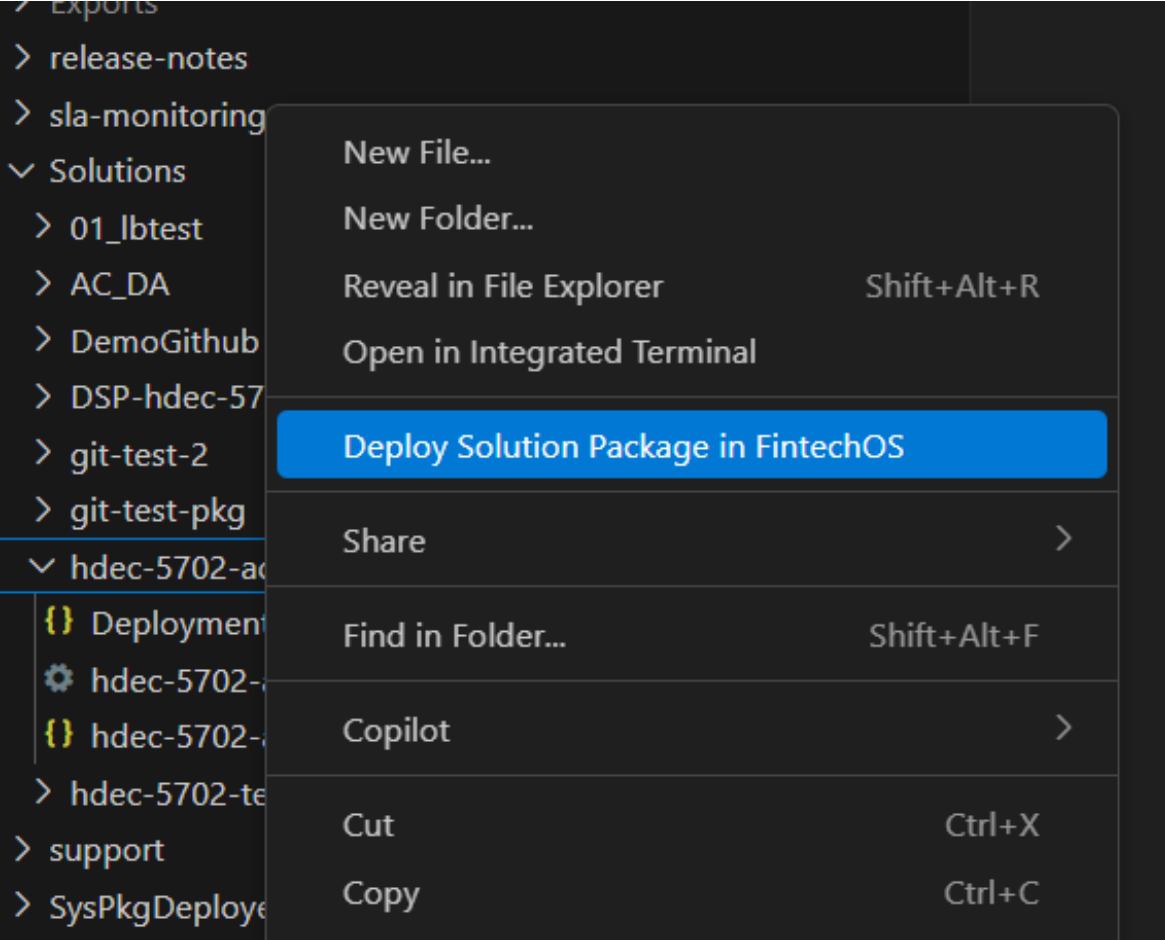
You can use intellisense in the Code Editor and get the definitions for your SDK methods.

To do so, add the `ftos.data.query.getAlias('property')` in your server automation script either in Studio or directly in the Code Editor. For example, you can add `ftos.data.query.getAlias('entityName')` and in Code Editor to see and use the properties of an entity.

Keep in mind to sync the script in the Code Editor to use the definitions.

## Code Editor in Sandbox

The Code Editor is also available with the [local developer sandbox](#), a local instance of the FintechOS Platform (v24.4.0) for building, testing, and syncing your development work. After setting up your local sandbox, you may notice the option **Deploy Solution Package in FintechOS**. Use this option after you've made changes that you want to deploy to Studio. Right-click the file and click **Deploy Solution Package in FintechOS** to include it in a solution package and deploy to Studio.



# Code Editor Development Guide


The Code Editor is your own workspace, leveraging the native capabilities of Visual Studio Code. It empowers you to collaborate on Digital Solutions and embrace a superior development approach.

The Code Editor is designed for developers and requires some familiarity with Visual Studio Code, GitHub, Sonar, Jira, and other tools.

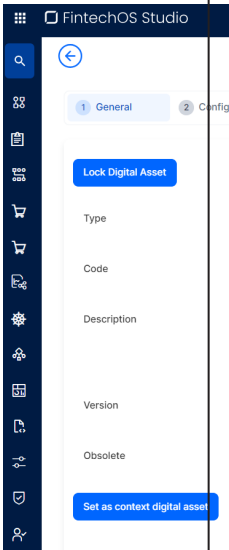

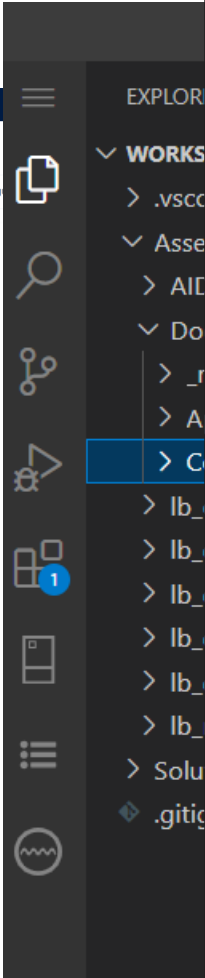
**NOTE**  
Follow the steps in the [Code Editor](#) documentation page to enable the editor on your FintechOS environment.

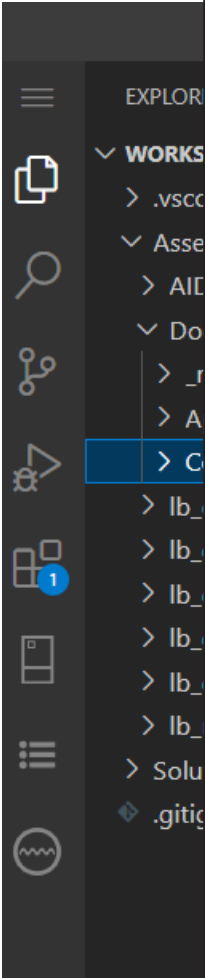
# Initial Preparation

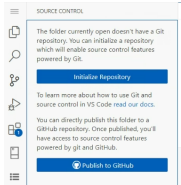
You need to complete a number of actions to prepare the workspace before you actually start working.

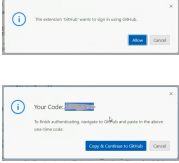
Step	Action	FintechOS Studio	Code Editor	GitHub
1.	Create a project structure including digital asset/s and digital solution.	<div>Follow the steps in the <a href="#">Configuration Management</a> page.</div> <div></div> <div><i>Digital Solution details page</i></div>		

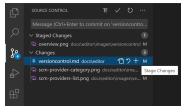
Step	Action	FintechOS Studio	Code Editor	GitHub
2.	<p>Create a new remote code repository using the FintechOS template, containing the solution build and other critical workflows. Template details and why it's needed can be found in the <a href="#">repository</a>.</p> <div><p><b>NOTE</b></p><p>Follow the steps in the following guide to connect to the <a href="#">FintechOS - GIT organization</a>.</p><p>You can skip this step if you already have a GitHub repo.</p></div>			<a href="#">Create a new repo</a>

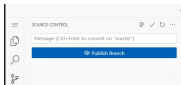
Step	Action	FintechOS Studio	Code Editor	GitHub
3.	<p>Perform an initial synchronization of your digital solution from FintechOS Studio into the Code Editor local workspace.</p> <p>Synchronization is based on package download, so you could be prompted to choose the package you are interested in.</p>	 <p><i>Open Digital Asset in Code Editor</i></p>  <p><i>Choose an existing solution and/or package</i></p>	 <p><i>Digital Solution structure displayed in editor</i></p>	

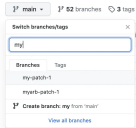
Step	Action	FintechOS Studio	Code Editor	GitHub
4.	If additional Digital Solutions are necessary in your context, you can sync using the FintechOS Explorer extension.		 <p><i>Sync Digital Solution from VS Code</i></p>	

Step	Action	FintechOS Studio	Code Editor	GitHub
5.	Run Git commands to initialize local repo.		<p>Run <code>git init</code> in Terminal or use the dedicated button in the Source Control panel.</p> 	
6.	Connect local repo to remote repo in Github.		Run <code>git remote add origin repo_URL</code> in Terminal.	
7.	Fetch origin.		Run <code>git fetch origin</code>	
8.	Keep the existing files in the local repository.		Run <code>git pull origin main --allow-unrelated-histories</code>	


Step	Action	FintechOS Studio	Code Editor	GitHub
9.	Sign in to Github and authorize the Code Editor.			

Step	Action	FintechOS Studio	Code Editor	GitHub
10.	Perform an initial commit on local and publish to remote repo.		<p>Staging (<code>git add</code>) and unstaging (<code>git reset</code>) can be done via contextual actions in the files or by drag-and-drop.</p>  <p>You can type a commit message above the changes and press <code>Ctrl+Enter</code> (macOS: <code>⌘ +Enter</code>) to commit them. If there are any staged changes, only those changes will be committed. Otherwise, you'll get a prompt asking you to</p>	

Step	Action	FintechOS Studio	Code Editor	GitHub
			<div>select what changes you'd like to commit and get the option to change your commit settings.</div> <div></div>	

Step	Action	FintechOS Studio	Code Editor	GitHub
11.	Create a new pull request			<div><div><div>1. In GitHub navigate to the main page of the repository.</div><div>2. In the <b>Branch</b> menu, choose the branch that contains your commits.</div><div></div><div>3. Above the list of files, in the yellow banner, click <b>Compare &amp; pull request</b> to create a pull request for the associated branch.</div></div></div>

Step	Action	FintechOS Studio	Code Editor	GitHub
				<div>4. Use the <i>base</i> branch dropdown menu to select the branch you'd like to merge your changes into, then use the compare branch drop-down menu to choose the topic branch you made your changes in.</div> <div>5. Type a title and description for your pull request.</div> <div>6. To create a pull request that is ready for</div>

Step	Action	FintechOS Studio	Code Editor	GitHub
				<div>review, click <b>Create Pull Request.</b></div> <div></div>

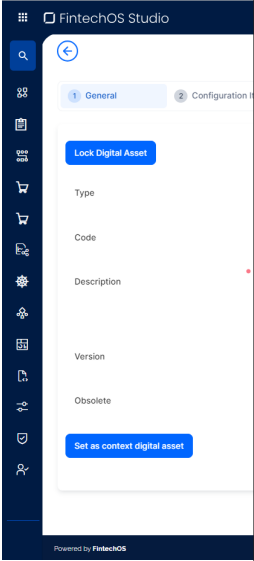
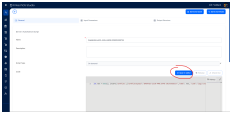
Step	Action	FintechOS Studio	Code Editor	GitHub
12.	Merge changes			<ol style="list-style-type: none"><li>1. Under your repository name, click <b>Pull requests</b>.</li><li>2. In the <b>Pull Requests</b> list, click the pull request you'd like to merge.</li><li>3. Scroll down to the bottom of the pull request. Depending on the merge options enabled for your repository, you can <a href="#">Merge all of the commits into the base branch</a> by clicking <b>Merge pull request</b>.</li></ol>

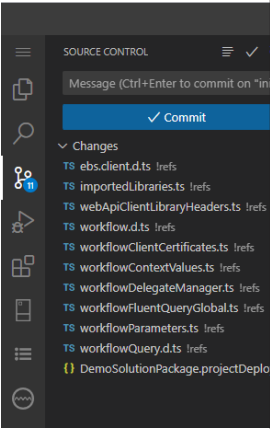
Step	Action	FintechOS Studio	Code Editor	GitHub
				

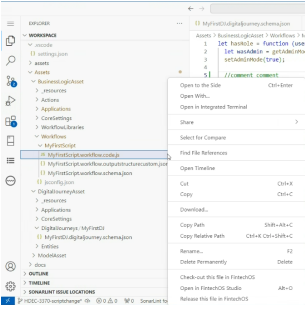
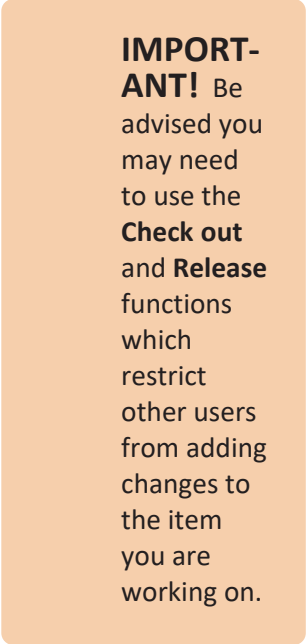
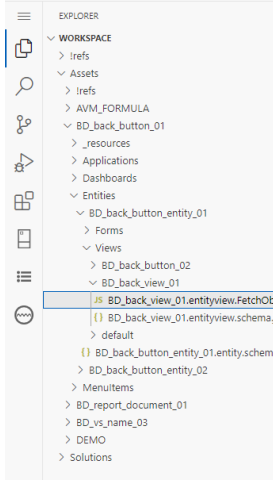
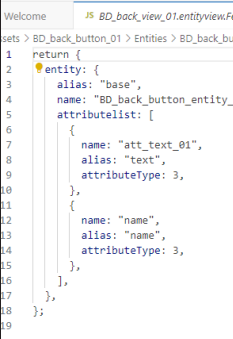
## Add a Change

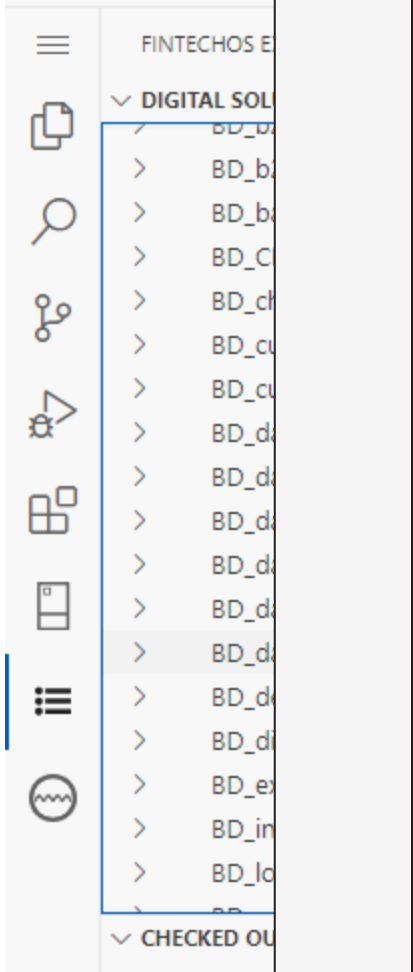
After completing the steps above, you can start writing code for a FintechOS solution. To do so, follow the steps below:

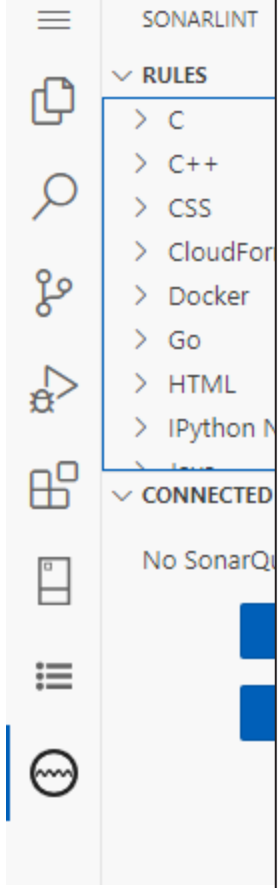
Step	Action	FintechOS Studio	Code Editor	GitHub
1.	Create the no-code configuration (server-side automation script, etc.)	<ul style="list-style-type: none"> <li>Create metadata for the modifications you are planning</li> <li>Create the file(s) for the modifications you are starting</li> </ul>		

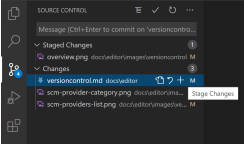
Step	Action	FintechOS Studio	Code Editor	GitHub
2.	Launch the Code Editor	<div></div> <p><i>Digital Asset</i></p> <div></div> <p><i>Automation Script</i></p>		

Step	Action	FintechOS Studio	Code Editor	GitHub
3.	<p>Create a new feature branch for your planned modifications.</p> <p>We advise you <a href="#">reference the ticket</a> you are working on so development information is fed back into Jira.</p>		<div></div> <p>Create new branch</p> <p>In Terminal, run <code>git checkout -b JRA-123-&lt;branch-name&gt;</code></p>	



Step	Action	FintechOS Studio	Code Editor	GitHub
4.	<p>Edit the desired item (script, stylesheets or HTML templates) and others. Changes are saved both locally and in the FintechOS database.</p> <div><p><b>IMPORTANT!</b> Be advised you may need to use the <b>Check out</b> and <b>Release</b> functions which restrict other users from adding changes to the item you are working on.</p></div> <div><p><i>Release and Check out</i></p></div>		<div><p><i>Fetch Expression in editor</i></p></div>	<div></div>
5.	<p>Test and debug changes.</p>	<p>Run the solution using FintechOS Portal or OpenAPI.</p>		

Step	Action	FintechOS Studio	Code Editor	GitHub
6.	Sync Code Editor with FintechOS Studio to ensure no impact from other modifications added by others.			

Step	Action	FintechOS Studio	Code Editor	GitHub
7.	Run local code quality checks (Sonar)		 <p>Connect to your team's Sonar server</p>	

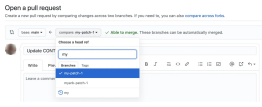
Step	Action	FintechOS Studio	Code Editor	GitHub
8.	If requirements are met, commit changes to local branch.		<p>Prepare the commit by choosing the only the files you are interested in changes. More details <a href="#">here</a>.</p>  <p>You can type a commit message above the changes and press <b>Ctrl+Enter</b> (macOS: <b>⌘ +Enter</b>) to commit them. If there are any staged changes, only those changes will be committed. Otherwise, you'll get a prompt asking you to select what changes you'd like to commit and get the option to change your commit settings.</p>	

Step	Action	FintechOS Studio	Code Editor	GitHub
			<div><b>NOTE</b> The commit action should include the Jira ID for cross-reference between Jira task and Git change: <code>git commit -m "JIRA-123 &lt;summary of commit&gt;"</code></div>	

Step	Action	FintechOS Studio	Code Editor	GitHub
9.	If everything is OK push your changes to the remote feature branch.		<p>There is a <b>Synchronize Changes</b> action in the Status Bar, next to the branch indicator, when the current checked out branch has an upstream branch configured.</p>  <p><b>Synchronize Changes</b> pulls remote changes down to your local repository and then push local commits to the upstream branch.</p>  <p>If there is no upstream branch configured and the Git repository has remotes set up, the <b>Publish</b> action is enabled. This will let you publish the current branch to a remote.</p>	

## Integrate changes

These are the generic steps for merging your changes into the codebase of a FintechOS solution.

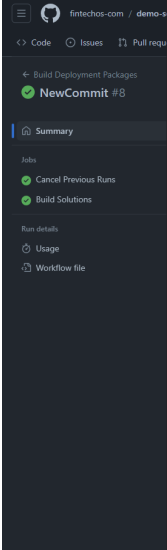

Step	Action	FintechOS Studio	Code Editor	GitHub
1.	<p>Create <b>Pull Request</b> and add a reviewer to your changes. The review must be done by another team member.</p> <div><p><b>NOTE</b></p><p>If the review or quality analysis reveals that changes are required, the flow will revert a to previous step of code changing until all reviews and quality checks are passed.</p></div>			<div></div> <ol style="list-style-type: none"><li>On <b>GitHub</b>, navigate to the main page of the repository.</li><li>In the <b>Branch</b> menu, choose the branch that contains your commits.</li><li>Above the list of files, in the yellow banner, click <b>Compare &amp; pull request</b> to create a pull request for the associated branch.</li><li>Use the <i>base</i> branch dropdown menu to select the branch you'd like to merge your changes into, then use the <i>compare</i> branch drop-</li></ol>

Step	Action	FintechOS Studio	Code Editor	GitHub
				<p>down menu to choose the topic branch you made your changes in.</p> <p>5. Type a title and description for your pull request.</p> <p>6. To create a pull request that is ready for review, click <b>Create Pull Request</b>.</p>

Step	Action	FintechOS Studio	Code Editor	GitHub
2.	<p><b>Merge PR in main branch</b> once everything is ok and create a new build.</p> <ul style="list-style-type: none"><li>• If there are conflicts between PR changes and current status of the main branch, these will need to be resolved before getting the changes in the main branch</li><li>• Conflicts will be solved by the developer manually in Git or VS Code depending on the complexity</li></ul>			<div><div><div><div><div></div><div>github / docs</div></div><div><div>&lt;&gt; Code</div><div>Issues 46</div><div><div><div></div><div>Pull requests</div><div>19</div></div></div><div>Discussion</div></div></div></div><div><div><div>1. Under your repository name, click <b>Pull requests</b>.</div><div>2. In the <b>Pull Requests</b> list, click the pull request you'd like to merge.</div><div>3. Scroll down to the bottom of the pull request. Depending on the merge options enabled for your repository, you can merge all of the commits into the base branch by clicking <b>Merge pull request</b>.</div></div><div><div><div>Merge pull request</div><div>You can also open this in GitHub Desktop</div></div><div><div><div>✓ Create a merge commit</div><div>All commits from this branch will be added to the base branch via a merge commit.</div></div><div><div>Squash and merge</div><div>The 1 commit from this branch will be added to the base branch.</div></div><div><div>Rebase and merge</div><div>The 1 commit from this branch will be rebased and added to the base branch.</div></div></div></div></div></div>

# Build and deploy

Follow the steps below to use Git actions on your repo, then build and deploy the code to an environment.

Step	Action	FintechOS Studio	GitHub	Code Editor
1.	<div>The repo <a href="#">template</a> includes the Git actions needed to create the build and (if properly configured) also promote to your desired environment.</div> <div><b>IMPORTANT!</b> To mitigate potential inconsistencies, GitHub should be viewed as single source of truth for generating and deploying packages.</div>		<div></div> <div>Build process</div> <div>Build artifact</div>	
2.	Run deploy workflow		<div>Go to actions and choose <b>Deploy to ENV</b></div> <div></div>	

## Advanced Code Editor (deprecated)

The Advanced Code Editor provides FintechOS Platform developers with a simple and powerful interface that allows them to insert and edit HTML, CSS and JavaScript forms' content using code.

### IMPORTANT!

The Advanced Code Editor is deprecated. Use the [Code Editor](#) instead.

The table below lists the attributes that can be inserted or edited using the Advanced Code Editor.

Entity	Attribute
EntityForm	After Events
EntityForm	Before Events
EntityForm	Template
Entity Form Field	Attribute Change Event
Entity Form Section	After Events
Entity Form Section	After Section Save
Entity Form Section	Before Section Save
Entity Form Section	Template
EntityView	Fetch Object Expression
EntityView	After Generate Js
EntityView	Display Options
Automation Script	Code
Automation Script Libraries	Code
Client Script Library	Definition
Client Script Library	Code
Custom Form (Custom Flow)	After Generate Js
Custom Form (Custom Flow)	Template
Html Widget	JavaScript
Html Widget	Html
Style Sheet	Code

## Features

- Browsing files and nodes
- Searching for specific nodes or specific content in files
- Simplified code editing using code snippets
- Live preview of HTML files
- Debugging right from the editor
- Insert and customize ["Code Blocks" on page 1227](#)

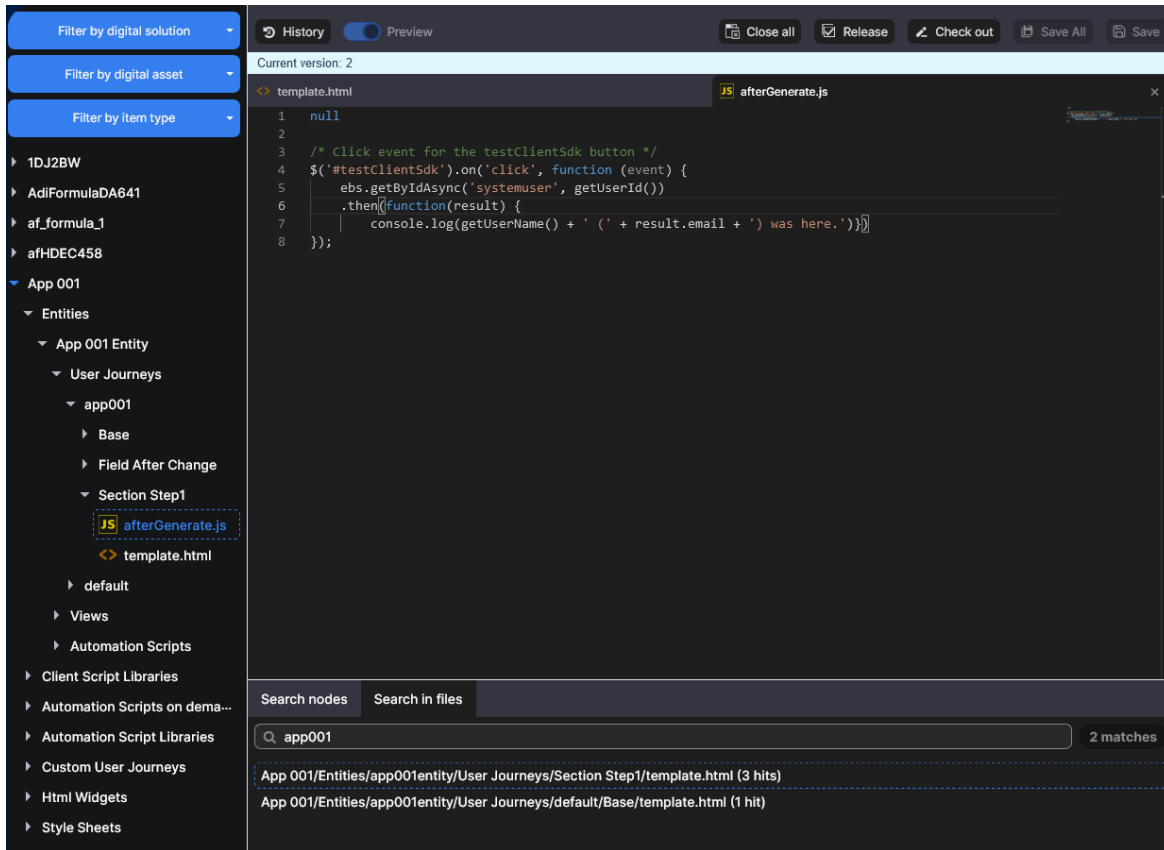
## Access the Advanced Code Editor

You can access the Advanced Code Editor in two ways:

- From the menu, click **Advanced > Advanced Code Editor**. By doing so, you will see all files and nodes available in FintechOS Studio.
- When editing one of the entities listed in the table above, click the **Open in Code** icon at the top-right corner of the page. By doing so, you will be able to browse the files of the current entity.

## General Layout

The Advanced Code Editor user interface is comprised of the following panels: an explorer on the left, showing all of the folders and files you have access to, main editor in the center, showing the content of the files you have opened, a property list on the right, showing the values of the attributes from the file you have opened, a toolbar on top and two search tabs at the bottom.



## Files Explorer

Displays in a tree-view the files you have access to, and you can browse and select the attribute files you want to edit. The files are organized in folders, each in its own Digital Asset, if available. The filters also allow you to display independent items (i.e. digital assets not set as context). If locked in a digital asset, that digital asset will have the drop-down arrow highlighted in blue in the list. The files can be filtered on three levels, as follows:

## By Digital Solution

Digital Solutions allow you to group multiple digital assets for easy distribution and deployment. For more information on how to create, edit, and delete digital solutions, see "[Digital Solutions](#)" on page 1077.

## By Digital Asset

Digital Assets group together configuration items (entities, attributes, libraries, digital journeys, etc.) that belong to a common context. For more information on digital assets, see ["Configuration Management" on page 1053](#).

## By Item Type

The Item Type filter lists all the configuration item types of a digital asset, that are editable with the Advanced Code Editor functionality (Customer User Journeys and HTML Widgets cannot be edited).

### NOTE

If an entity belonging to one digital asset has configuration items shared with another digital asset, the entity will be displayed also in the second digital asset tree, so that its configuration items can be shown in its context.

## Toolbar

The toolbar displayed on top provides the controls to perform basic operations like saving or closing file(s) and also debugging the code directly in the editor or live previewing the HTML files. The following controls are available.

Control	Description
History	Displays the version history of the currently opened file. See <a href="#">"Debug Files using File History" on page 1149</a> for information on how you can use the file history to debug your code.
Preview	Available only for HTML files, allows you to toggle between the source code or the live preview of the HTML file. When choosing live preview, below the HTML source code, a panel is displayed within the main editor showing how the HTML file will look like in the UI. For more information, see <a href="#">"Previewing HTML Files" on page 1147</a> .
Close all	Closes all opened files. When you open multiple files, they are displayed as tabs.

Control	Description
Release	Releases the file and disables file editing (see <a href="#">"Manage Locked Files" on page 1104</a> for details).
Check out	Locks the file and enables editing (see <a href="#">"Manage Locked Files" on page 1104</a> for details).
Save all	Saves all opened files. You can also save all opened files by pressing <b>CTRL+SHIFT+S</b> .
Save	Saves the file you're currently working on. You can also save the file you're working on by pressing <b>CTRL+S</b> .

## Search Nodes

The tab allows you to search available nodes by name. Enter the name of the node you want to search for and press **Enter**. The search returns the list of nodes partially matching the name of the node you provided.

## Search in Files

The tab allows you to search for specific content within the available files. Enter the content to search for (e.g., an attribute or a function name) and press **Enter**. The search returns the list of nodes which contain the content you provided.

## Properties List

For some of the nodes a list of properties is displayed, on node selection, on the right-side panel.

DEFAULT (FORM)	
Is Default	true
Is Default Edit	false
Show Tooltips Value	USER_SETT
Auto Generate Template	false
Auto Generate Template Type	Inherit
Sticky Header Items	false
Wizard Mode	false
Render Section Tabs As Bullets	false

## Previewing HTML Files

To preview an open HTML file, from the toolbar toggle the **Preview** button on.

Below the preview panel, a toolbar with three colored icons is available. The icons allow you to perform specific actions, as follows:

- Green icon - maximize the live preview to the main editor panel.

To maximize the preview panel even more (by hiding the search tabs), click the down-arrow.

- Orange icon - minimize the live preview.
- Red icon – close the live preview. The **Preview Mode** button is automatically toggled off.

## Code Editing Supporting Features

Browse or search for the files you want to edit and open them by clicking on the files. Edit the code which defines how and what the files should do and save the changes. You can also insert predefined ["Code Blocks" on page 1227](#) and change them to best suit your needs.

To write the code faster and typo-free, use the built-in IntelliSense suggestions and the ["Code Snippets Support" on page 354](#).

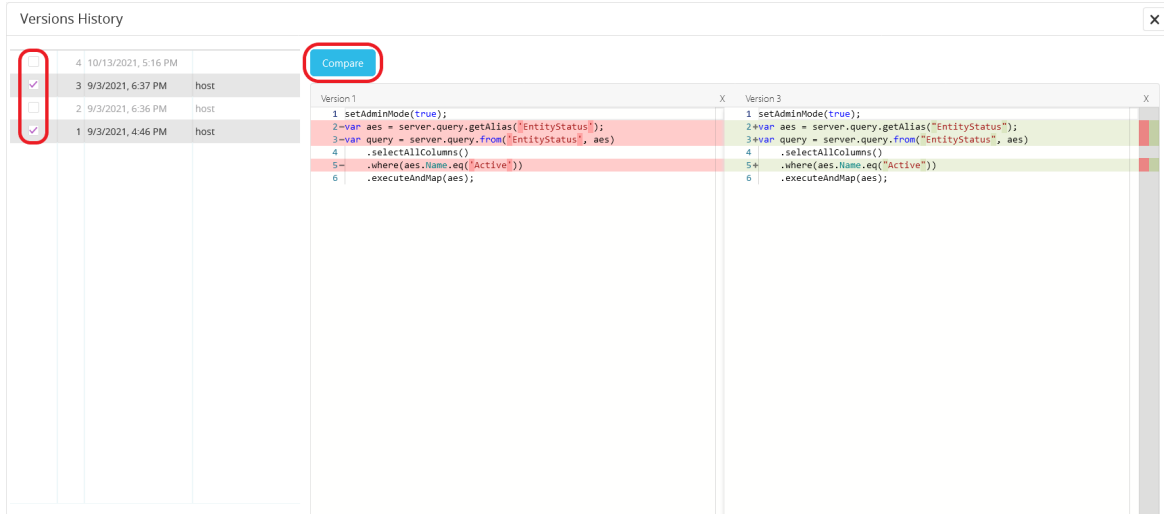
## HTML Syntax Guidelines

- The `<head>` and `<style>` tags are restricted.
- `<!-- Write your comments here -->` tags are removed.
- Preferably use `<br>` instead of `<br />` for the line break tag and `<img>` instead of `<img />` for insert image.
- Only inline CSS is supported.

### HINT

You can use an external **CSS inliner** tool in order to restructure your HTML code to inline CSS.

## Debug Files using File History



To debug a file, open it and click the **History** button in the toolbar. The panel on the left shows the saved file versions in descending order, who made the changes, and the date and time of the change. To compare two versions of your code, select the checkboxes next to the desired versions and click the **Compare** button. The panel on the right will show a side-by-side comparison of the two versions with changes highlighted for easy tracking.

## Business Service Components

Configured and managed in Studio, business service components help with a better organization of server automation code in medium and large projects. These components allow developers to continue using workflows and workflow libraries.

A server side component has 3 JavaScript modules or code files:

- **Data Service:** used for defining functions, inserting or updating data, calling data, run database queries or execute database tasks.

- **Business Logic:** this layer contains the core logic of the code and supports SDK methods for importing libraries, uploading files, calling reports, changing business status, and more.
- **Routes:** used to import modules from other server side components by referencing the module name.

**NOTE**

FintechOS v24.2 comes with the possibility to switch to Jint v3, from Jint v2. Before doing this, take into consideration the following mandatory changes. Enable Jint v3 from the sys-workflowengine-type [system parameter](#).

## Using SDK Methods in Modules

When using SDK methods in business service components, it is recommended to follow the segregation from below:

## SDK Segregation

SDK Method	Data Service	Business Logic	Routes
<a href="#">ftos.data.insert</a>	✓		
<a href="#">ftos.data.update</a>	✓		
<a href="#">ftos.data.delete</a>	✓		
<a href="#">ftos.data.getById</a>	✓		
<a href="#">ftos.data.getByQuery</a>	✓		
<a href="#">ftos.data.getByQueryAndCache</a>	✓		
<a href="#">ftos.data.updateBusinessUnit</a>	✓		
<a href="#">ftos.data.importFile</a>	✓		
<a href="#">ftos.data.getNextSequenceNumber</a>	✓		
<a href="#">ftos.documents.generateById</a>		✓	
<a href="#">ftos.documents.generateByName</a>		✓	
<a href="#">ftos.files.mergeAsPdf</a>		✓	
<a href="#">ftos.files.upload</a>		✓	
<a href="#">ftos.files.download</a>		✓	
<a href="#">ftos.files.listFolder</a>		✓	
<a href="#">ftos.files.move</a>		✓	
<a href="#">ftos.files.recycle</a>		✓	
<a href="#">ftos.files.addToArchive</a>		✓	
<a href="#">ftos.files.getBase64String</a>		✓	

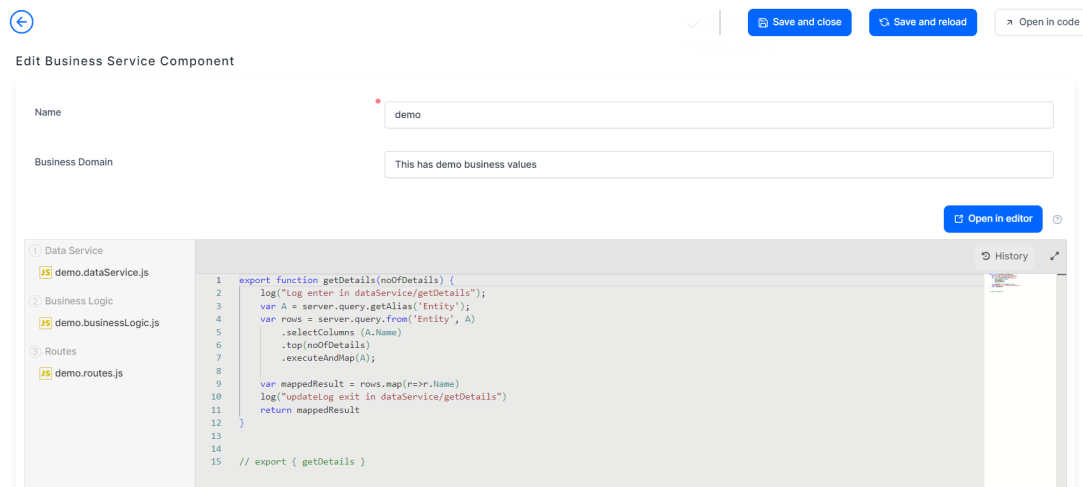
SDK Method	Data Service	Business Logic	Routes
ftos.convert.base64.toPdf		✓	
ftos.files.pdf.getPagesCount		✓	
ftos.convert.image.toPdf		✓	
ftos.convert.docx.toPdf		✓	
ftos.files.pdf.convertPageToImage		✓	
ftos.files.pdf.insertImage		✓	
ftos.files.pdf.writeText		✓	
ftos.files.pdf.writeTextStamp		✓	
ftos.files.pdf.getPages		✓	
ftos.files.pdf.flatten		✓	
ftos.files.pdf.optimize		✓	
ftos.files.pdf.encrypt		✓	
ftos.files.pdf.setPdfType		✓	
ftos.files.pdf.replaceTokens		✓	
ftos.files.pdf.merge		✓	
ftos.files.pdf.findMatchingRows		✓	
ftos.businessStatusWorkflow.update		✓	
ftos.businessStatusWorkflow.getId		✓	
ftos.businessStatusWorkflow.getName		✓	
ftos.businessStatusWorkflow.getStatuses		✓	
ftos.businessStatusWorkflow.checkCondition		✓	
ftos.businessStatusWorkflow.testCondition		✓	
ftos.convert.date.toLocalDateAsString	✓	✓	✓
ftos.convert.date.toUtcDateAsString	✓	✓	✓
ftos.utils.getLocalizedString	✓	✓	✓
ftos.businessStatusWorkflow.wasUpdated	✓	✓	✓
ftos.context.setAdminMode	✓	✓	✓
ftos.context.getAdminMode	✓	✓	✓
ftos.logging.log	✓	✓	✓
ftos.context.propertyBag.set	✓	✓	✓
ftos.context.propertyBag.get	✓	✓	✓
ftos.logging.clearProperty	✓	✓	✓
ftos.logging.isPropertySet	✓	✓	✓
ftos.utils.getSystemParameterByName	✓	✓	✓
ftos.utils.getAppSetting	✓	✓	✓

SDK Method	Data Service	Business Logic	Routes
<a href="#">ftos.metadata.getOptionSet</a>	✓	✓	✓
<a href="#">ftos.metadata.getOptionSetItemId</a>	✓	✓	✓

## Create Business Service Components

Business Service Components can be created and managed in Studio or the Code Editor:

1. In **Studio**, go to the main menu and access **Advanced > Business Service Components**. A list of all available components opens.
2. Click **Insert** to create a new business service component. The **Add Business Service Component** page opens.
3. Add a name for the new component and a business domain.
4. Add your code in the **Data Service**, **Business Logic**, and **Routes** sections.



5. Click **Save and close** when done.

### NOTE

You can use IntelliSense when writing code in business service components.

Existing business service component can be managed in the [Code Editor](#) by clicking the **Open in code** button, or in Studio, by clicking **Check Out** to start editing and **Release** to allow other users to edit.

You can import modules from other server side components by referencing the module name, for example "demo.businessLogic.js".

## Example

In this example, we import a module from a business service component into a server script.

1. Create a [digital asset](#) of type Resources and set it to context.
2. Create a business service component, we are going to name it "test". In the Routes section add the following code:

```
export function bsc_log(){  
  var e = 1;  
  var f =2;  
  log('Test 1');  
}
```

3. Save and close the business service component.
4. Create a server script that imports the business service component and uses the Routes section. Make sure you write the name of the routes component correctly.

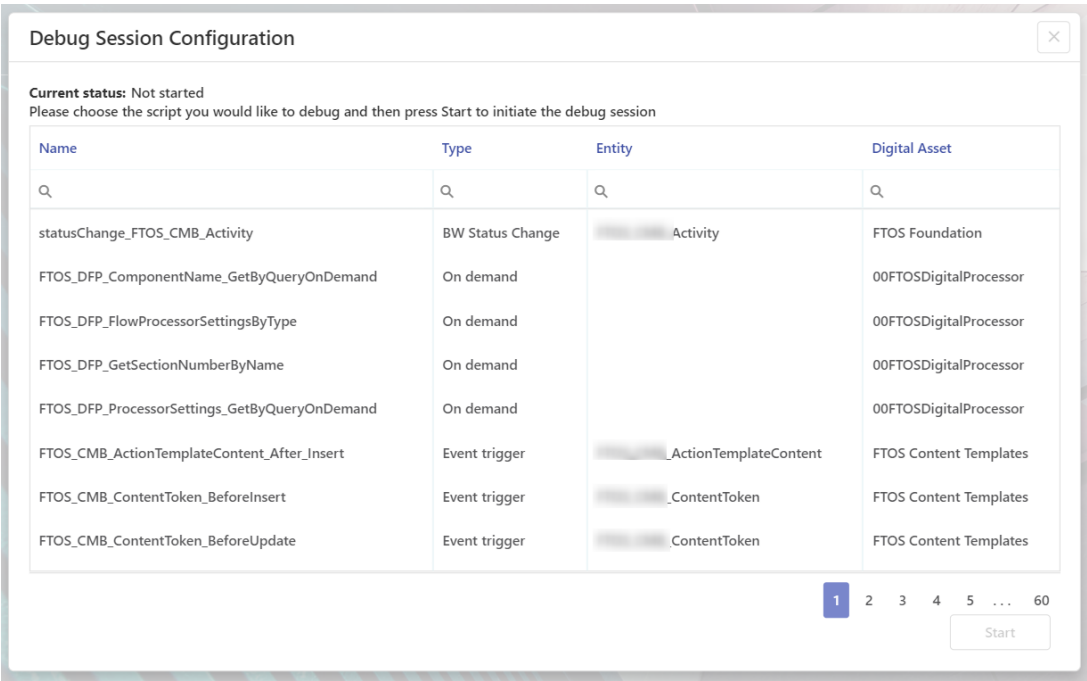
```
log('BD_script_01');  
(async () => {{  
  const routes = await import('test.routes.js');  
  
  routes.bsc_log();  
  setMessage(routes);  
}}})();
```

# Debug Components

To debug the business service components make sure the following prerequisites are met:

- your user has the Debugger role
- sys-workflowengine-enable-debugging system parameter is set to 1
- system parameter sys-workflowengine-type is set to V8

1. Login to the **Portal** on your environment and click the play button. The list of scripts open.



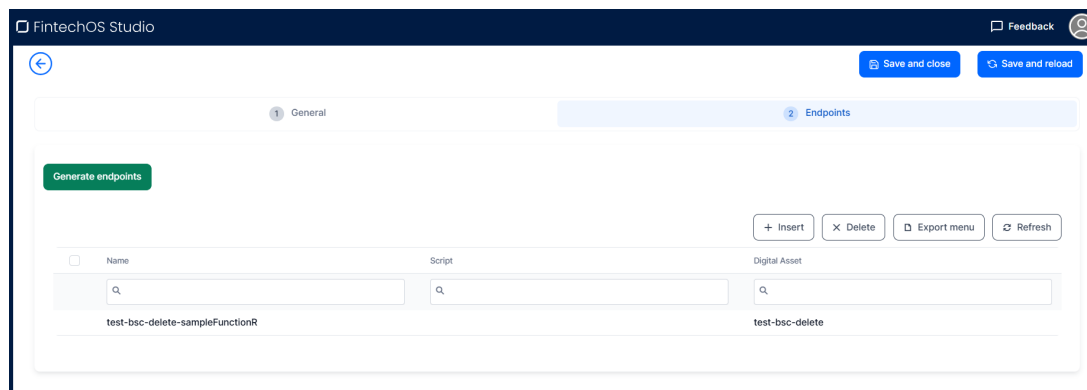
2. Choose the script you want to debug and click **Start** to initiate the debug session.
3. Call out the endpoint of the script from console and click Get debug URL.
4. Paste the URL in a separate page of the browser. You should see his script on the left side of the page.

## Generate Endpoints

You can generate endpoints for business service components that have export functions in the routes component. An endpoint is generated for each export function in the routes component. This is done by accessing the **Endpoints** tab, after opening a business service component. This is available starting with v24.3.2.

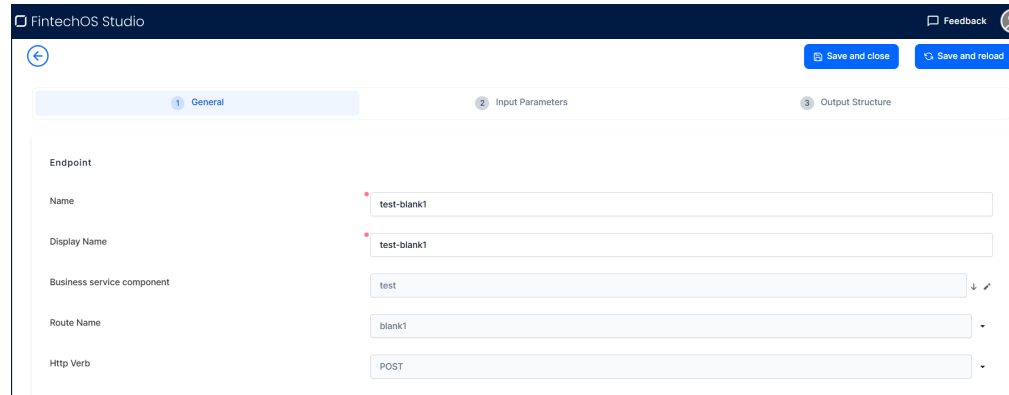
It's important to know that if you generate an endpoint from an export function, and then make changes to that function, you must delete the initial endpoint and generate it again to apply said changes.

1. Open a business service component with an export function in the routes section.
2. Click the **Endpoints** tab and click **Generate Endpoints**. The endpoint is generated and displayed in the list.



3. Double click the endpoint to open a form for editing. Fill in the following fields:
  - **Business service component:** the name of the component which generated the endpoint.
  - **Route name:** the name of the route which generated the endpoint.
  - **Http verb:** select the request method to indicate the purpose of the request and what is expected if the request is successful. Pick from:

delete, get, patch, post, put.



4. Click the **Input Parameters** tab and **Output Structure** to complete editing your endpoint. The steps here are the same as the ones described in the [Endpoints](#) page.

## Import or Export Components

Business service components can be exported from the environment where they were created and imported to another one in the same way that other configuration items are added to [digital assets](#) which are then bundled together in digital solutions and exported as digital solution packages.

## Client Script Libraries

Client script libraries allow you to define collections of custom JavaScript functions that you can use or reuse in your client scripts. Once you bundle functions that you frequently use together in a client script library, you can access those functions in any client script by importing the library.

## Create a Client Script Library

1. In FintechOS Studio, go to **Main Menu > Advanced > Client Script Libraries**.
2. Click **Insert**. The Add Client Script Library page appears.

← Save and close Save and reload Open in code

Edit Client Script Library

Name

Code

Release Check Out

```

1 function customRegisterVirtualAttributesWithOptions(arrOfObjects, entity, prefix) {
2   if (arrOfObjects && arrOfObjects.length) {
3     if (entity && prefix) {
4       arrOfObjects.forEach(function (object) {
5         ebs.rendering.
6           registerVirtualRelatedAttribute(prefix + object.name, entity, obj
7       });
8     }
9   }
10 }
11
12
13 function registerRelatedVirtualAttributes(lookupName, objVirtualAttributes, prefixName
14   if (!prefixName) {
15     prefixName = lookupName;
16   }

```

Definition

Release Check Out

```

1 interface ILibrary{
2   // Add here the definition for the "
3   registerRelatedVirtualAttributes(loo
4   loadRelatedVirtualAttributes(lookupN
5
6   }
7

```

3. In the **Name** field, provide a unique name for the library.
4. In the **Code** field, provide the actual code that will be executed (typically functions definitions). E.g:
 

```
function capitalize(word){
    return word.substr(0,1).toUpperCase() + word.substr(1);
}
```
5. For usability purposes and to avoid calling the wrong methods and attributes, when creating script libraries and scripts you can use code snippets. For information on how to use code snippets, see [Code Snippets Support for JavaScript](#).
6. In the **Definition** field, provide the TypeScript definition of the code you provided in the **Code** field. For more information on how to write TypeScript declarations, see the

TypeScript documentation. E.g:

```
interface ILibrary{  
    capitalize(word: string): string;  
}
```

7. Click **Save and close**. The new function is added to the Client Script Library and it will be displayed in the Client Script Libraries List.

#### HINT

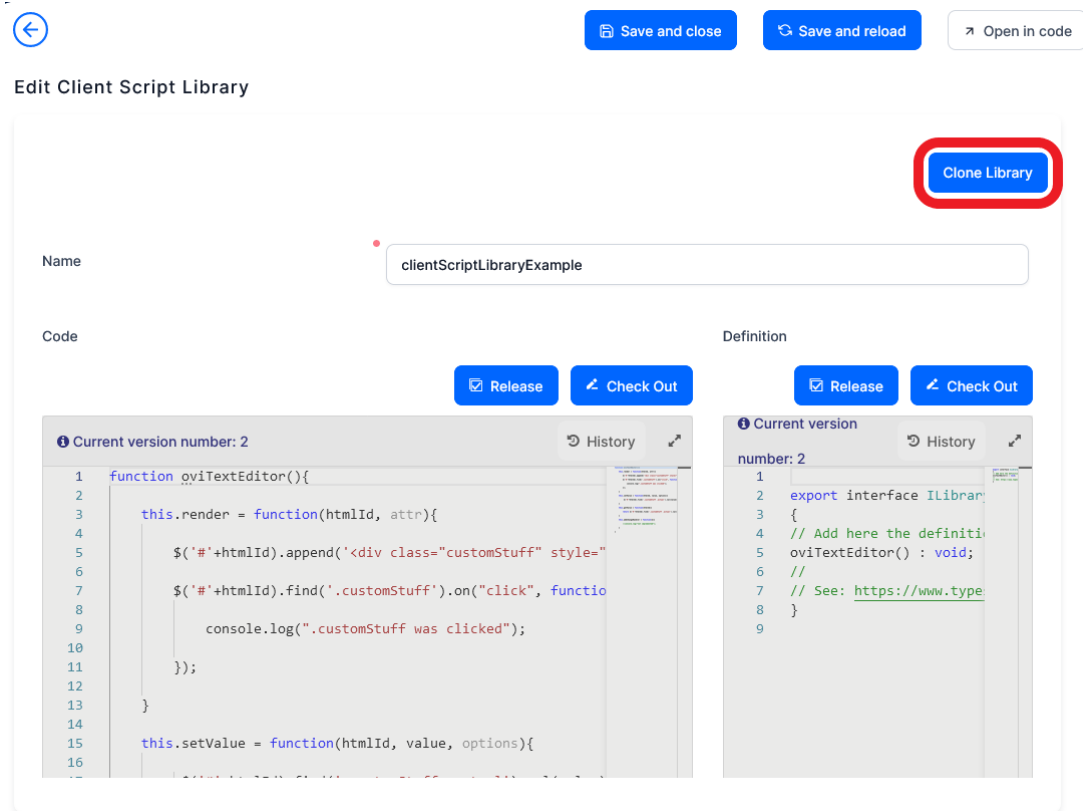
You can import a Client Script Library into another library but make sure that you avoid circular references.

## Clone a Client Script Library

If you wish to create a variation based on an existing client script library, you can clone the desired library and use it as a starting point for further modifications.

To clone a client script library

1. Open the source library in the editor and press the **Clone Library** button.



2. In the pop-up window that opens, provide a name for the clone.
3. Click **Clone** to create your new library.

## Import Client Script Libraries in your Client Side Code

The use of custom functions defined in client script libraries is available in the following entities and JavaScript attributes:

Entity	Attributes
Entity Form	Before Events, After Events
Entity Form Section	After Events, After Section Save
Custom Form	After Generate Events
Entity View	After Generate Js
HtmlWidget	JavaScript

To use a custom function in a JavaScript field, import the client script library by using the `ebs.importClientScript` Client SDK function in the following format:

```
ebs.importClientScript("");
```

Place the cursor between the two quotation marks ("" ) and press **CTRL+SPACE**. You will be suggested the available client script libraries.

Choose the client script library that defines the custom function that you want to use. E.g:

```
var myLib = ebs.importClientScript("myLibrary");
```

For usability purposes and to avoid calling the wrong methods and attributes, when creating script libraries and scripts, you can use the '\$m' mechanism as follows:

- To transform text in an entity\_name string, type `$m.entity_name` and then press the **TAB** key.
- To transform text in an attribute\_name string, type `$m.entity_name.attribute_name` and then press the **TAB** key.
- To transform text in a relationship\_name string, type `$m.entity_name.relationship_name` and then press the **TAB** key.

#### NOTE

All entities and their attributes and relationships are available regardless of the current entity.

Once imported, you can access the library functions as methods of the library object. E.g.:

```
console.log(myLib.capitalize("aWord"));
```

The TypeScript definitions of a client script library describe their functions. Therefore, the IntelliSense assistant will suggest the function name after you type the variable name.

If you're using Chrome, open the client application where the code will run and by using the **Developer tools** > **Console**, check that the custom function returns the expected result.

## Example: Print the word “AWord”.

### Code execution

When retrieved, the code is transformed using the module pattern:

#### INPUT

```
var c1 = 1;

function A()
{
    return c1;
}

var c2 = 2;
function B()
{
    var y = function C()
    {
    };
    return c2;
}
```

#### OUTPUT

```
///  
sourceURL=clientScriptLib_myScript.js;  
  
(function(){  
    var $export = {};  
    var c1 = 1;  
    $export.A = function ()  
    {  
        return c1;  
    };  
    var c2 = 2;  
    $export.B = function ()
```

```
    {  
      var y = function c()  
      {  
      };  
      return c2;  
    };  
    return $export;  
  }());
```

If the module pattern is detected in the INPUT, then no transformation occurs.

## Server Automation Scripts

FintechOS Studio allows you to manage complex and secure automation and validation tasks by triggering the execution of custom scripts on the server side. Some of the main features of server automation scripts are:

- Automation scripts are executed synchronously, which means that the execution happens in a single series. Subsequent operations cannot be performed until the current operations is finished.
- They execute in the current transaction, immediately after they are triggered.
- Automation scripts log errors if logging is enabled.
- Reusable "Code Blocks" on [page 1227](#) can be included in "Server Automation Script Libraries" on [page 1206](#) and associated later to scripts.

In FintechOS Studio, you can create and use two types of automation scripts, as follows:

- "Event Triggered Automation Scripts" on the next page
- "On-Demand Automation Scripts" on the next page

## Event Triggered Automation Scripts

These automation scripts are automatically triggered when a CRUD event (Create, Read, Update, or Delete) occurs in the database.

A list of predefined methods and functions with corresponding code-snippets are available within a dedicated development library (automation script library), covering most common use cases and technical applicability scenarios.

## On-Demand Automation Scripts

While event-triggered automation scripts are context-based and linked to a specific entity within the data model, on-demand automation scripts are context independent and available for being called from any object or context (APIs, user journeys, other automation scripts, etc.).

On-demand automation scripts can be triggered manually if they are attached to an action. For usability purposes, you can organize actions performed on an entity into action groups.

---

## Create Event Triggered Automation Scripts

1. In FintechOS Studio, go to **Main Menu > Advanced > Server Automation Scripts**.
2. Click **Insert**.
3. In the **Name** field, type the script name.
4. In the **Description** field type a description of the script logic. The field is optional, but we recommend you to provide a description, so that developers have a clear view on what the script does.
5. From the **Script Type** drop-down, select **Event triggered**.

6. In the **Event** drop-down, select the event CRUD operation that triggers the script: **Read**, **Update**, **Insert**, or **Delete**.

Event	Description
<b>Read</b>	Triggers when information from the target entity is read, including by other automation scripts or <code>ftos.data.getByQuery</code> functions. You should be careful not to alter other functionality related to the entity.
<b>Update</b>	<p>Triggers when the target entity is updated.</p> <p>If an automation script updates another entity, and that entity also has an automation script on update, both scripts will be triggered. If any automation script in the execution chain throws an exception, the entire transaction is roll-backed. That means that every operation is enlisted to the master Ebs Core transaction.</p> <p>In every automation script code you have a context variable that holds data about the current scope.</p> <div> <p><b>IMPORTANT!</b></p> <p>If you write an update script in which you update the same record, it will trigger a recursion. Although the FintechOS Studio has been build to prevent such situations, the transaction will be rolled back after several iterations. There is also a Prevent Recursivity feature that you can enable for such situations.</p> </div>
<b>Insert</b>	Triggers when a new record is added to the target entity.

Event	Description
<b>Delete</b>	<p>Triggers when a record of the target entity is deleted.</p> <p>Can be used in conjunction with:</p> <ul style="list-style-type: none"> <li>• The 'before' stage, otherwise, the automation script tries deleting a record ID which has already been deleted.</li> <li>• The 'after insert' script. Do not try to use it 'before insert' as you do not have a record ID yet to relate to.</li> </ul>

7. In the **Stage** drop-down, choose the stage of event that triggers the automation script:

Stage	Description
<b>Before</b>	<p>Executes the code before a read, update, insert or delete event occurs.</p> <p><b>Use cases:</b> To validate information for update/read events and restrict access/filter information for read events.</p>
<b>After</b>	<p>Executes the code after a read, update, insert or delete event occurs. Using the record ID, you can create single or cascading events.</p>

8. In the **Entity** drop-down, select the entity triggering the server automation script.
9. In the **Code** field, enter the script code (using [Server SDK functions](#)). If you want to use ["Server Automation Script Libraries"](#) on [page 1206](#), you can do so, by calling functions defined in the library. E.g.:

```
log('Client - BeforeUpdate - START - ' + ftos.context.id);
log('CONTEXT: ' + serialize(context));
ftos.context.setAdminMode(true);
if(ftos.context.ExecutionDepth < 2){
```

```

        var clientName = isNullOrEmpty
(ftos.context.values.AgreementCounterpartyName) ?
ftos.context.beforeValues.AgreementCounterpartyName :
ftos.context.values.AgreementCounterpartyName;
        clientName = clientName.trim();
        ftos.context.values.AgreementCounterpartyName =
clientName;
        ftos.context.values.InitialCounterpartyName = clientName;
        var fenergoClientId = isNullOrEmpty
(ftos.context.values.FenergoClientId) ?
ftos.context.beforeValues.FenergoClientId :
ftos.context.values.FenergoClientId;
        var clientId = ftos.context.id;
        if(!isNullOrEmpty(fenergoClientId)){
            var duplicateClient = getByQuery({
                "entity": {
                    "alias": "a",
                    "name": "Client"
                },
                "attributelist": [
                    {
                        "name": "AgreementCounterpartyName"
                    },
                    {
                        "name": "Clientid"
                    }
                ],
                "where": {
                    "type": "and",
                    "conditionlist": [{
                        "first": "a.FenergoClientId",
                        "type": "equals",
                        "second": "val(" + fenergoClientId + ")"
                    },
                    {
                        "first": "a.Clientid",
                        "type": "notequals",
                        "second": "val(" + clientId + ")"
                    }
                ]
            })
        }
        if(duplicateClient != null && duplicateClient.length != 0){
            throwException(getErrorMessage("61111"));
        }
    }
}

```

```

var duplicateClient = getByQuery({
  "entity": {
    "alias": "a",
    "name": "Client"
  },
  "attributelist": [
    {
      "name": "AgreementCounterpartyName"
    }
  ],
  "where": {
    "type": "and",
    "conditionlist": [{
      "first": "a.AgreementCounterpartyName",
      "type": "equals",
      "second": "val(" + clientId + ")"
    },
    {
      "first": "a.Clientid",
      "type": "notequals",
      "second": "val(" + clientId + ")"
    }
  ]
});
if(duplicateClient !== null && duplicateClient.length !== 0){
  throwException(getErrorMessage("61111"));
}
log('Client - BeforeUpdate - END - ' + ftos.context.id);

```

### HINT

To avoid calling the wrong methods and attributes, you can use the '\$m' mechanism when writing the code. For more information on how to the mechanism, see [Code Snippets Support for JavaScript Text Boxes](#)

If you want to use an automation script library (the one whose functions you

appended in the **Code** field), after saving the automation script, go to the Edit Automation Script page and, in the List of Automation Script Libraries section, click the **Insert existing** button and select the desired script library by double-clicking it.

10. To prevent recursive runs of the scripts, tick the **Prevent Recursivity** checkbox.
11. Specify if the script is active or disabled. Tick the **Disable** checkbox to disable the automation script code execution (select it for debugging purposes or for obsolete automation scripts).
12. Use the **Server Automation Script Libraries** section to select any "[Server Automation Script Libraries](#)" on [page 1206](#) your automation script uses.
13. Click **Save and close** to save the automation script.

## Create On-demand Server Automation Scripts

1. In FintechOS Studio, go to **Main Menu > Advanced > Server Automation Scripts**.
2. Click **Insert**. The **Add Server Automation Script** form is displayed.
3. In the **General** tab, enter a unique **Name** for the automation script.
4. In the **Description** field type a description of the automation script logic. The field is optional, but we recommend you to provide a description, so that developers have a clear view on what the automation script is intended to do.
5. From the **Script Type** drop-down, select **On-Demand**.

1 General 2 Input Parameters 3 Output Structure

Server Automation Script

Name

Description

Script Type

Code

RELEASE CHECK OUT

Current version number: none. CHECKED OUT

```

1 setDataInStore(true);
2 var asyncInstanceFetch = {};
3 asyncInstanceFetch.entity = {};
4 asyncInstanceFetch.entity.name = "FTOS_IntegrationProcessInstance";
5 asyncInstanceFetch.entity.alias = "p";
6 asyncInstanceFetch.entity.attributelist = [];
7 asyncInstanceFetch.entity.attributelist.push({"name": "Name"}, {"name": "FTOS_IntegrationProcessInstanceId"}, {"name": "re
8 asyncInstanceFetch.where = {};
9 asyncInstanceFetch.where.type = "and";
10 asyncInstanceFetch.where.conditionlist = [];
11 asyncInstanceFetch.where.conditionlist.push({"type": "equals", "first": "p.runCount", "second": "val(true)"});

```

History

6. In the **Code** field, enter the automation script code (using [Server SDK functions](#)). If you want to use ["Server Automation Script Libraries"](#) on page 1206, you can do so, by calling functions defined in the library.

```

//parameter called input1 received from client <a
href="https://docs.fintechos.com/ClientSDK/#Other/ebs.callAc
tionByName.htm">callActionByName</a> function
var input1 = context.Data.input1;

//call to a method defined in server script library
var cnt = new FTOSExample().getCount();

//the custom returned object
var acc = {totalCount: cnt, test:"example1", resInput1:
input1};

//returns data in UI (on callback of the callActionByName))
setData(acc);

```

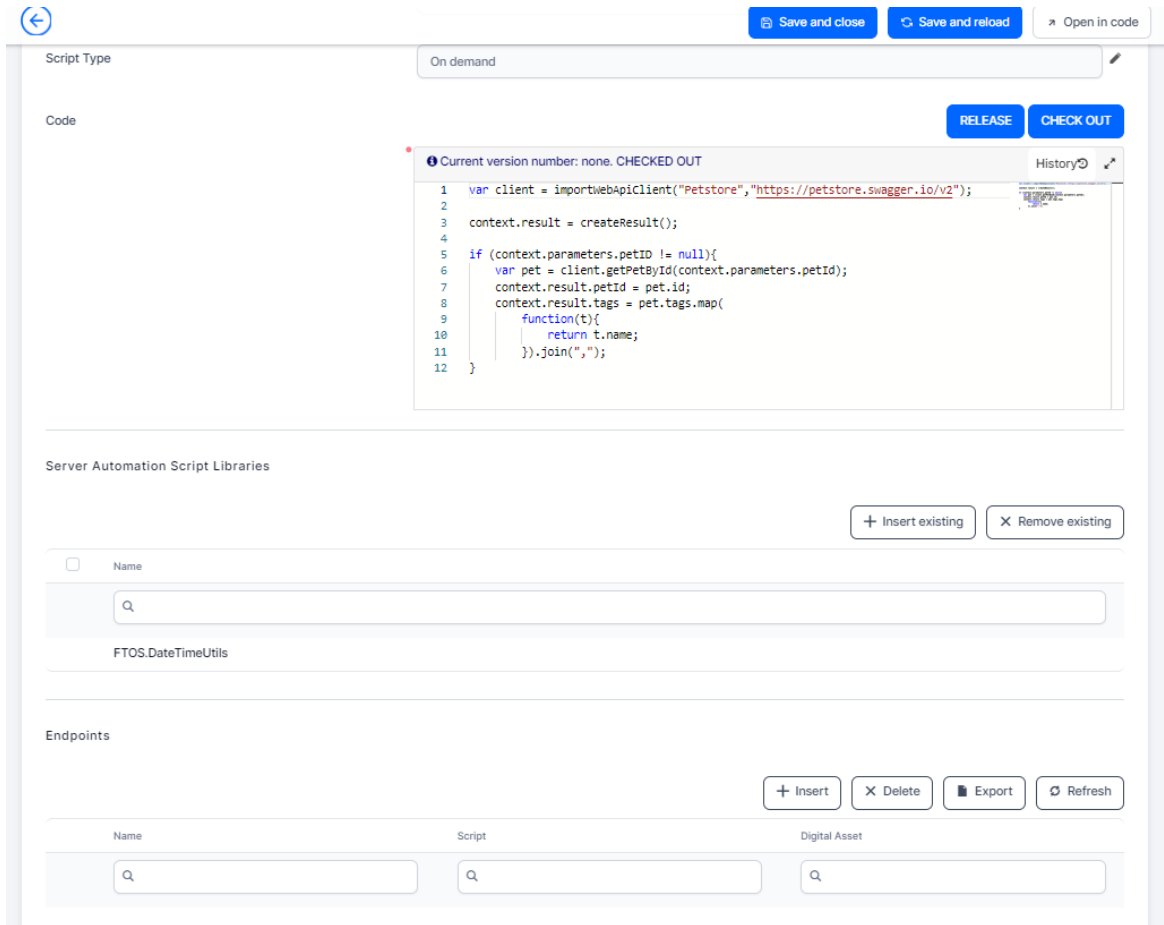
To avoid calling the wrong methods and attributes, you can use the '\$m' mechanism when writing the code. For more information, see [Code Snippets Support for JavaScript Text Boxes](#).

## NOTE

If you want to use an automation script library (the one whose functions

you appended in the **Code** field), after saving the automation script, go to the Edit Automation Script page and, in the List of Automation Script Libraries section, click the **Insert existing** button and select the desired script library by double-clicking it.

7. Click **Save and reload**. This will enable the Server Automation Script Libraries and the Endpoints sections at the bottom of the page.
8. Use the **Server Automation Script Libraries** section to select any "[Server Automation Script Libraries](#)" on page 1206 your automation script uses.
9. Use the **Endpoints** section to define any "[Endpoints](#)" on page 1213 that call the automation script.
10. Click **Save and reload**.



## Call Actions

**Prerequisite:** To append a callAction to execute a server automation script on demand, you need to have the automation script that you want to execute and an endpoint for that script (see "Endpoints" on page 1213 for details).

On the form where you want the automation script to be executed on demand, in the JavaScript fields (**Advanced** tab), append the `ebs.callActionByName` method. E.g.:

```

ebs.callActionByName("FTOS.Example1", { input1: "test1" }, function
(e) {
    console.log("Response", e);
    if (e.IsSuccess) {
        //response received from server script set with method
        setData
        console.log("ResponseFromScript ", e.UIResult.Data);
    }
}

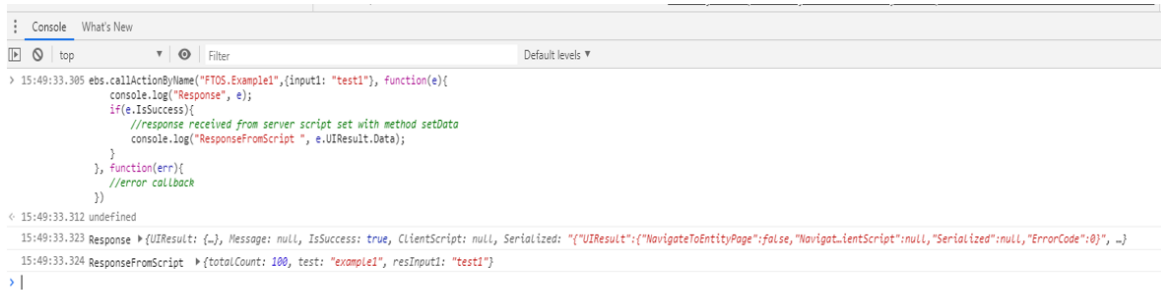
```

```

    }
  }, function (err) {
    //error callback
  });

```

The action button will be displayed in the user interface on the customer journey / data form. When the user clicks the button, the automation script associated with it will be executed in the browser's console:



```

> 15:49:33.305 ebs.callActionByName("FTOS.Example1",{input1: "test1"}, function(e){
  console.log("Response", e);
  if(e.isSuccess){
    //response received from server script set with method setData
    console.log("ResponseFromScript ", e.UIResult.Data);
  }
  }, function(err){
    //error callback
  })
< 15:49:33.312 undefined
15:49:33.323 Response {UIResult: {Message: null, IsSuccess: true, ClientScript: null, Serialized: '{"UIResult":{"NavigateToEntityPage":false,"NavigateToEntityPage":false,"NavigateToEntityPage":false,"Serialized":null,"ErrorCode":0}, ...}'}, Message: null, IsSuccess: true, ClientScript: null, Serialized: '{"UIResult":{"NavigateToEntityPage":false,"NavigateToEntityPage":false,"NavigateToEntityPage":false,"Serialized":null,"ErrorCode":0}, ...}'}
15:49:33.324 ResponseFromScript {totalCount: 100, test: 'example1', resInput: 'test1'}
> |

```

## NOTE

callAction connects an automation script with the front end (UI); therefore, you cannot append an action from another action.

If you have some functionalities (e.g. functions1) in an automation script and other functionalities (e.g. functions2) in another automation script, you have two ways to execute them on demand:

- On forms / user journeys, in aftergenerateJs fields, append both actions.

```

Ebs.callActionByName("action1",{param1:"asd"} callback(e){
  Ebs.callActionByName("action2",{param2:"aaa"} callback
(e){
  });
});

```

where:

- action1 is the name of the endpoint to the automation script which contains functions1
- action2 is the endpoint to the automation script which contains functions2

- Group functions 1 and functions 2 using "[Server Automation Script Libraries](#)" on [page 1206](#). Both automation scripts can reference the same library while in the **Code** field of the scripts you can append only the functions specific for that script,

## XML Support

XML support is available in server automation scripts and libraries, allowing you to create and parse XML.

This is helpful if you want to use data formatted as XML. For example, you might want to use product or other related information you have already stored in XML format, instead of creating new entities and attributes.

### Load XML from a String

#### Method

```
server.Xml.Load(string xml)
```

Loads the XML from the specified string.

#### Parameter

```
xml string
```

String containing the XML to load. The string is an XML formatted text.

```
var xml = '<Order></Order>';  
var doc = server.Xml.Load(xml);
```

### Catch an XML Load Error

When loading XML from a string, errors that might occur on XML load from a string are not automatically logged. To log any errors that might occur on XML schema load, use `catch(err)`.

```
var xml = '<Order />';
try
{
    var doc = server.Xml.Load(xml);
}
catch(err)
{
    log(err);
}
```

## Run XPath Queries

You can perform XPath queries to navigate through nodes (elements, attributes) in an XML document.

### Method

```
Query( xpath : string ) : Element[]
```

### E.g.:

```
var xml = '<?xml version="1.0" encoding="utf-8" ?>
<Orders>
  <Order id="01">
    <Product id="1" name="Nexus 5">
      <Price>400.00</Price>
      <Qty>1</Qty>
    </Product>
    <Product id="5" name="Wireless Charger">
      <Price>50.00</Price>
      <Qty>1</Qty>
    </Product>
  </Order>
  <Order id="02">
    <Product id="2" name="IPhone">
      <Price>800.00</Price>
      <Qty>1</Qty>
    </Product>
    <Product id="5" name="Wireless Charger">
      <Price>50.00</Price>
```

```

        <Qty>1</Qty>
    </Product>
</Order>
</Orders>';
var products = doc.Query("/Orders/Order/Product[@name='Wireless
Charger']");
log(products[0]['id']);
log(products[1]['id']);

```

Where:

- **<Orders>** is the root element
- **Order** is an element node
- **<Price>** and **<Qty>** are child elements of **<Product>**
- **Wireless Charger** is an attribute node

## Run XPath Queries with Namespaces

You can perform XPath queries with namespaces to navigate through nodes (elements, attributes) in an XML document.

### Method

```

Query( xpath : string, namespaces : { key : string, value : string
)

```

E.g.:

```

var xml = '<?xml version="1.0" encoding="utf-8" ?>
<x:Orders xmlns:x="http://myuri">
  <x:Order x:id="01">
    <Product id="1" name="Nexus 5">
      <Price>400.00</Price>
      <Qty>1</Qty>
    </Product>
    <Product id="5" name="Wireless Charger">
      <Price>50.00</Price>
      <Qty>1</Qty>
    </Product>
  </x:Order>
</x:Orders>';

```

```

    </x:Order>
    <x:Order x:id="02">
      <Product id="2" name="iPhone">
        <Price>800.00</Price>
        <Qty>1</Qty>
      </Product>
      <Product id="5" name="Wireless Charger">
        <Price>50.00</Price>
        <Qty>1</Qty>
      </Product>
    </x:Order>
  </x:Orders>';
  var doc = server.Xml.Load(xml);
  var products = doc.Query("/x:Orders/x:Order/Product[@name='Wireless Charger']", { 'x' : 'http://myuri' });
  log(products[0]['id']);
  log(products[1]['id'])

```

## Node API Calls

The table below lists the properties you can access on the nodes within an XML document.

Property	Returns	Description
HasAttributes	Boolean	Indicates if an element has at least one attribute. <b>Property Value:</b> true if the current node has attributes; otherwise, false.
HasElements	Boolean	Indicates if an element has at least one child element. <b>Property Value:</b> true if the current node has child elements; otherwise, false.
AttributeCount	Number	Returns the number of attributes on the current node (element). <b>Property Value:</b> The number of attributes if the current node (element) has attributes; otherwise, null.

Property	Returns	Description
ElementCount	Number	Returns the number of elements on the current node. <b>Property Value:</b> The number of elements if the current node (element) has child elements; otherwise, null.
Elements() : Element[]	Array of strings	Returns all child elements of a node element.
Elements(name : string) To specify a namespace, use following syntax for name <code>{http://myuri.org}name</code>	Array of strings	Returns all child elements with the specified node element.
this[attributeName : string ] To specify a namespace for the attribute, use following syntax for name <code>{http://myuri.org}name</code>	String	Returns the value of the specified node attribute.

## Fluent Queries

Fluent queries allow you to run database queries in your server automation scripts using an SQL-like fluent interface. Intelligent code completion is available in the code editor both for the query inputs and for the result sets.

### IMPORTANT!

- Fluent queries can only be used in administrative context. See the [Server SDK Reference Guide](#) for information on how to temporarily change the transaction context.
- Result sets include only the columns specified in the select statements. Lookup fields are not automatically expanded.
- Data ownership is per organization.
- Date and datetime values are not returned as strings. The wrapper for datetime values (JsDateTime) stores the date as UTC and provides utility methods for data manipulation.

## Execute a Fluent Query

To execute a fluent query in a server automation script:

1. Use the `ftos.data.query.getAlias` method to define an entity alias.
2. Use the `ftos.data.query.from` method to run the query on the desired entity.
3. Use the `.selectColumns` method to select the returned attributes.
4. Use the `.distinct()` method to return only distinct values in the result set (no duplicates).
5. Use the `.execute()` method to run the query.

```
var A = ftos.data.query.getAlias('Account');

var myFluentQuery = ftos.data.query.from('Account', A)
    .selectColumns (A.Name, A.Email)
    .distinct()
    .execute();

log(myFluentQuery)
```

The code above will log an output similar to the following:

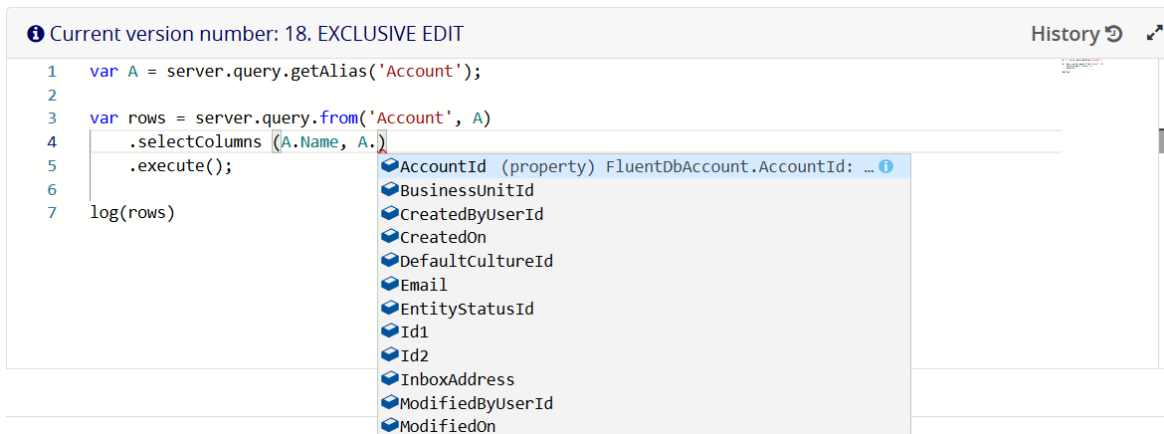
```
-[START]-----
Timestamp: 7/1/2020 5:01:51 PM
Message: INFO [CID=200524bb-8d28-4079-80c3-a003a9ecad7d] [
  {
    "values": {
      "A_Name": "Jane Doe",
      "A_Email": "janedoe@fintechos.com"
    }
  },
  {
    "values": {
      "A_Name": "Andrew Jones",
      "A_Email": "adrew.jones@fintechos.com"
    }
  },
  {
    "values": {
      "A_Name": "John Doe",
```

```

        "A_Email": "john.doe@fintechos.com"
    }
}
Severity: Information
-[END]-----

```

The entity alias definition enables intelligent code completion for the corresponding entity attributes:



## Comparison Operators

The following comparison operators are supported in fluent queries: `equals` (eq), `notEquals` (neq), `greaterThan` (gt), `greaterThanOrEquals` (gte), `lessThan` (lt), and `lessThanOrEquals` (lte)

## Logical Operators

The following logical operators are supported in fluent queries: `and`, `or`, `andNot`, and `orNot`.

## Inner Joins

Use the `.innerJoin` and `.on` methods to define inner join clauses for the queried entities.

```

var O = ftos.data.query.getAlias('optionset');
var A = ftos.data.query.getAlias('attribute');

var myFluentQuery = ftos.data.query.from('attribute', A)

```

```
.innerJoin('optionset', 0)
    .on(A.OptionSetId.eq(O.OptionSetId))
.top(5)
.orderBy(A.Name)
.execute()
```

### Left Joins

Use the `.leftJoin` and `.on` methods to define left outer join clauses for the queried entities.

```
var O = ftos.data.query.getAlias('optionset');
var A = ftos.data.query.getAlias('attribute');

var myFluentQuery = ftos.data.query.from('attribute', A)
    .leftJoin('optionset', O)
        .on(A.OptionSetId.eq(O.OptionSetId))
    .top(5)
    .selectColumns(A.Name, O.DisplayName)
    .orderBy(A.Name)
    .execute();
```

### Attribute Aliases (Projections)

To define aliases for the queried attributes, use the `.selectProjection` method instead of `.selectColumns` and the `.executeAndMap` method instead of `.execute`.

Projections are useful to customize the result set field names when different entities have similar attribute names or when attribute names are not expressive (see ["Work with Fluent Query Result Sets" on page 1184](#) for more details about fluent query outputs).

## Example: Return customized field names in a fluent query result set

In this example, we return information about 5 random entity-attribute pairs.

```
var E = ftos.data.query.getAlias('entity');
var A = ftos.data.query.getAlias('attribute');

var P; //projection alias
```

```

var rows = ftos.data.query.from('entity', E)
    .innerJoin('attribute', A)
    .on(E.EntityId.eq(A.EntityId))
    .top(5)
    .selectProjection( P =
    {
        EntityName : E.Name,
        AttributeName : A.Name,
        Type : A.AttributeType
    })
    .executeAndMap(P);

log(rows)

```

The result set for the above fluent query will include the customized **EntityName**, **AttributeName**, and **Type** field names.

```

-[START]-----
Timestamp: 7/2/2020 2:00:55 PM
Message: INFO [CID=200524bb-8d28-4079-80c3-a003a9ecad7d] [
    {
        "EntityName": "Account",
        "AttributeName": "Phone",
        "Type": "0a39db15-7634-4af3-8bd8-004fcf27e8a6"
    },
    {
        "EntityName": "Account",
        "AttributeName": "Accountid",
        "Type": "ddce8347-794d-4a8d-b9d0-42437f653ae4"
    },
    {
        "EntityName": "Account",
        "AttributeName": "Email",
        "Type": "0a39db15-7634-4af3-8bd8-004fcf27e8a6"
    },
    {
        "EntityName": "Account",
        "AttributeName": "modifiedOn",
        "Type": "2e33740e-5026-43b5-919c-cc2d422c280f"
    },
    {
        "EntityName": "Account",
        "AttributeName": "Id2",

```

```

        "Type": "0a39db15-7634-4af3-8bd8-004fcf27e8a6"
    }
]
Severity: Information
-[END]-----

```

### Where Clauses

Where clauses are implemented using the `.where`, `.wherenot`, `.andWhere`, `.orWhere`, `.andWhereNot`, and `.orWhereNot` methods.

### Example: Fluent query with multiple where clauses

```

var F = ftos.data.query.getAlias('FinChartItemValue');

var rows = ftos.data.query.from('FinChartItemValue', F)
    .top(5)
    .where(F.Percent.gte(100).and(F.Percent.lte(200)))
    .orWhere(F.Percent.gte(300).and(F.Percent.lte(400)))
    .execute();

```

### Example: Fluent query with comparison operators

```

var F = ftos.data.query.getAlias('FinChartItemValue');

var rows = ftos.data.query.from('FinChartItemValue', F)
    .top(5)
    .where(F.Percent.isNull)
    .execute();

//equivalent query

var rows = ftos.data.query.from('FinChartItemValue', F)
    .top(5)
    .where(F.Percent.equals(null))
    .execute();

```

## Aggregate Functions

To define aggregate functions on a set of values from the result set, use the `.getCountAlias`, `.getSumAlias`, `.getMaxAlias`, and `.getMinAlias` methods of the `ftos.data.query` property.

## Example: Get attribute aggregates for 5 random entities

In this example, we get the number of attributes, maximum attribute length, minimum attribute length, and total length of all attributes for 5 random entities.

```
var E = ftos.data.query.getAlias('entity');
var A = ftos.data.query.getAlias('attribute');

var Count = ftos.data.query.getCountAlias();
var SumLength = ftos.data.query.getSumAlias(A.Length);
var MaxLength = ftos.data.query.getMaxAlias(A.Length);
var MinLength = ftos.data.query.getMinAlias(A.Length);

var P; //projection alias

var rows = ftos.data.query.from('entity', E)
    .innerJoin('attribute', A)
    .on(E.EntityId.eq(A.EntityId))
    .top(5)
    .where(E.Name.startsWith('a'))
    .selectProjection( P =
    {
        'Entity Name' : E.Name,
        'Number of attributes' : Count,
        'Total attributes\' lengths': SumLength,
        'Maximum attribute length' : MaxLength,
        'Minimum attribute length' : MinLength
    })
    .executeAndMap(P);

log(rows);
```

The code above will log an output similar to the following:

```
-[START]-----
```

```

Timestamp: 7/2/2020 11:21:28 AM
Message: INFO [CID=200524bb-8d28-4079-80c3-a003a9ecad7d] [
{
  "Entity Name": "Account",
  "Number of attributes": 15,
  "Total attributes legth": 1638,
  "Maximum attribute length": 500,
  "Minimum attribute length": 64
},
{
  "Entity Name": "action",
  "Number of attributes": 9,
  "Total attributes legth": 3200,
  "Maximum attribute length": 3000,
  "Minimum attribute length": 0
},
{
  "Entity Name": "actiongroup",
  "Number of attributes": 6,
  "Total attributes legth": 200,
  "Maximum attribute length": 200,
  "Minimum attribute length": 0
},
{
  "Entity Name": "ActionXSecurityRole",
  "Number of attributes": 3,
  "Total attributes legth": null,
  "Maximum attribute length": null,
  "Minimum attribute length": null
},
{
  "Entity Name": "applicationLanguage",
  "Number of attributes": 11,
  "Total attributes legth": 213,
  "Maximum attribute length": 100,
  "Minimum attribute length": 1
}
]
Severity: Information
-[END]-----

```

## Work with Fluent Query Result Sets

Use the `.field()` method to get field values from a row in the result set.

```

var E = ftos.data.query.getAlias('entity');

var rows = ftos.data.query.from('entity', E)
    .top(5)
    .execute();

let output = '\n\nThe following entities were found:\n';

rows.map(function(r)
{
    output += ( ' ' + r.field(E.Name) + '\n' )
});

log(output);

```

The code above will log an output similar to the following:

```

-[START]-----
Timestamp: 7/2/2020 1:15:34 PM
Message: INFO [CID=200524bb-8d28-4079-80c3-a003a9ecad7d]

The following entities were found:
entityBWTransitionActionGroup
SystemUserSecurityRole
BW
optionset
lookupCorrelationAttribute

Severity: Information
-[END]-----

```

### Map Result Sets to POCO Objects

Use the `.executeAndMap()` method to map a fluent query result set to an entity alias. This creates a plain old CLR object (POCO) which allows you to access field values as object properties, instead of using the `.field()` method.

```

var E = ftos.data.query.getAlias('entity');

var rows = ftos.data.query.from('entity', E)
    .top(5)
    .executeAndMap(E);

let output = '\n\nThe following entities were found:\n';

```

```
rows.map(function(r)
{
    output += ( ' ' + r.Name) + '\n'
});

log(output);
```

To map a fluent query result set to multiple entity aliases, use the `.executeAndMapComplex()` method.

```
var E = ftos.data.query.getAlias('entity');
var A = ftos.data.query.getAlias('attribute');

var rows = ftos.data.query.from('entity', E)
    .innerJoin('attribute', A)
    .on(E.EntityId.eq(A.EntityId))
    .top(20)
    .selectColumns(
        E.Name,
        A.Name)
    .executeAndMapComplex({ entity : E, attribute : A});

rows.map(function(r)
{
    var entityName = r.entity.Name;
    var attrName    = r.attribute.Name;

    // do

});
```

## Fluent Query Caching

To improve query performance, you can cache the results of specific queries on their first run. On subsequent runs, the query results will be retrieved directly from the memory cache instead of accessing the database again.

### NOTE

Cached results are stored in the memory cache for 30 minutes.

To set or retrieve a cached result, you must assign a unique **cache key** to your query. Then, when you execute the query:

- If the cache key is not found in the memory cache (on the first execution), the database is queried and the result set is saved in the memory cache with the corresponding cache key.
- If the cache key is found in the memory cache (on subsequent executions within 30 minutes from the first execution), the result set is retrieved directly from the memory cache.

### Assign a Cache Key to a Fluent Query

You can assign a cache key to a query with either the `.cacheAs` or the `.cacheScopedAs` methods:

- `.cacheAs` will cache the result set at the global scope.
- `.cacheScopedAs` will cache the result set at the server automation script's scope.

For example:

```
var acc = ftos.data.query.getAlias('Account');
var phoneId = '07111111';

var fq = ftos.data.query.from(acc)
    .where(acc.Phone.eq(phoneId))
    .cacheAs(phoneId); //Cache the result of the fq query with the
                        phoneID cache key.

//To cache the result at the server automation script's scope
use:
    .cacheScopedAs(phoneId);

//Will be retrieved from the database.
var result1 = fq.executeAndMap(acc);

//Will be retrieved from the memory cache.
var result2 = fq.executeAndMap(acc);
```

### IMPORTANT!

Once a query is cached (on the first run), all subsequent runs will retrieve the cached result set, until the cache expires (in 30 minutes). You can refresh the cache by

providing a different cache key as input to the `.cacheAs` or `.cacheScopedAs` method.

**IMPORTANT!**

`.cacheAs` and `.cacheScopedAs` set the same cache key on the query and will overwrite each other if you use both of them.

**HINT**

The global scope of the `.cacheAs` method allows a server automation script to retrieve result sets cached by previous runs of the same script.

## Create Asynchronous Processing Flows

The FintechOS Platform provides an [asynchronous processing engine](#) that allows you to address scenarios that require time-consuming logic or a high volume of requests through asynchronous processing. Some use cases include:

- Large scale data processing
- Time consuming operations
- Rate limiting and throttling
- Resilient processing with redelivery and Dead Letter Queues (DLQ)
- Scalable workload distribution
- Service decoupling

The Async Engine uses message queues that collect incoming inputs for each step (on-demand automation script) in a processing flow. Each step can operate independently as it consumes inputs from its queue, allowing for parallel processing and efficient resource utilization. Additionally, the message queues provide a buffer that can handle bursts of incoming requests, ensuring smooth processing and improved resilience in case of spikes in load.

## 1 Create the Asynchronous Steps

1. "Create On-demand Server Automation Scripts" on page 1168 for each step in your asynchronous flow.
2. "Create On-demand Server Automation Scripts" on page 1168 of each script so that they match the following structure:

```
{
  "correlationId": "a66e0eef-fa3d-421c-83f6-e7e7c8caf4a5",
  "payload": {
    "your-data": "here"
  }
  "additional-properties: optional"
  "...."
}
```

When an asynchronous flow is called, the Async Engine receives the **payload**, generates a **correlation-id** for tracing, and passes them to the first step of the flow.

3. "Create On-demand Server Automation Scripts" on page 1168 for each step in the flow except the last one to also match the above input parameters (as their outputs become inputs for the next step in the flow).

### IMPORTANT!

Make sure to propagate the **correlation-id** in the output, otherwise the Async Engine will lose track of the requests.

## 2 Configure the Message Queues and Flow in the Configuration Manager

See the Administration Guide for instructions on how to [configure message queues](#) for each step and how to [set up the flow](#). Follow the instructions to configure queues' sizes and polling, flow name, sequence of steps, processing threads, etc.

## Call an Asynchronous Flow via API

The endpoint for calling an asynchronous flow is available at the following address:

```
<platform_address>/async/api/start?flowName=<flow_name>
```

The payload (which will be propagated by the Async Engine to the first step of the flow) must be provided in JSON format:

```
{  
  "your_data": "here"  
}
```

### IMPORTANT!

The user account making the call must have the corresponding user role set up in the [Async Engine's rbac.config](#) property.

A Swagger UI interface with all the asynchronous flow endpoints is also available at:

```
<platform_address>/async/specifications/swagger-ui/index.html
```

## Call an Asynchronous Flow via Service Pipes (Deprecated)

To set an asynchronous flow as the destination endpoint of a Service Pipes route, add the following properties to the route:

```
.setProperty("flowName", constant("your-flow-name"))  
.to("direct:callAsyncEngine");
```

For more information about working with Service Pipes, see ["Call External Services via Service Pipes"](#) on page 967 and the [Service Pipes](#) tutorial.

## Debugging

FintechOS Studio offers several options for debugging automation scripts from the development and testing environments.

## Debugging Server

On development environments, a dedicated JavaScript debugging engine can load your server automation scripts and expose them client-side in a FintechOS Portal instance. This allows you to take advantage of your browser's developer tools to debug your server automation scripts just like you would with client scripts. For instance, you can:

- execute your code line-by-line
- pause the code execution with breakpoints
- monitor variable values or custom expressions with watch expressions

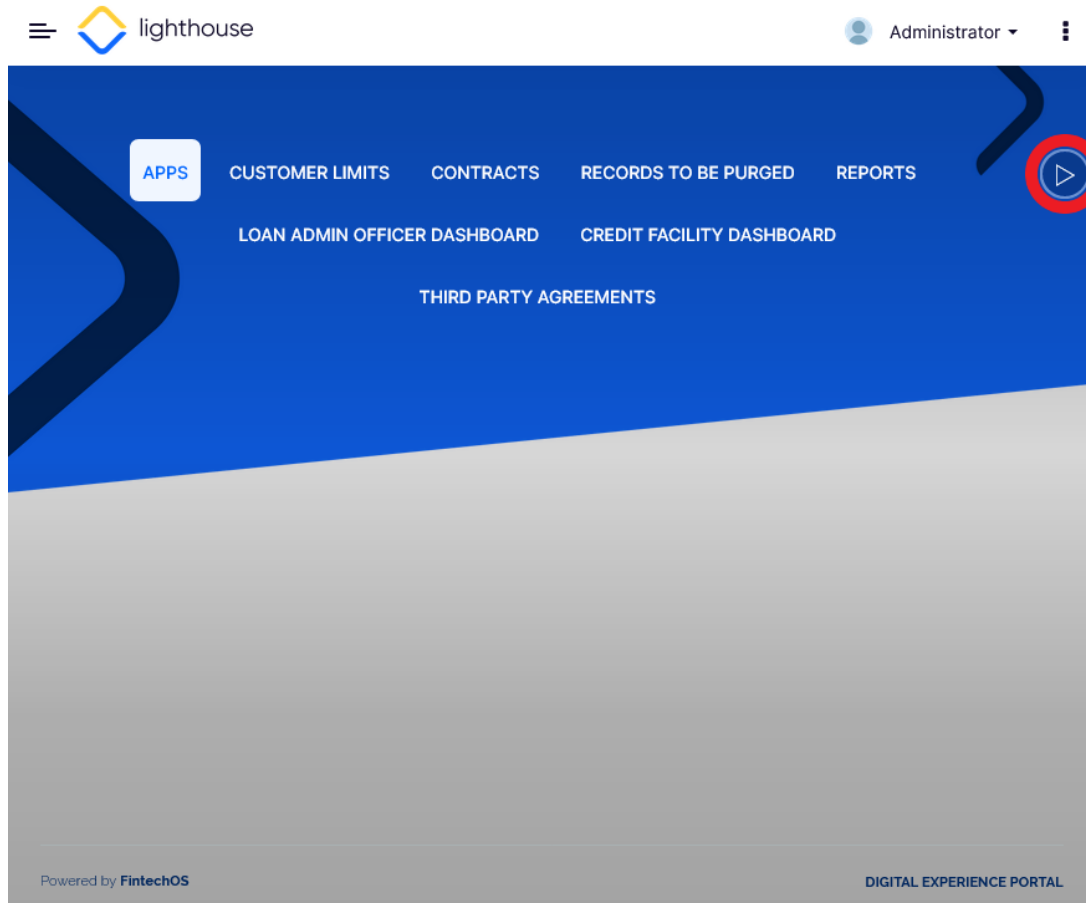
### Debugging Server Prerequisites

- user account must have the **Debugger Users** security role (see ["Security Roles" on page 1245](#)).
- the `sys-workflowengine-enable-debugging` ["System Parameters" on page 1285](#) must be set to 1 to start the debugging server.
- the `sys-workflowengine-type` ["System Parameters" on page 1285](#) must be set to v8, instead of JintV3. V8 is a JavaScript engine that compiles JavaScript directly into machine code for fast execution in web browsers and server environments like Node.js.

### Use the Debugging Server

1. Log in to a FintechOS Portal instance with a developer user account (see ["Debugging Server Prerequisites" above](#)).

2. A debug button is displayed on the portal's dashboard. Click the **Debug** button.



Alternatively, you can use the **Ctrl+Alt+D** or **Command+Alt+D** shortcut on Windows or MacOS operating systems respectively.

3. A list containing all server automation and all scripts executed in the [status transition](#) in a [business workflow](#). Select the script(s) you wish to debug and click **Start** to initiate

the debug session.

Debug Session Configuration

Current status: Not started

Please choose the script/s you would like to debug and then press Start to initiate the debug session

<input type="checkbox"/>	Name	Type	Entity	Digital Asset
	Q	Q	Q	Q
<input type="checkbox"/>	Account_BeforeInsert	Event trigger	Account	FTOS Foundation
<input checked="" type="checkbox"/>	Account_BeforeUpdate	Event trigger	Account	FTOS Foundation
<input type="checkbox"/>	AccountRelOwnership_BeforeDelete	Event trigger	AccountRelOwnership	FTOS Foundation
<input type="checkbox"/>	AccountRelOwnership_BeforeInsert	Event trigger	AccountRelOwnership	FTOS Foundation
<input type="checkbox"/>	AccountRelOwnership_BeforeUpdate	Event trigger	AccountRelOwnership	FTOS Foundation
<input type="checkbox"/>	app001serverAutomationScript	On demand		App 001
<input type="checkbox"/>	armandTest	On demand		PolicyAdminResources
<input type="checkbox"/>	B2C_CloseSession	On demand		Accelerators Prerequisites

1

2

3

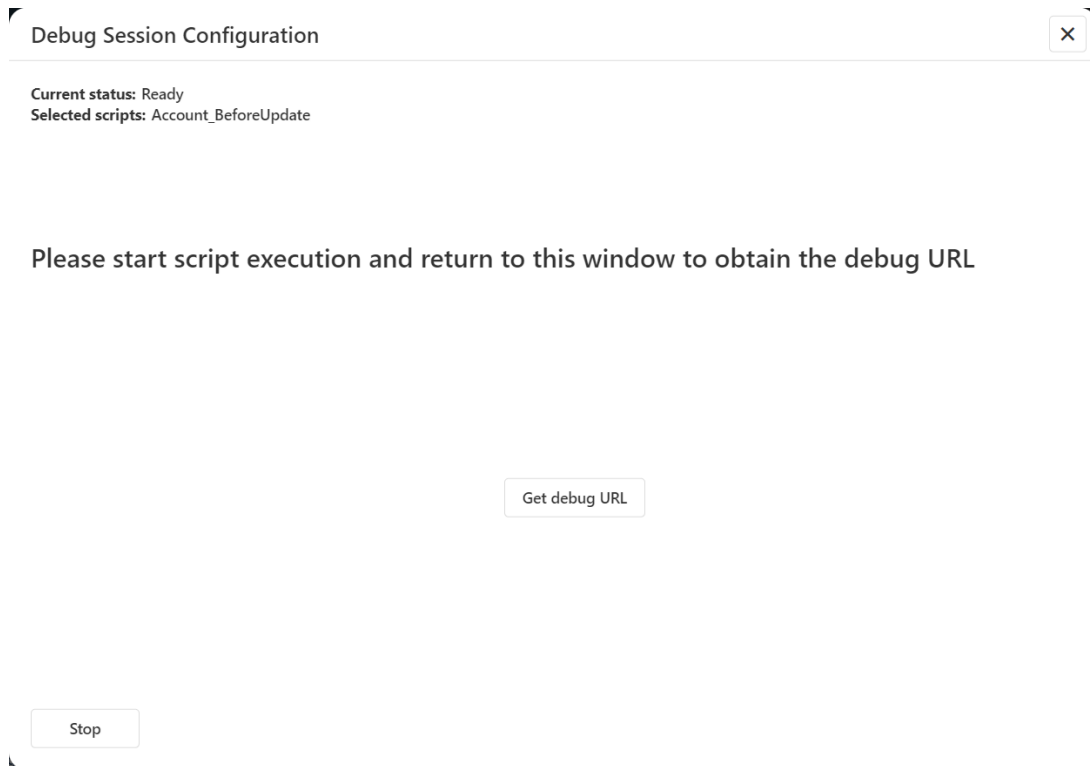
4

5

...

Start

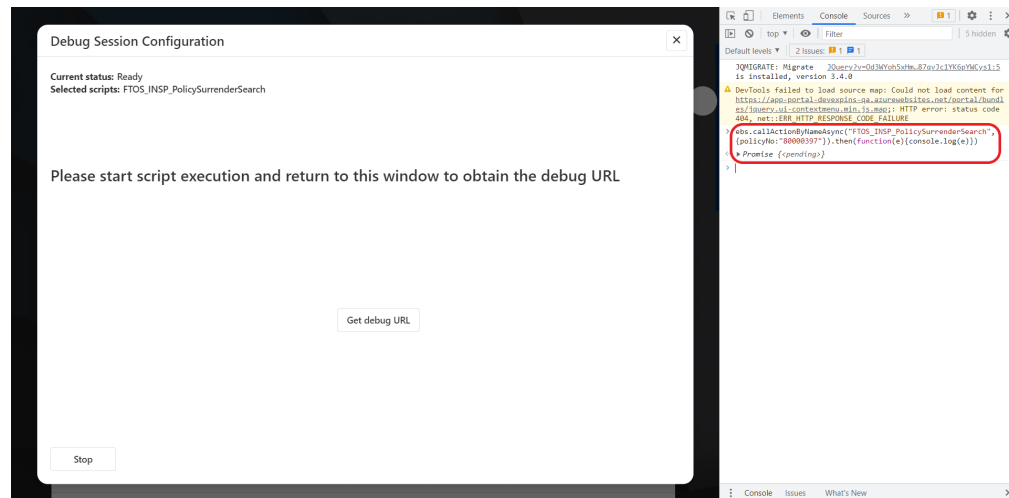
4. The **Debug Session Configuration** engine is now waiting for you to trigger the script execution.



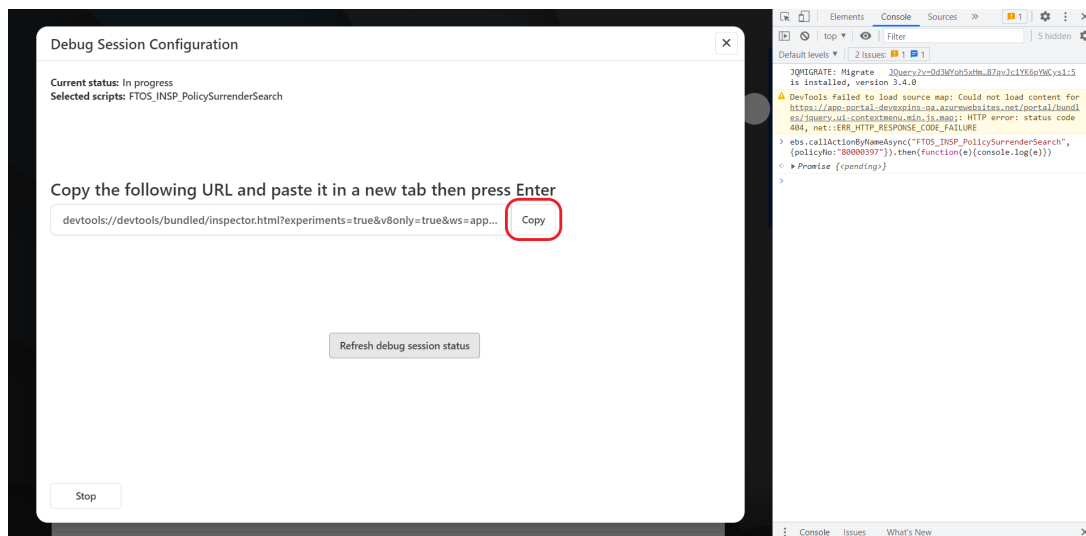
5. Trigger the script:

- For "[Event Triggered Automation Scripts](#)" on [page 1163](#), trigger the corresponding CRUD event in the database.
- For "[On-Demand Automation Scripts](#)" on [page 1163](#) and status transition custom scripts, trigger them via an [API call](#), or from a client process (for instance, from a form driven flow or directly through the browser console) via the

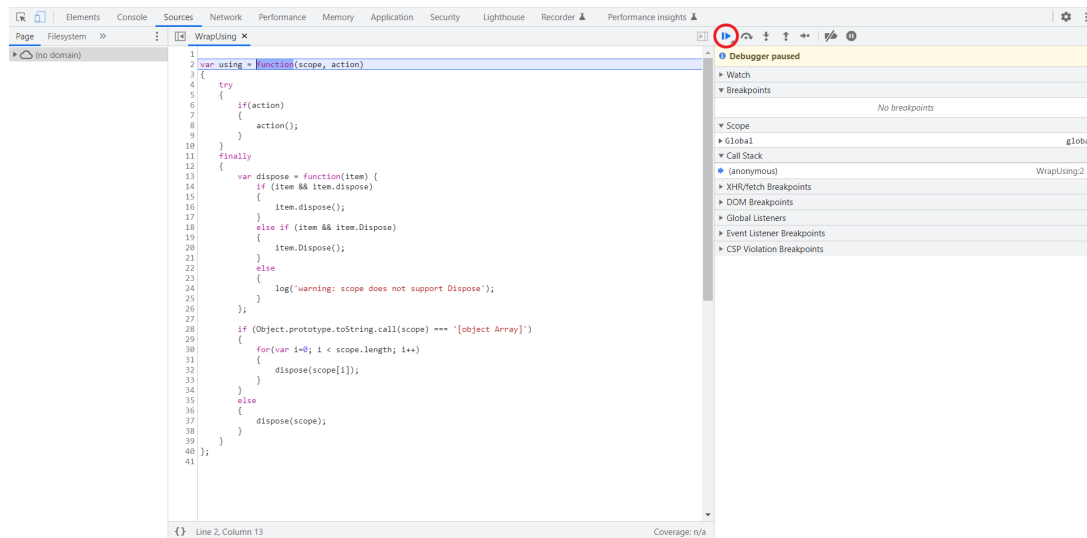
## ebs.callActionByNameAsync function.



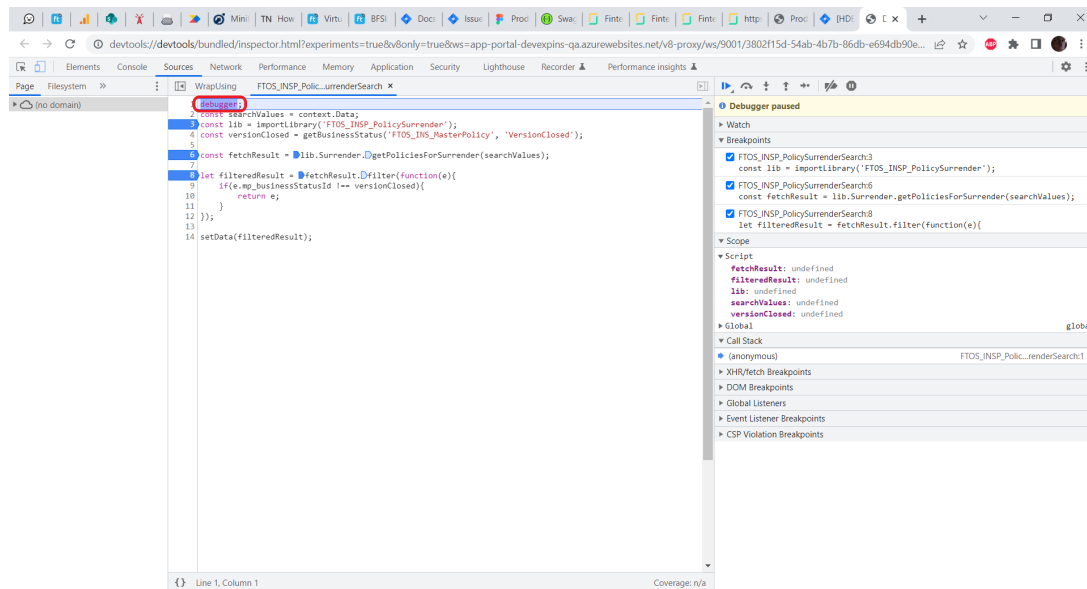
6. Click **Get debug Url** to retrieve the debug session URL.
7. **Copy** the debug session URL and paste it in your browser's address bar (we recommend **Google Chrome**), then press **Enter** to open the link in your browser's DevTools window.



8. The link initially opens a wrapper function for your script. Click the **Resume Script Execution** button in your browser's debugger.



9. Your script opens and its execution is paused at the start (a **debugger** instruction is inserted automatically at the beginning of your script).



10. Start debugging your code using **your browser's developer tools**.

## Debugging Log

The Debugging Log generates log events and continues the code execution. If the code execution breaks due to a **throwException** or to an unexpected error, the information will be recorded in the log.

Use the [ftos.logging.log](#) Server SDK function to generate custom log messages from code.

### Logging Destinations

The FintechOS Platform platform can send logging information to the following destinations:

- System console
- Local file storage
- A Seq structured log server
- An Azure Application Insights service

For more information, see the [Administration Guide](#).

### Logging Context

In addition to the actual log events, you can track the context in which a log event occurred (machine name, portal profile, thread ID, etc.), as well as define your own custom log context properties via server-side scripting (see "[Custom Log Context Properties](#)" on the next page). If a default log context property is not protected, you can use the custom log context property functions to edit it.

By default, the following log context properties are recorded:

Field	Protected	Description
base-machine-name	Yes	NetBIOS name of the machine that initiated the process.
correlation-id	Yes	GUID which correlates the initial call made from one of the internal FintechOS Platform components or from an external client application with the subsequent calls triggered between FintechOS Platform components.
portal-profile	Yes	Name of the originating FintechOS Portal profile.
application-name	Yes	Name of the application where the log event takes place.

Field	Protected	Description
base-thread-id	Yes	Currently managed Thread ID which generated the log event.
djc-form-name	No	Name of the form currently in use.
djc-form-id	No	ID of the form currently in use.
djc-record-id	No	ID of the business entity record currently in use.
djc-record-type	No	Attribute name of the primary key for the business entity record currently in use.
djc-journey-name	No	Name of the digital journey currently in use.
djc-journey-instance-id	No	ID of the digital journey instance currently in use.
djc-activity-id	No	ID of the form section currently in use.
djc-activity-type	No	Name of the form section currently in use.
djc-activity-instance-id	No	Digital journey instance step currently in use.
djc-journey-id	No	ID of the digital journey currently in use.

For instance, a typical log message may appear like this:

```
-[START]-----
Timestamp: 17/12/2021 02:10:15 PM
Message: Record saved successfully.
Severity: Information
Exception:
Properties: { CorrelationId: "23cdb1f5-0529-4377-b200-399efb8bc32f", portal-profile: null, base-machine-name: "FTOSBUCLP385", application-name: "Platform", automationscript: "sample-script-log-api" }
-[END]-----
```

### Custom Log Context Properties

To set up your own custom log context properties use the following Server SDK functions:

Function	Description
<a href="#">ftos.logging.setProperty</a>	Sets a custom log property to be included in log messages.
<a href="#">ftos.logging.getProperty</a>	Retrieves the value of a custom logging property.
<a href="#">ftos.logging.clearProperty</a>	Clears the value of a custom logging property.
<a href="#">ftos.logging.isPropertySet</a>	Checks if a custom logging property has been defined.

## Throw Exceptions

Break the automation script execution and display the message as specified within the `throwException` statement.

When the `throwException` method is called in an automation script, the passed error message is available to users.

Examples of how `throwException` can be used for debugging purposes:

```
throwException(serialize("test"));
throwException(serialize(result.TotalNetProfit));
throwException(serialize(context));
throwException(String(Topic[0].a_Topic));
throwException(JSON.stringify(result));
throwException(JSON.parse(result));
```

## JavaScript Exceptions

When parsing invalid Xml in automation scripts, JavaScript Exceptions are caught by `try catch` in JavaScript:

```
try
{
    // Load invalid XML
    var doc = ftos.Xml.Load( '<a> ... <' );
}
catch(err)
{
    // handle error
}
```

## Console Debugging

Server-side errors are displayed by the developer tools. During development and testing phases, developers are able to track errors raised on the server-side directly in the browser's Developer Tools.

Example browser developer tools to use for debugging automation scripts:

- [JSON Parser](#)
- [Javascript beautifier](#)

The error output displayed in the Console is particularly useful when raising issues. Include the error output in the issue description to provide a complete overview of the error and reduce the investigation time.

**NOTE**

Console debugging can be used only on development machines and in testing environments.

## On development machines

For IISExpress, on the development machine, open an elevated Command Prompt and run the following command:

```
C:\work\EBSCore\current>SETX\ebs-development-mode 1
```

For IIS, make sure that you set the variable at the system level (not on the user level):

```
C:\work\EBSCore\current>SETX\ebs-development-mode 1 /m
```

## On testing environments

To debug on testing environments, open the *web.config* file in a text editor and add the following section:

```
<appSettings>
  ...
  <add key="ebs-development-mode" value="1" />
</appSettings>
```

## JavaScript Engine Jint V3

With the launch of FintechOS v24.2, you can switch to Jint v3 JavaScript engine. However, keep in mind that this can trigger a number of mandatory changes that can be mitigated by refactoring some existing server automation scripts.

### Switching to Jint V3

In Studio, on the System Parameters page, find the **sys-workflowengine-type** parameter and change its value to Jint V3. Find more details about this in the [System Parameter](#) documentation page.

### Mandatory changes with Jint v3

#### Redefine the same variable multiple times in the same script

Mixing `let/const` and `var` for re-declaring or updating a global variable was allowed in Jint v2, but is not JavaScript ES6 compliant and not allowed in Jint v3. A variable declared with `let/const` must be declared only once global/in a code block. For example, the code below does not work with Jint v3:

```
// this declaration may be in a referenced workflow library
var a = 15;
// do something
const a = 20; // or let a = 20;
```

This also applies when a variable is declared in a referenced workflow library, not a library accessed with the SDK method `importLibrary`.

For reference, read the [Var, Let, and Const – What's the Difference?](#) article.

#### NOTE

You can use the `globalThis` object to store global variables or to check if a global variable exists before registering one, this object is available in all engine types.

```
globalThis.a = 15;
// do something
if (typeof globalThis.a == 'undefined') {
```

```
globalThis.a = 20;
}
```

Or you can use `var` to redeclare or update global variables, so you could replace `let/const` with `var` as a quick fix, albeit a dirty one:

```
// in a workflow library
var a = 15;
// do something
// suppose here was const a= 20;
// change to var instead of const to
// redeclare a variable with the same name in the workflow
var a = 20;
```

### IMPORTANT!

Redeclaring a variable that was declared earlier during the script execution (for example in a workflow library) might generate duplicated code or an unintended behavior. For example, hiding the business meaning of a variable used by other functions. It is recommended not to use variable names declared in workflow libraries or use a prefix for variable name.

This example also falls under the variable redeclaration topic:

```
function doSomething(paramName) {
  // in the function declare a variable with
  // the same name as the expected function parameter
  let paramName = 15;
  // do something
}
```

### IMPORTANT!

The code above is not syntactically correct, but Jint v2 and Jint v3 will execute it (v8 will not). Future versions of Jint v3 might fix this issue, so you should change the code and not declare a variable in the scope of a function with the same name as a parameter.

### Using exception message in try/catch

The following code was used prior to v24.2 to access an exception message, it works in v24.2 and above but only if the script engine is set to Jint v2:

```

try {
    // do something that throws an exception
}
catch (e) {
    // e is of type string
    if(e == 'expected message') {
        // do something based on exception message
    }
}

```

For v24.2 with Jint v3 or v8, the code should be refactored as:

```

try {
    // do something that throws an exception
}
catch (e) {
    // e is of type object - call new SDK method to get error message
    as string
    var m = getExceptionMessage(e);
    if(m == 'expected message') {
        // do something based on exception message
    }
}

```

The SDK method [setMessage](#) will work when an exception object is passed as parameter and will set the exception message as in Jint v2, so the code below doesn't need to be changed:

```

try {
    // do something that throws an exception
}
catch (e) {
    // e is of type object in Jintv3/v8
    setMessage(e);
}

```

## JavaScript ES6 Support

By default, the FintechOS Platform uses a Jint JavaScript interpreter with full support for the [ECMAScript 5.1](#) language specification (ES5) to run the server automation scripts. Starting with release 22.1.4, an alternate scripting engine based on the

Microsoft ClearScript library allows you to work with JavaScript using the [ECMAScript 6](#) language specification (ES6) (and also to ["Use the Debugging Server" on page 1191](#)).

## Switch between the Jint (ES5) and ClearScript (ES6) Scripting Engines

To change the JavaScript scripting engine, use the `sys-workflowengine-enable-debugging` system parameter:

- 0 - Enables the Jint (ES5) scripting engine (default).
- 1 - Enables the ClearScript (ES6) scripting engine.

For more information, see ["System Parameters" on page 1285](#).

## ES6 Compatibility Impact

If you switch from the Jint (ES5) to the ClearScript (ES6) scripting engine, make sure that your legacy code meets the [ES6 language specifications](#). Some of the potential compatibility issues are listed below.

### The JavaScript Date Object Is Always Set to UTC

A `DateTime` object constructed from a JavaScript `Date` object always represents a Coordinated Universal Time (UTC) and has its `Kind` property set to UTC.

The [EnableDateTimeConversion](#) flag specifies that the script engine will perform automatic conversion between .NET `DateTime` objects and JavaScript `Date` objects. This conversion is bidirectional and lossy.

This should not impact your existing code, since FintechOS Platform installations are set to UTC by default. For example, the code below returns the same values for Jint and ClearScript engines, with the observation that every `DateTime` value is converted to UTC:

```
let date = new Date();
let dateFromParts = new Date(date.getFullYear(), date.getMonth(),
date.getDate());

setData({
    date: date,
    dateFromParts: dateFromParts,
    invariant_date: new InvariantDate(date),
    invariant_dateFromParts: new InvariantDate(dateFromParts),
```

```

    invariantFromParts_date: new InvariantDate(date.getFullYear(),
    date.getMonth() + 1, date.getDate()),
    invariantFromParts_dateFromParts: new InvariantDate
    (dateFromParts.getFullYear(), dateFromParts.getMonth() + 1,
    dateFromParts.getDate())
  });

```

### Enforced ES6 JavaScript Syntax

ClearScript enforces strict ES6 JavaScript syntax, while Jint has a more relaxed syntax interpretation. Some of the situations that were not treated as errors by Jint, but are reported as syntax errors by ClearScript are listed below:

- If an object is declared with UPPERCASE property names, the same case needs to be used when reading those properties.
- Declaring the same variable multiple times.
- Using `InvariantDate().today` throws an exception. The correct ES6 syntax is `InvariantDate.today`. For more information, see the [Server SDK documentation](#).

### Processor Settings

When working with [automation blocks](#), the JSON settings and mappings must be correctly formatted - if you add a comma (',') after the last array element, you will receive an error similar to:

```

EBS.Core.Data.Services.DataConverterException:
The value for attribute unpaidAmount on entity FTOS_PYMT_
OutgoingPayment could not be converted to a valid entry.
Attribute type: numeric ---> System.InvalidCastException: Unable
to cast object of type 'Microsoft.ClearScript.Undefined' to type
'System.IConvertible'.

```

#### HINT

Use the no-code capabilities of the FintechOS Platform to edit automation block settings and mappings.

### Numeric Attributes and Undefined Variables

Updating a numeric attribute with a variable that is undefined will trigger an error.

```
EBS.Core.Data.Services.DataConverterException:
The value for attribute unpaidAmount on entity FTOS_PYMT_
OutgoingPayment could not be converted to a valid entry.
Attribute type: numeric ---> System.InvalidCastException: Unable
to cast object of type 'Microsoft.ClearScript.Undefined' to type
'System.IConvertible'.
```

### **ftos.data.query.getAlias Case Sensitive Attribute Names**

Fluent query field names are now case sensitive. For instance, assuming an entity with an option set attribute named `Os1`, using case insensitive alias field names works with Jint, but doesn't produce the same results with ClearScript:

```
var E1 = ftos.data.query.getAlias("FTOS_Entity");
var O1 = ftos.data.query.getAlias("OptionSetItem");
var vresults = ftos.data.query.from(E1)
    .leftJoin(O1).on(O1.OptionSetItemId.eq(E1.op1))
```

When ClearScript is enabled, `E1.op1` evaluates to undefined and the result of the query execution is different (it translates to an SQL join clause of `O1.OptionSetItemId is null`).

The correct syntax for ClearScript is:

```
var E1 = ftos.data.query.getAlias("FTOS_Entity");
var O1 = ftos.data.query.getAlias("OptionSetItem");
var vresults = ftos.data.query.from(E1)
    .leftJoin(O1).on(O1.OptionSetItemId.eq(E1.Op1))
```

## Server Automation Script Libraries

Use server automation script libraries to organize code on the server and reuse the code in automation scripts by attaching the automation script library to the desired automation scripts.

**NOTE**

You can attach multiple automation script libraries to automation scripts but you cannot attach automation script libraries to other automation script libraries.

Libraries allow you to easily maintain the code by modifying it once (within the automation script library). The code updates will be automatically propagated to all the automation scripts that are calling the automation script library.

The typical use case scenarios for server automation script libraries include:

1. Executing a specific block of code after an entity business status is changed. To use an automation script library to execute a block of code after a business status is changed, you need to create it first.
2. Call the automation script library from a server script using server-side functions.

To see the list of defined server automation script libraries, go to **Main Menu > Advanced > Server Automation Script Libraries**.

Server Automation Script Libraries list

<input type="checkbox"/>	Name	Digital Asset
	<input type="text"/>	<input type="text"/>
	FTOS.DateTimeUtils	
	FTOS.FTOSServices - 1.0.0	

## Create Automation Script Libraries

1. In FintechOS Studio, go to **Main Menu > Advanced > Server Automation Script Libraries**.
2. Click **Insert**. The Add Server Automation Script Library page appears.
3. In the **Name** field, enter the name of the library.

4. In the **Code** field, enter the script code (write code using [Server SDK functions](#)).

### NOTE

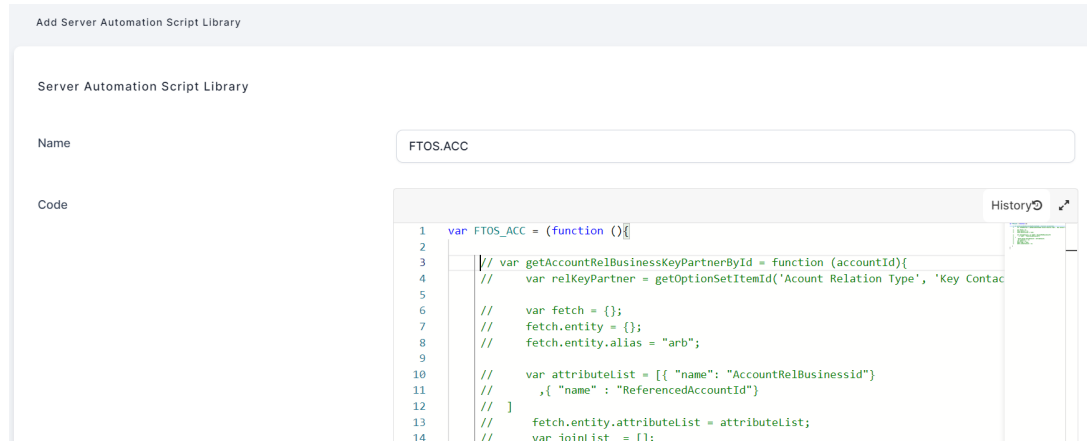
Make sure to include in the code at least a function you wish to call from your server automation scripts.

For example:

```
var FTOSEExample = FTOSEExample || (function () {
    var count = 100;

    this.getCount = function () {
        return count;
    }

    return {
        getCount: this.getCount
    }
});
```



To avoid calling the wrong methods and attributes, you can use the '\$m' mechanism when writing the code. For more information on how to use the \$m mechanism, see [Code Snippets Support for JavaScript](#).

5. Click **Save and close**. The server automation script library is saved and will be displayed in the Server Automation Script Libraries List.

# Style Sheets

Style sheets allow you to define your own styles for forms and digital journeys for better accessibility and improved usability. Using style sheets, you can apply your own text style, text color, padding, etc.

## Create a New Style Sheet

1. In FintechOS Studio, go to **Main Menu > Advanced > Style Sheets**.
2. Click **Insert**.
3. In the **Name** field, enter the name of the new style sheet.
4. In the **Code** field, provide the CSS classes that define your styles.

### NOTE

If in the custom style sheet you reference files (e.g., images), make sure that you're using the following referencing convention

**../custom/<filename>.extension** instead of  
**custom/<filename>.extension**; otherwise the files will not load.

You can limit the style impact on the current form driven flow and you can also overwrite css variables. For more details, see the sections below.

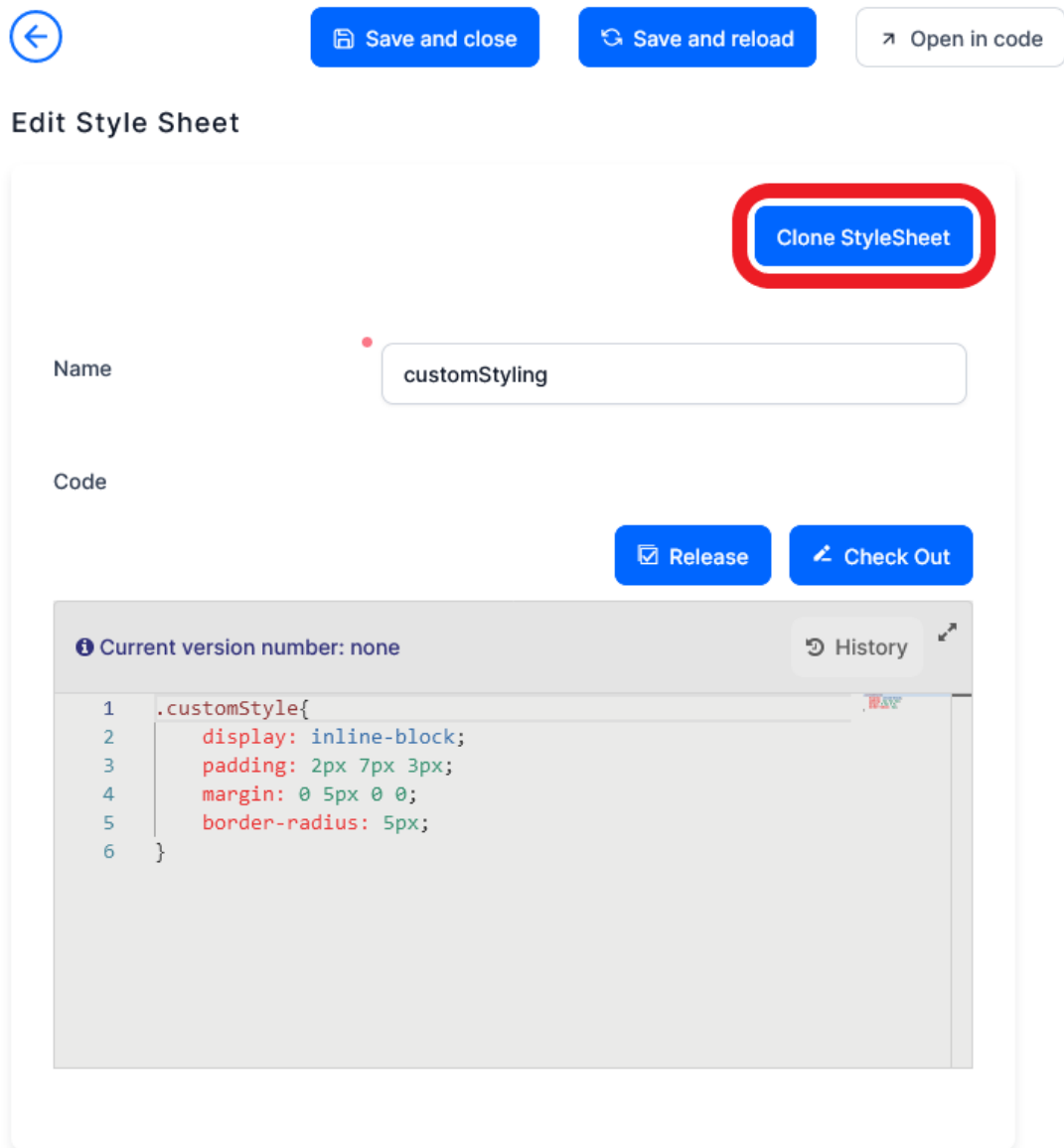
5. Click **Save and close**. The new style sheet will be displayed in the Style Sheets List page.

## Clone a Style Sheet

If you wish to create a variation based on an existing style sheet, you can clone the desired style sheet and use it as a starting point for further modifications.

To clone a style sheet:

1. Open the source style sheet in the editor and press the **Clone StyleSheet** button.



2. In the pop-up window that opens, provide a name for the clone.
3. Click **Clone** to create your new style sheet.

# Apply Style Sheets to Form Driven Flows (No-Code)

In the form or form driven flow's **General** section, use the **Style Sheets** property to select the style sheets you wish to apply.

The screenshot shows the 'General' configuration tab for a form in Fintechos Studio. The interface includes a top navigation bar with tabs numbered 1 to 10: General, Data Model, Flow Map, Steps, Field Options, Filtered Fields, Header Items, Actions, Advanced, and Security Roles. The 'General' tab is active. The form configuration fields are as follows:

- Name:** Cust
- Display Name:** Customer
- Description:** (Empty text area)
- Show Tooltips:** User Settings (with a dropdown arrow and edit icon)
- IsDefault:** ☐
- Is Default For Edit:** ☐
- Clone Form:** (Blue button)
- Properties:** (Expanded section)
  - Hide Business Workflow:** Default
  - Read Only:** Default
  - Disable Save Keyboard Shortcut:** Default
  - Show Bullets Progress Bar:** Default
  - Flow Title:** Default
  - Style Sheets:** (Highlighted with a red box) FTOS\_DFP x FTOS\_DFP\_II x
  - Hide Business Transaction:** Default
  - Hide Action Buttons:** Default
  - Disable prompt for unsaved changes:** Default
  - Save automatically data on leave:** No

You can select multiple style sheets for the same form. In this case, the order in which the style sheets will be applied matches the order in which you entered them.

## Apply Style Sheets Using Code

You can apply custom style sheets to the following components in the corresponding locations:

- **Entity Forms** - Before Events, After Events
- **Entity Form Sections** - After Events, After Section Save, Before Section Save
- **Custom Flows** - After Generate Js
- **Entity Views** - After Generate Js
- **HTML Widgets** - JavaScript
- **Charts** - After Generate Js

Use the [ebs.importStyleSheet](#) Client SDK method to import your custom style sheets.

If you want to remove your own styles applied on data forms and digital journeys, use the [ebs.removeAllImportedStyleSheets](#) method.

## Limit Style Impact to Current Form

To better qualify your selectors, use the following css class:

form\_{entityName}\_{formName}, with the spaces removed.

```
>.form_myEntity_myForm .mySelector{  
  color: #f00;  
}
```

## Overwrite Variables

If you want to style content based on its relationship with the parent and sibling content, use the **:root** selector. It enables you to target the highest-level element of the DOM, overwriting css variables.

```
:root{  
  --defaultFontSize: 12px;  
}
```

## Endpoints

Endpoints specify the location where clients can access on-demand "[Server Automation Scripts](#)" on page 1162. Endpoints can also be created from the routes section of a [business service component](#).

You can create an endpoint, attach security roles, assign an automation script to it, then on digital journeys or data forms run "[Call Actions](#)" on page 1171 to it.

## Create an endpoint

1. In FintechOS Studio, go to **Main Menu > Advanced > Endpoints**.
2. Click **Insert**.
3. In the **Name** field, type the name of the endpoint which will be used by the system.  
Endpoint names must begin with a letter and contain only alphanumeric characters and underscores.
4. In the **Display Name** field, type the name of the endpoint which will be displayed in the user interface.

5. (Optional) If you are working with actions groups, in the **Action Group** field, select the desired action group you have previously created. For more information on action groups, see ["Defining Action Groups" on page 263](#).
6. From the **Script** drop-down, select the on-demand automation script you previously created.

The screenshot shows the 'FintechOS Studio' interface with the 'General' tab selected. The form has the following fields:

- Name:** FTOS\_DFP\_FlowProcessorSettingsByType
- Display Name:** FTOS\_DFP\_FlowProcessorSettingsByType
- Action Group:** (Empty dropdown menu)
- Script:** FTOS\_DFP\_FlowProcessorSettingsByType
- Executes Save:** ☐
- Security Roles:** (Empty list)

At the bottom right, there are two buttons: '+ Insert existing' and 'X Remove existing'.

7. If the endpoint is set on a form's action group, when the action group is triggered in the user interface, you can save the form's data before the action group is executed by ticking the **Executes Save** checkbox.
8. If you are generating an endpoint from a route section part of a [business service component](#), pick the component from the drop-down list in the **Business service component** field.
9. Choose the **Route Name** from the drop-down list.
10. Pick the **Http Verb**, which is the request method to indicate the purpose of the request and what is expected if the request is successful. Pick from: delete, get, patch, post, put.
11. Click **Save and close** to save the endpoint or **Save and reload** to ["Attach a security role to an endpoint" on the next page](#).

You can now go to the digital journey or data form and run a [call action](#) to the server automation script.

## Attach a security role to an endpoint

To enforce security, you can choose which ["Security Roles" on page 1245](#) have the privileges to call actions on an endpoint.

When calling actions on endpoints which have no security roles attached, errors will occur and the actions will not be performed.

### NOTE

For backwards compatibility, the security role "Registered Users" is automatically added to all endpoints created in previous versions of FintechOS Studio. But, this does not ensure the backwards compatibility for unauthenticated portals. In this case, you need to manually configure or import the security roles assignment.

To attach a security role to an endpoint:

1. Open the endpoint, scroll down to the **Security Roles** section, and click **Insert existing**.
2. Select a security roles from the list by double-clicking or add a new security role (click **Insert** and provide all details required to create a new security role). The selected security role is displayed in the Security Roles section.

The screenshot shows the 'Security Roles' section of the FintechOS Studio interface. At the top, there is a title 'Security Roles' and two buttons: '+ Insert existing' and '× Remove existing'. Below these, there is a search bar with a magnifying glass icon and the text 'Name'. Under the search bar, there is a list of security roles: 'PolicyUser', 'Insurance Operational', and 'PolicyInfo'. The 'Insurance Operational' role is highlighted with a light blue background.

3. Click **Save and close** to save the endpoint.

## Customize the Input Parameters

You can define mandatory input parameters that must be passed to the script by the client process and you can enable intelligent code completion in the code editor for the script's input parameters.

To define an input parameter:

1. Open the endpoint in the editor and select the **Input Parameters** tab.
2. In the **Workflow Input Parameters** list, click **Insert** to add a new parameter to the list.
3. Fill in the input parameter's details:

### Name

Enter a name for the input parameter. The name must match the incoming variable name provided by the client process.

### Description

(Optional) Enter a description for the parameter.

### Data Type

Select the parameter's data type. Currently supported data types are any, numeric, boolean, string JavaScript object and custom types.

When the custom data type is selected, the Custom Type Definition field appears. There, you can manually add a JSON schema or create one automatically using the **Generate JSON schema** button. A JSON schema helps manage the required format for a JSON object.

### IMPORTANT!

The schema is used to validate the custom data-type parameters when triggering a server automation script using an [ebs.callActionByNameAsync](#) type of Server SDK function. The objects of the parameter should match the declared JSON schema.

Check the **JSON object properties are required** box if the properties of the JSON object are mandatory. If the properties are optional, either leave the box unchecked or delete them from the **"required":** field.



Generate JSON Schema from JSON object

```

1  {
2    "proposition1": false,
3    "proposition2": {
4      "prop2_1": ""
5    }
6  }

```

☒ JSON object properties are required

Generate JSON Schema

The value for the object properties of the **"type": "string"** have the corresponding format set in the JSON schema: some of the possible formats are **"date-time"**, **"time"**, **"email"**, **"uuid"**.

**NOTE**

After saving the input parameter, the description of the JSON object and its properties will be displayed when using intelligent code completion (IntelliSense) in the code editor.

```

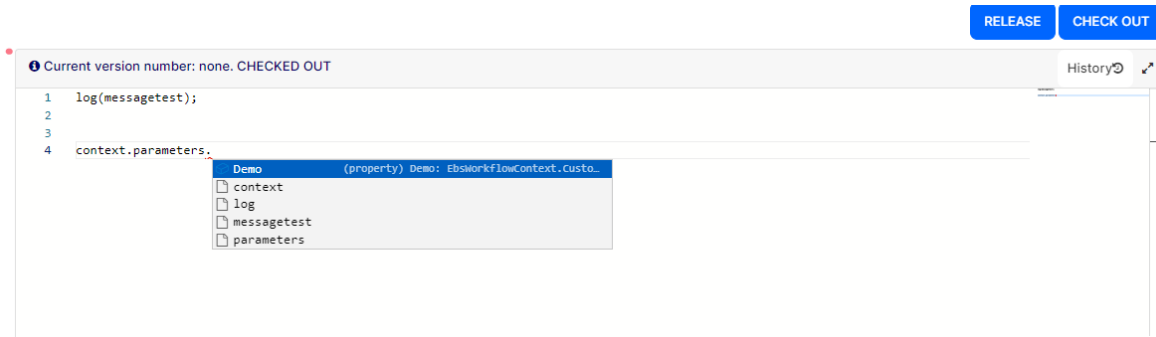
1  {
2    "type": "object",
3    "description": "Test description for object",
4    "required": [
5      "proposition1",
6      "proposition2"
7    ],
8    "properties": {
9      "proposition1": {
10       "type": "boolean",
11       "description": "Test description used"
12     },
13     "proposition2": {
14       "type": "object",
15       "description": "",
16       "required": [
17         "prop2_1"
18       ],
19       "properties": {
20         "prop2_1": {
21           "type": "string",
22           "description": ""
23         }
24       }
25     }
26   }
27 }
```

## Allow null or empty value

When unchecked, this parameter requires a value. Check to allow null, making it optional in the automation script. By default, this option is unchecked, meaning it requires a value.

4. Click **Save and close**.

Once defined as above, you can use the `context.parameters` property in the code editor to access the script's input parameters with intelligent code completion.

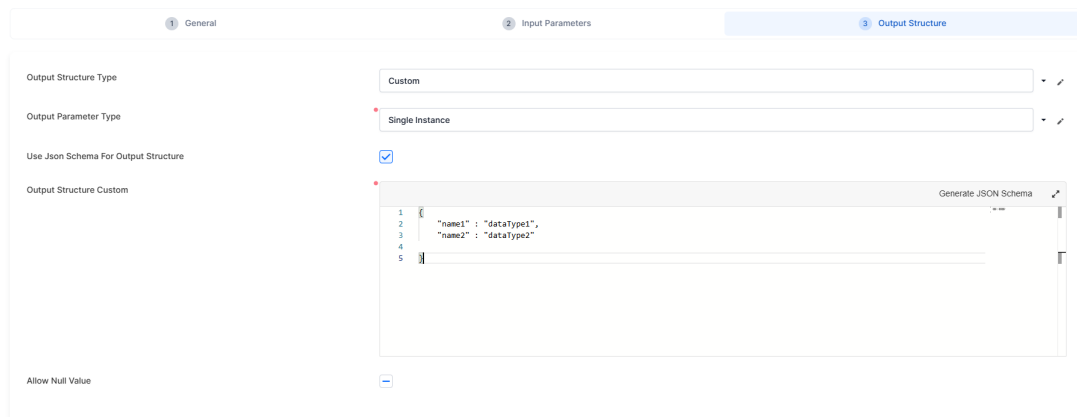


## Customize the Output Structure

You can customize the script's output structure to enable intelligent code completion for the result passed to the client-side callback function. You can map the output structure to an entity data model, or you can define your own custom structure. Also, you can specify if the output is in the form of a single object instance, or if it is a collection of objects each matching this output structure.

To define the script's output structure:

1. Open the endpoint in the editor and select the **Output Structure** tab.



2. Select the **Output Structure Type:**

- **Entity** - The output structure is based on an entity data model, matching the entity's attributes' names and types.

- **Custom** - Select this option if you wish to define the script's output structure manually.
  - **Boolean** - Select this option if the script returns either a *true* or *false* result, for instance if you wish to use it for validations.
3. Check the **Use Json Schema For Output Structure** to use a JSON schema for the output structure field.
  4. Select the **Output Parameter Type**:
    - **Single Instance** - The script result is a single object instance.
    - **Collection** - The script result is a collection of objects.
  5. If you selected an output structure type based on an entity, select the **Output Structure Entity**. This is the entity providing the data model for the output structure.
  6. If you selected a custom output structure , fill in the **Output Structure Custom** field in the following format:

```
{
  "name1" : "dataType1",
  "name2" : "dataType2"
}
```

#### NOTE

If you checked the **Use Json Schema For Output Structure** box, add in the Output Structure Custom field a JSON schema or create one from a JSON object using the **Generate JSON schema** button as mentioned in the ["Customize the Input Parameters" on page 1216](#) section.

7. When leaving the **Allow null value** option unchecked, this parameter requires a value. Check to allow null, making it optional in the automation script. By default, this option is unchecked, meaning it requires a value.

8. Click **Save and close**.

## Plugin Assemblies (Deprecated)

### IMPORTANT!

Plugin assemblies have been deprecated. Information below is provided for backward compatibility reference only.

Plugin assemblies allow you to use dynamic link libraries (DLLs) containing custom C# code which can be triggered by on-demand or event driven "[Server Automation Scripts](#)" on [page 1162](#). Plugin assemblies are triggered similarly to scripts on insert/update/delete.

This section walks you through the steps you need to follow to use plugin assemblies.

### 1 Add the Plugin Assembly

1. From the FintechOS Studio main menu, click **Advanced** > **Plugin Assemblies**. The Plugin Assemblies List page appears.
2. At the top-right corner of the page, click the **Insert** icon. The Add Plugin Assembly page appears.
3. Click **Select file**, browse for the plugin assembly (dll) file and select it.

**NOTE** The version of the dll file must be the same with the product version. When upgrading FintechOS Platform, make sure that you manually upgrade the plugins.

4. In the **Name** field, enter the plugin name that will be used by the system.

5. At the top-right corner of the page, click the **Save and reload** icon. The Edit Plugin Assembly page appears.

Now you can add the plugin and the UI processor.

## 2 Add the IEbsPlugin Plugin

In the Edit Plugin Assembly page, go to the **Plugins** section and click the **Insert** button. The Add Plugin page appears. In the **Name** field, type **IEbsPlugin**. At the top-right corner of the page, click the **Save and reload** icon.

## 3 Add the UI Processor

In the Edit Plugin Assembly page, go to the **UIProcessors** section and click the **Insert** button. The Add UIProcessor page appears. In the **Name** field, type **IEbsProcessor**. At the top-right corner of the page, click the **Save and reload** icon.

# Sequencers

Sequencers allow you to create complex alphanumeric sequences (a prefix followed by a sequence number) such as invoice numbers or other serial identifiers.

To configure a sequencer:

1. In FintechOS Studio, go to **Main Menu > Advanced > Sequencers**.
2. Click **Insert** to add a new configuration or open an existing one by double-clicking it.
3. Fill in the following:

Fields	Description
Name	Insert a name for the sequencer.

Fields	Description
Code	Insert a unique code for the sequencer. You will use this code to call the sequencer via server side scripts using the <a href="#">ftos.data.getNextSequenceNumber</a> function.
Prefix	Insert an alphanumeric prefix for the sequences. For example, if you set the prefix to AAA, the sequences will be AAA1, AAA2, and so on.
Padding	This is the number of characters for the sequence number. For example, if you set the padding to 4, the sequence numbers will be 0001, 0002, and so on.
RangeMin	Insert the minimum sequence number.
RangeMax	Insert the maximum sequence number.
Number	This is the current number in the sequence (which will be returned on the next sequencer call). You can use it to set the initial number of the sequence or to manually advance the sequence to a specific number. As the sequencer is used, this field updates automatically to indicate the current sequence number.
Start date	Date when the sequencer becomes available.
End date	Date when the sequencer will be disabled.

Fields	Description
Filter JS	<p>JavaScript code that allows you to process or filter the sequence numbers. The following variables are predefined.</p> <ul style="list-style-type: none"> <li>sequenceNumber - numeric value representing the current sequence number.</li> <li>skip - boolean value instructing the sequencer to skip the current sequence number. Default: false.</li> </ul> <p><b>Example:</b> Skip even numbers and return only odd numbers:</p> <pre>skip = (sequenceNumber % 2 == 0);</pre> <p><b>Example:</b> If the sequence number contains 666, skip it and replace it with 667. This implies, for example, that 666000 becomes 667000, and we actually skip 1000 numbers. This covers the left most match only, as the other matches will be skipped automatically.</p> <pre>var s = sequenceNumber.toString(); var m = s.match(/666/); if(m) {     var rightPaddingZeroes = s.length -     m.index - '667'.length;     s = s.substring(0, m.index) + '667' +     Array(rightPaddingZeroes + 1).join('0');     sequenceNumber = s; }</pre>

4. Click **Save and reload**.

## Roll the Sequencer Back or Forward

You can use the Sequencer Items grid at the bottom of the sequencer page to manually roll the sequencer back or forward. The sequencer will always jump to the first sequence number in the grid that is manually marked as not used and then resume numbering from there, reusing any allocated sequence numbers in the process.

For instance, in the example below, the sequencer is currently at number 30.

SequencerItems

+ Insert

× Delete

Export

Refresh

<input type="checkbox"/>	Name	SequencerId	Number	↑	IsUsed
	<input type="text"/>	<input type="text"/>	<input type="text"/>		(All) ▾
	ten	app001seq	10		<input checked="" type="checkbox"/>
	fifteen	app001seq	15		<input type="checkbox"/>
	twenty	app001seq	20		<input type="checkbox"/>
	twentyfive	app001seq	25		<input checked="" type="checkbox"/>
	forty	app001seq	40		<input type="checkbox"/>

If we manually uncheck the Is Used flag for sequence numbers 15 and 20 (as shown above), on the next call, the sequencer will reallocate number 15. Then, it will reallocate numbers 20, 21, 22, 23, etc. all the way to 30. Then, it will resume allocating new numbers: 31, 32, 33, etc.

To add a sequencer item:

1. In the Sequencer Items grid, click **Insert**.
2. Fill in the following fields:

Fields	Description
isUsed	<ul style="list-style-type: none"> <li>• Default state (neither checked, nor unchecked) - Doesn't affect the sequencer behavior. Leave it like this if you plan to activate the sequence number later by unchecking it.</li> <li>• Unchecked - Activates the sequence number and forces the sequencer to jump to it on the next iteration (provided there are no other lower sequence numbers set up identically).</li> <li>• Checked - Doesn't affect the sequencer behavior. Manually unchecked sequence numbers will transition to the checked state once they are allocated by the sequencer.</li> </ul>
SequencerID	This field is automatically filled with the sequencer code.
Number	Insert the sequence number you want to customize.
Name	Insert a name for the customized sequence number.

3. Click **Save and close**.

## Retrieve a Sequence Number from the Sequencer

You can call the sequencer using "[Server Automation Scripts](#)" on page 1162 via the `ftos.data.getNextSequenceNumber` Server SDK function.

### HINT

To optimize concurrent access to the sequencer, avoid locking the sequencer for an extended time. Try to optimize your scripts' code so that the sequencer logic runs

only when necessary. Also, try to trigger the sequencer call toward the end of the script, so that the lock remains set for a shorter time.

# Code Blocks

Code Blocks are predefined code templates you can insert in your scripts. Code blocks are designed to be extensible and configurable. You can define your own code blocks and configure them based on your needs.

**NOTE**  
Do not confuse script libraries with code blocks. Unlike the "Client Script Libraries" on page 1156 and "Server Automation Script Libraries" on page 1206 which you call directly in scripts, you need to adapt a code block to fulfill your needs (attributes, etc.). However, you can call existing script libraries in your code blocks.

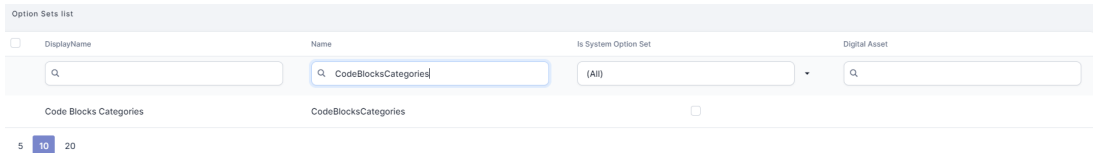
Code blocks are grouped into categories to help you easily filter the relevant blocks.

## Define Code Blocks Categories

Categories help you group code blocks.

To add a code block category:

- 1. In FintechOS Studio, go to **Main Menu > Admin > Option Sets**.
- 2. Search for the **CodeBlocksCategories** option set (you can filter either the CodeBlocksCategories name or the Code Blocks Categories display name).



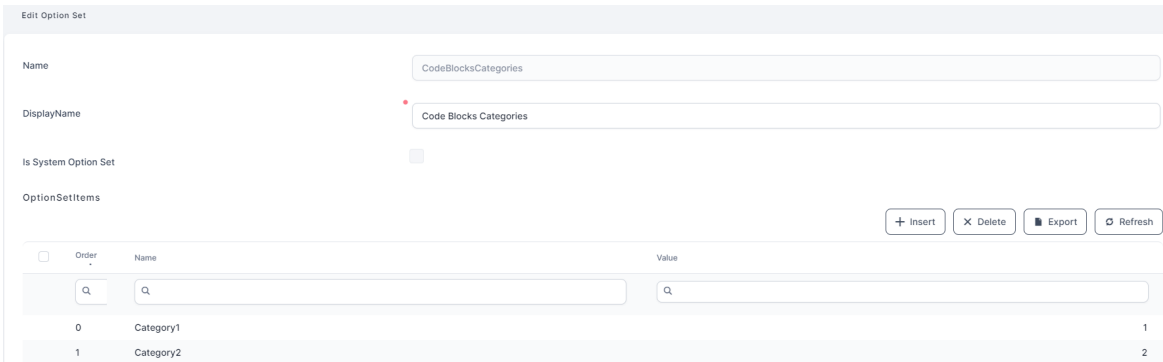
- 3. Double-click the option set.

4. Click **Insert**.
5. In the **Name** field, type a name that will be used by the system and in the **Display Name** field, type the name that will be displayed in the user interface.



6. Click **Save and close**.

Add as many categories as best suit your needs. As you create categories, they are displayed in the Option Set Items section.



## Add Code Blocks

1. In FintechOS Studio, go to **Main Menu > Advanced > Code Blocks**.
2. Click **Insert**.
3. In the Add Code Block page, fill-in the following fields:

Field	Description
Name	The category name used by the system.

Field	Description
Display Name	The category name to be displayed in the user interface.
Usage Location	The place where the code block will be available: <b>Client Side</b> or <b>Server Side</b> .
Documentation	If you have the code block documented, provide the URL to the documentation.
Category	The categories to which the code block belongs to.
Description	Provide a brief description of the code block to help others easily identify the scope of the code block.
Code	JavaScript code which will be inserted when using the code block.

Add Code Block

Code Block

Name

Display Name

Usage Location

Documentation

Category

Description

Code

```

1 ebs.getByIdAsync("myEntity", "[recordGuid]")
2   .then(function(record){
3     if (record && record["attributeName"]) {
4       var attributeValue = record["attributeName"];
5     }
6   })
7   .catch(function(err) {

```

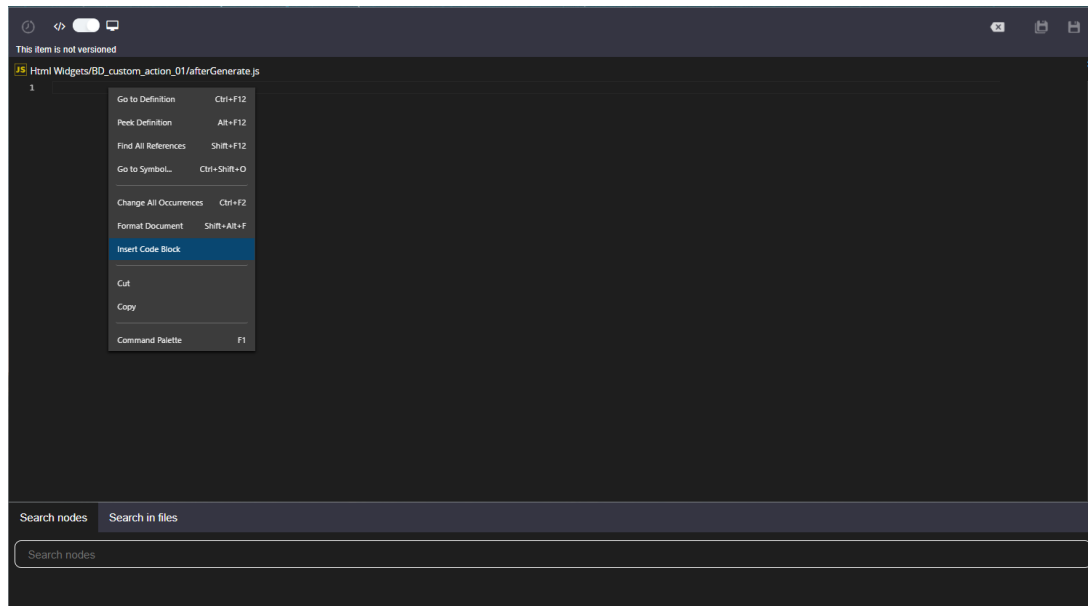
#### 4. Click **Save and close**.

Add as many code blocks as you need.

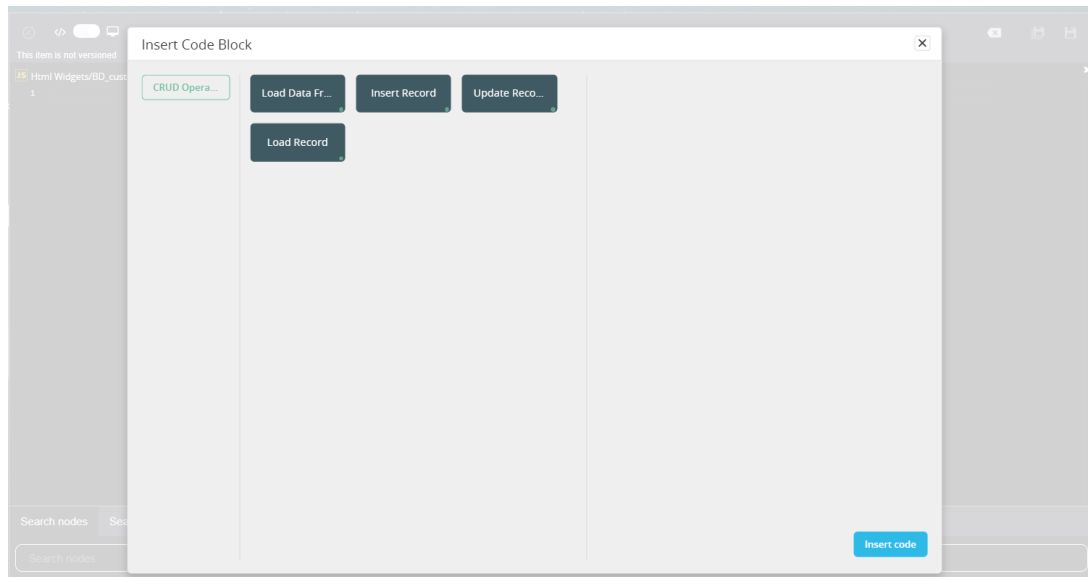
## Use Code Blocks

You can use code blocks in any code editor (either the "[Advanced Code Editor \(deprecated\)](#)" on [page 1142](#) or regular editors):

1. In your code editor, right click and, from the context menu, select **Insert Code Block**.



2. The Insert Code Block page appears.



3. Filter the code blocks by selecting the relevant category.
4. Double click on the desired code block or click it and select **Insert code**. The code from the selected code block is added on a new line at the cursor position in the editor.

5. Modify the inserted code template based on your requirements.

## DB Tasks

A stored procedure is a set of Structured Query Language (SQL) statements which are saved in the database (DB).

If your data processing requires substantial computing resources, (e.g. automated calculation of data from imported tabular files), we recommend using DB tasks to execute such stored procedures under the 'Ebs' database schema.

### 1 Add DB Tasks

To add a DB task:

1. In FintechOS Studio, go to **Main Menu > Advanced > DB Tasks**.
2. Click **Insert**.
3. In the Add DB Task page, type a **Name** for the DB task. This is the name used by the system. You will need it when executing the DB task.

#### IMPORTANT!

When naming your stored procedure in SQL, make sure to start the name with "ebs." to make the procedure visible later on in the DB Objects tab of [Digital Assets](#) page in Studio.

4. In the **Statement** field, type the name of the stored SQL procedure.

If you want to add security roles to DB tasks, at the top right corner of the page, click **Save and reload** and proceed to "[2 Add Security Roles to DB Tasks](#)" on the next page. Otherwise, click **Save and close**.

**IMPORTANT!**

If no security roles are added, any user is able to execute the DB task.

## 2 Add Security Roles to DB Tasks

If your business case requires that the DB tasks are executed only by users with designated roles within your organization, in the Edit DB Task page, click the **Insert existing** button at the top of the Security Roles section.

The screenshot shows the 'Edit DB Task' form. The 'Name' field is populated with 'MKT\_GetStageInstanceActivityDate' and the 'Statement' field with 'uspFTOS\_DCO\_ReturnGeneratedActivityDate'. Under the 'Security Roles' section, there are two buttons: '+ Insert existing' and '- Remove existing'. Below these buttons is a search bar with a magnifying glass icon. Underneath the search bar, a list of security roles is displayed, including 'Debugger Users' and 'Developer'.

A pop-up listing all defined security roles appears. Double-click the desired security role from the list. The pop-up closes and the selected security role is displayed in the Security Roles section.

Add the security roles that comply with your organization's security policies.

After you finish adding the security roles, click **Save and close** to save the DB task updates.

## 3 Execute DB Tasks

Users who have a security role defined on a DB task can execute that DB task in "Server Automation Scripts" on page 1162 by using the `ftos.data.executeDbTask` function:

**Syntax:**

```
function executeDbTask(dbTaskName: string, parameters: any):
  IExecuteDbTaskResult
```

### Request Parameters

Parameter	Type	Description
dbTaskName	string	The name of the DB Task registered in FintechOS Studio.
parameters	any	JSON object containing key-value pairs matching the names and values of the database stored procedure's input parameters.

### Returns

The function returns an array of objects mapped to the columns from the result set of the stored procedure.

**NOTE**

If the DB Task has no security roles associated, all users can execute the DB task.

## Operation Authorization

Operation authorization allows you to initiate a multi-factor authorization process on-demand, whenever you wish to secure an operation in a form driven flow. This is particularly useful for sensitive actions, such as online payments, when you wish to impose an additional layer of security on top of the regular user authentication based on username and password. The authorization process is described below:

1. The authorization process is initiated either explicitly by the user or automatically when the user progresses past a specific point in the form driven flow.
2. The user receives a secret pass-code via an alternate communication channel (such as email) while a custom authorization page is displayed on screen.
3. If the user authorizes the operation by entering the pass-code in the authorization page, a custom authorization script is run to continue the flow as desired. Otherwise, a different rejection script is used to guide the flow on a different path.

The sections below describe the components necessary to set up operation authorization and how to initiate authorization for an operation.

## 1 Prerequisites

Before you set up authorization for a specific operation, make sure the following items are in place:

- **Bearer Entity** - Make sure that the authorization process references a specific entity in the data model for data persistence. For instance, a payment authorization process could use a payment entity that includes attributes for the payment amount, source IBAN, destination IBAN, etc., as well as ["Business Workflows" on page 418](#) states to track whether the payment authorization is pending, approved, or rejected.
- **Communication Channel** - A provider that allows you to send the pass-code via an alternate communication channel, such as email. For more information, see the [Omnichannel Communication Automation documentation](#).

## 2 Details Automation Script

The details automation script provides the inputs for the pass-code generation and the available tokens for the pass-code message and authorization page templates. The script output must include a JSON object called `operation`, containing keys for the `operationDetails` (the components that will be used to generate the pass-code) and `operationInfo` (the components that will be available as tokens for the pass-code message and authorization page templates).

For example:

```
setAdminMode(true);

var E = ftos.data.query.getAlias('myPayment');
var rows = ftos.data.query.from('myPayment', E)
    .where(E.myPaymentId.equals(context.Data.recordId))
```

```

.selectColumns(E.PayeeName, E.PaymentAmount, E.DestIBAN,
E.SourceIBAN)
.executeAndMapComplex({ entity: E});
log("Rows = ");
log(rows);

var operation =
{
  operationDetails: {
    paymentAmount: rows[0].entity.PaymentAmount,
    sourceIBAN: rows[0].entity.SourceIBAN,
    payeeName: rows[0].entity.PayeeName,
    destIBAN: rows[0].entity.DestIBAN
  },
  operationInfo: {
    paymentAmount: rows[0].entity.PaymentAmount,
    sourceIBAN: rows[0].entity.SourceIBAN,
    payeeName: rows[0].entity.PayeeName,
    destIBAN: rows[0].entity.DestIBAN
  }
};

setData(operation);

```

### 3 Pass-Code Message and Authorization Page Templates

Use [content templates](#) and ["Custom Flows" on page 283](#) to create templates for the pass-code message and the authorization page respectively.

#### Template Tokens

- For the pass-code field:
  - In the pass-code message, use the `{OTPCode}` token.
  - In the authorization page, use the value defined for the `formTag` key, as specified in the Authorization Scenario field of the ["5 Action Authorization Type" on page 1237](#).

- To include tokens for the operation information (such as the payment amount, payer name, account numbers, etc.), use the values defined in the `operationInfo` key of the "2 Details Automation Script" on page 1234.
- Use the `{btn:acceptButton:btn}` and `{btn:rejectButton:btn}` tokens to place **Accept** and **Reject** buttons on the authorization page template. These will be replaced with a button tag with localized text. The localization resources are `AuthorizePayment_Accept` and `AuthorizePayment_Reject`.

### Pass-Code Message Template Example

←

CONTENT TEMPLATE NAME  
SmsTemplate

Save and close

Save and reload

### Edit Content Template Item

📄

Name

AC\_Item

📄

Subject

Item1

📄

Channel

SMSOtp

↓

✎

Culture

English GB

↓

✎

📄

Max Message Length

160

📄

Remaining Characters

102

📄

Send Message as SMS

☒

Enable Highlight

☒

📄

Tokens

↗

1 Use verification code {OTPCode} to authorize your payment.

Edit Channel Extended Properties

### Authorization Page Template Example

```
<ebs-resource data-resource-key="customPage" data-culture="ro-RO">Autorizezi plata sumei de {!paymentAmount!} din contul {!sourceIBAN!} pt {!payeeName!} in contul {!destIBAN!}?</ebs-resource>
<ebs-resource data-resource-key="customPage" data-culture="en-GB">Do you authorize payment of {!paymentAmount!} from account number {!sourceIBAN!} for {!payeeName!} to account {!destIBAN!}?</ebs-resource>
<div data-resource-key="customPage">
</div>
<div class="show-mask">OTP:{OTP}</div>
{btn:acceptButton:btn} {btn:rejectButton:btn}
```

## 4 Authorization and Rejection Automation Scripts

Once the user accepts or rejects the operation, use the corresponding scripts to advance the flow along the desired path. These server-side scripts typically include instructions to navigate to a desired step in the flow. For example:

```
log("Payment authorized.");
setNavigationEdit("Payment", context.Data.recordID, "Payment", "2");
```

## 5 Action Authorization Type

You can define multiple types of authorization operations that you can apply in various scenarios, depending on the settings you wish to use for the communication channels, pass-code and authorization page templates, authorization and rejection automation scripts, etc.

To create an action authorization type:

1. In FintechOS Studio, go to **Main Menu > Advanced > Authorization Operation Types**.
2. Click **Insert** to open the authorization type editor.

Edit Action Authorization Type

The screenshot shows the 'Edit Action Authorization Type' form. It has the following fields and values:

- Name:** Test
- Display Name:** Test
- Bearer Entity:** Test
- Details Automation Script:** DetailsAutomation
- Authorization Scenario:** A code editor showing the following JSON:
 

```
1 [{
2   "name" : "otp",
3   "formTag" : "OTPx",
4   "messageTemplateName" : "SmsTemplate",
5   "channelProviderName" : "GatewaySmsOTP"
6 }]
```
- Session Expiration Time in seconds:** 600
- Custom Page Template:** CustomPageTemplate
- On Authorization Automation Script:** AuthorizeScript1
- On Reject Automation Script:** RejectScript1

3. Fill in the following fields:

Field	Description
Name	Name of the action authorization type.
Display Name	Name of the action authorization type as it will be displayed in the user interface.
Bearer Entity	Entity referenced by the authorization process for data persistence. For details, see <a href="#">" 1 Prerequisites" on page 1234</a> .
Details Automation Script	Server-side on-demand script endpoint containing the inputs for the pass-code generation and the available tokens for the pass-code message and authorization page templates. For details, see <a href="#">" 2 Details Automation Script" on page 1234</a> .

Field	Description
Authorization Scenario	<p>JSON object containing keys for the:</p> <ul style="list-style-type: none"> <li>• Name of the scenario.</li> <li>• Token used for the pass-code input field in the authorization page template (for details, see "<a href="#">3 Pass-Code Message and Authorization Page Templates</a>" on page 1235).</li> <li>• <a href="#">Content template</a> used for the pass-code message.</li> <li>• <a href="#">Channel provider</a> used to send the pass-code message.</li> </ul> <p>For example:</p> <pre>[{   "name" : "otp",   "formTag" : "OTP",   "messageTemplateName" :     "ContentTemplate",   "channelProviderName" :     "GatewayEmailOTP" }]</pre>
Session Expiration Time in Seconds	Number of seconds the user has to authorize the operation from the time the pass-code message is sent. If this duration is exceeded, the operation is rejected.
Custom Page Template	Template for the authorization page. For details, see " <a href="#">3 Pass-Code Message and Authorization Page Templates</a> " on page 1235.
On Authorization Automation Script	Server-side on-demand automation script endpoint executed if the operation is authorized. For details, see " <a href="#">4 Authorization and Rejection Automation Scripts</a> " on page 1237.

Field	Description
On Reject Automation Script	Server-side on-demand automation script endpoint executed if the operation is rejected. For details, see " <a href="#">4 Authorization and Rejection Automation Scripts</a> " on page 1237.

4. Click **Save and Close**.

## Initiate an Operation Authorization Process

Once you have an " [5 Action Authorization Type](#)" on page 1237 in place, you can initiate an authorization process, either when the user explicitly interacts with a control in the user interface, or automatically when progressing to certain stage in the form driven flow.

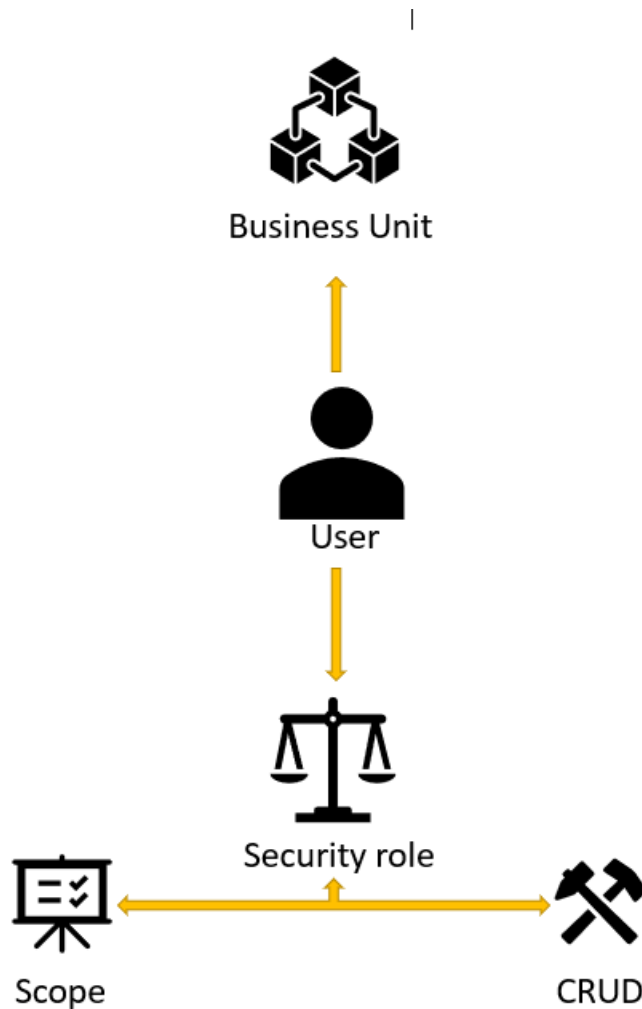
To initiate an authorization process, use the `requestOperationAuthorization` Server SDK function. For example:

```
requestOperationAuthorization(context.id, "payment", "0093B783-
E7A1-9C10-79C9689AB56B", "1C9A6B83-B1A4-C9A0-7C096C9A0501",
"1A93C703-97AB-0C30-7929A69AB86B");
```

# Security

Security design is essential to protect information from being mishandled by users and ensure that users have access to information based on business needs.

To set up the organizational structure, you need to create the business units, security roles, and assign users the appropriate security roles to map the job-related responsibilities with the required level of access privileges within the platform.



The security of our technology rests on four major poles: **data encryption, authentication, authorization, data governance and data audit**. FintechOS uses several methods for authentication and a role-based access control; data ownership is given by security roles, while sensitive data is being protected because it can become anonymous.

For example, to build an internal security mechanism, a user will have to build an organizational chart that includes business units, security roles and users. Create business units beside the root one, configure the security roles that will be given to the users using CRUD privileges, assign users to each role allowing each to see what they need by associating the security role to the digital journey/analytics.

This section covers the following topics:

---

<b>Business Units</b> .....	<b>1243</b>
Create Business Units .....	1243
Remove Business Units .....	1244
Synchronization in IDP .....	1244
<b>Security Roles</b> .....	<b>1245</b>
Default Security Roles .....	1245
1 Add the security role .....	1247
2 Assign security items to the security role .....	1248
3 Edit Security Roles .....	1250
<b>System Users and Service Accounts</b> .....	<b>1251</b>
Add System Users .....	1252
Create Service Accounts .....	1254
Edit System Users or Service Accounts .....	1255
Service Accounts in Digital Assets .....	1256
<b>Recover Password (for users)</b> .....	<b>1257</b>
<b>Share Access to Uploaded Files</b> .....	<b>1258</b>

# Business Units

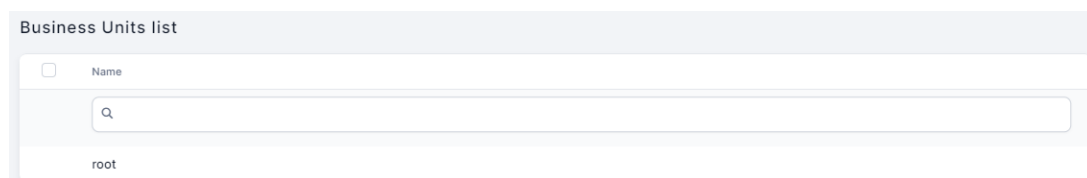
Business units are the foundation of the security structure. Each user must be part of a business unit. When FintechOS is installed, a business unit is created by default - the root business unit. You can rename it, but you cannot remove or disable it. A user configured under the root business unit can see all the records of the entities based on granted access rights.

To define the organization structure, super users can add as many business units as necessary to fulfill the need of configuring several levels of access to information for specific to groups within the organization.

Depending on your organizational needs, there is always only one root business unit, and at least one business unit. The root business unit acts as the top level of the organizational hierarchy, and all other business units are its children. The root business unit is the company, while the child business units are subsidiaries or departments within the company.

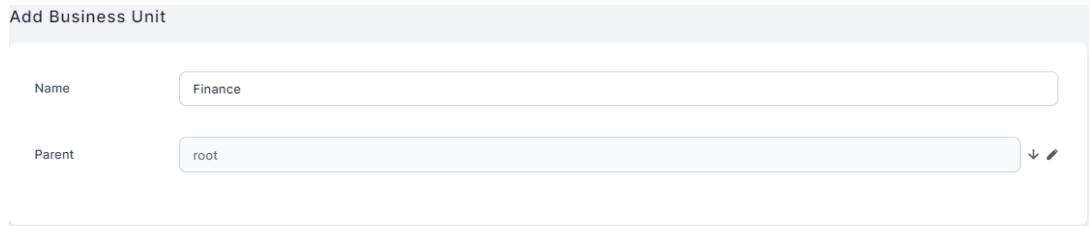
## Create Business Units

1. From the menu, click **Security > Business Units**. The **Business Units List** page appears.



2. Click **Insert** . The **Add Business Unit** page appears.
3. In the **Name** field, enter the name of the business unit.
4. From the **Parent** drop-down, click the down-arrow. A pop-up appears listing all existing business units.

5. If this is the first business unit you add, then select root otherwise, select the parent business unit so that it reflects your organization structure. The figure below shows an example on how to add the first business unit.



The screenshot shows a form titled "Add Business Unit". It contains two input fields. The first field is labeled "Name" and contains the text "Finance". The second field is labeled "Parent" and contains the text "root". To the right of the "Parent" field, there is a small downward arrow icon and a small pencil icon, indicating a dropdown menu and an edit function respectively.

6. When you finish adding business units, click **Save and close** . The page closes and the business unit is displayed in the **Business Units List** page.

## Remove Business Units

### NOTE

You cannot remove business units that contain users. If you try deleting such business units, an error message informing you that the operation has failed is displayed.

1. To remove business units, from the menu, click **Security > Business Units**.
2. The Business Units List page is displayed.
3. Select the ones that you want to remove and click **Delete**.

## Synchronization in IDP

Business units can be synched in FintechOS Identity Provider. This is useful in some complex scenarios when a user is part of 2-3 groups in other identity and access management solutions, such as Azure Active Directory, which is authorized in FintechOS through identity brokering. Those groups are automatically synchronized as business units in the FintechOS Identity Provider, each business unit is inserted and relations are set with the system user.

To enable/disable this sync, go to the [Configuration Manager](#) on your environment and search for the `ShouldSyncBusinessUnit` key under app-settings in the auth section. The default value is 0 (false). To turn the feature on, set the value to 1 (true) and restart the Studio.

## Security Roles

A security role is a set of privileges and the level of access to various actions/functions within the platform. Security roles allow you to configure the security items, that is, the access privileges on CRUD operations for entities in the data model.

Users with elevated privileges (admin users) can control data access by setting up the organizational structure to protect sensitive data and configure various organization layers to allow communication, collaboration, or reporting.

You can grant even more granular access privileges in FintechOS, by associating security roles to digital journeys, workflows, analytics, Portal Profiles, etc. Such security roles are then associated to user accounts, hence those users are able to access those digital journeys, workflows, analytics, or Portal Profiles. Such an example is given in Access to Portal Profiles based on Security Roles. The data is automatically filtered based on the privileges and level of access defined within the security role via the security items.

The most granular level of access privileges you can grant to users in FintechOS is the attribute level.

## Default Security Roles

- **Debugger Users** - This is a development role. It is used by the implementation team to debug issues on the Portal using the Debugger in the kit.
- **Developer** - Assigned to users who work with configuration items (such as entities and form driven flows) in order to develop digital solutions. Developers can create and

configure configuration items, but they cannot lock/unlock them. They can also create Digital Solutions and import/export Digital Solutions Packages.

- **DjAnalytics** - This role ensures read access rights for users who work with built-in analytics in digital journeys.
- **Guest** - Inherited by the platform; it doesn't have any special platform access meaning.
- **Integration Users** - A role designed for integration with other systems. It is not an actual user, but rather a process that authenticates and calls various functions exposed inside the platform.
- **JobServer** - This role is used by the JobServer service to execute scripts from the platform with a specific schedule (see [Schedule Jobs](#)).
- **Observability** - Provides access to observability features inside the Studio.
- **Product Factory** - Allows access to the Product Factory, unless you have a User Admin account.
- **Registered Users** - Users with this security role have access rights to edit their account from My Account and to access a minimum list of entities in order to log in without errors in the application.
- **Release Manager** - Users with this security role can lock/unlock Configuration Management and create Digital Solutions Packages.
- **Security Role Management** - Allows security roles management, but no system users management. Security role managers can manage security roles, but don't have access to system user accounts.
- **User Management** - Users with this role can manage the user accounts and user roles without having elevated privileges. System users who have been granted the User Management security role cannot manage existing Administrator users.

- **User Admin** - Allows system users management, but no user roles management. User admins can manage system users and allocate user roles to them, but cannot create/delete user roles or modify existing user roles.
- **Widget** - Inherited by the platform; it doesn't have any special platform access meaning.

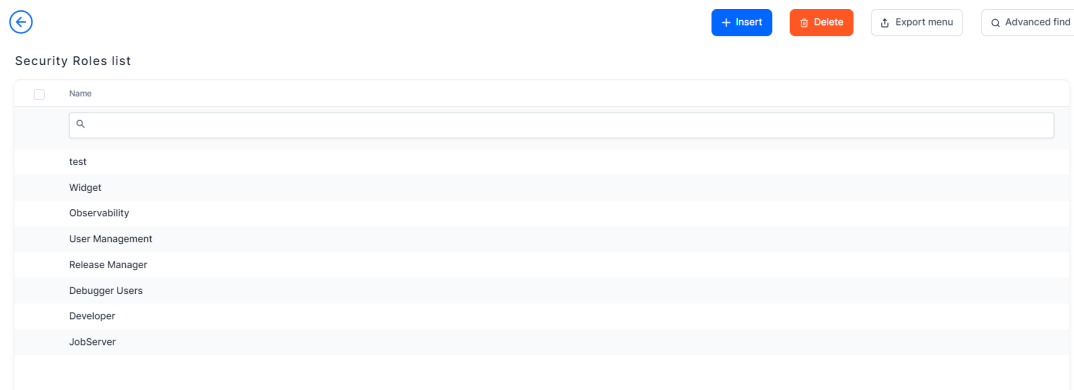
New security roles can be added to the list depending on your business needs. Once you configure the security roles, you can attach them to configuration items (such as form driven flows or reports) on which you wish to enforce access control.

### IMPORTANT!

Default security roles must not be used for securing business solutions.

## 1 Add the security role

1. From the menu, click **Security > Security Roles**. The **Security Roles List** page opens.



2. Click **Insert** . The **Add Security Role** page opens.
3. In the **Name** field, type a name for the new security role.
4. Click **Save and reload** . The **Edit Security Role** page opens.

## 2 Assign security items to the security role

Security items specify the entities and the privileges (CRUD operations on those entities) that the security role grants access to.

Users assigned with a security role are able to perform only the CRUD operations on entity records assigned in the security items.

To add a security item, follow these steps:

1. From the **Security Items** section, click **Insert**. The **Add Security Role Item** page is displayed.
2. In the **Entity** field, type the entity name or click the down-arrow and select it from the list.
3. In the **Security Scope** field, type the level of access or click the down-arrow and select it from the list:
  - **Business Unit** - Privileges to all records owned in the business unit to which the user belongs to.
  - **Organization** - Privileges to all records in the organization regardless of their owner.
  - **Parental** - Privileges to all records owned in the business unit to which the user belongs to, including privileges to the records owned in the child business units.
  - **User** - Privileges to the records owned by the user or assigned to the user.
  - **Custom** - If the security item entity has the *Support Data Ownership Custom Scope* option enabled, allows the user account to receive custom data ownership privileges over specific records of the entity. For more information, see the Server SDK documentation for the following methods:

- [ftos.identity.authz.createGrantTag](#) - Create a custom data ownership tag.
  - [ftos.identity.authz.setGrantTagOnRecord](#) - Assign a custom data ownership tag to an entity record.
  - [ftos.identity.authz.getGrantTagOnRecord](#) - Retrieve any custom data ownership tag assigned to a record.
  - [ftos.identity.authz.RemoveGrantTagOnRecord](#) - Unassign a custom data ownership tag from a record.
  - [ftos.identity.authz.setGrantTagToUser](#) - Assign a custom data ownership tag to a user account.
  - [ftos.identity.authz.RemoveGrantTagToUser](#) - Unassign a custom data ownership tag from a user account.
4. In the **Operations** field, select the record-level privileges (CRUD operations) for which you want to assign permissions. You can choose one or more of the following:
- Create
  - Read
  - Update
  - Delete
5. Click **Save and reload**.
6. If your security role item is an entity with attributes tagged as "[isSecurable](#)" on [page 88](#), you can use the **Permissions for secured attributes** table at the bottom of the page to set the Create, Read, and Update permissions for the entity's secured attributes:
- **Restricted** (default) - Prevents access to the attribute.
  - **Allow** - Allows access to the attribute.

- **Explicit deny** - Prevents access to the attribute even if another security role allows access.

Add Security Role Item

Security Role Item

Entity: Account

Security Scope: User

Operations: ☒ Create ☒ Read ☒ Update ☒ Delete

Create Read Update

Permissions for secured attributes	Restricted	Allow	Explicit deny
Email	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

7. Click **Save and close**.

### 3 Edit Security Roles

To edit a security role, from the menu, click **Security > Security Roles**. The **Security Roles List** page appears. Double-click the security role that you want to edit. The **Edit Security Role** page appears. You can edit security roles by changing their name or by adding or removing security role items.

Edit Security Role

Name: AS\_Role

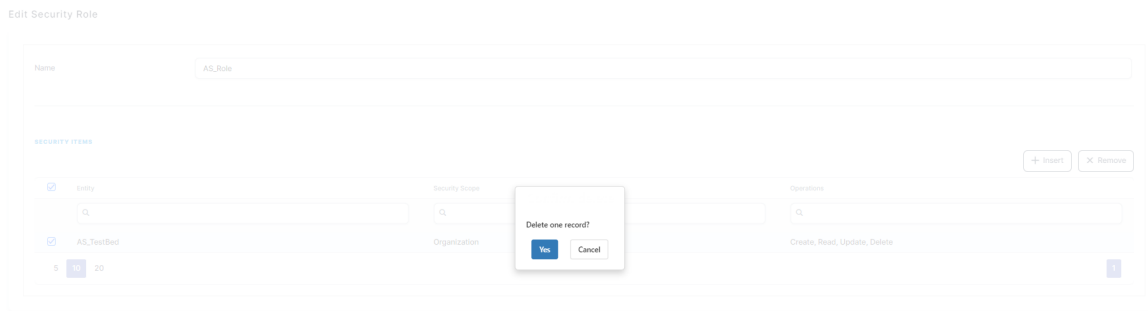
**SECURITY ITEMS**

+ Insert X Remove

Entity	Security Scope	Operations
AS_TestBed	Organization	Create, Read, Update, Delete

5 10 20 1

To remove a security item, scroll-down to the **Security Items** section, select the item that you want to remove and at the top of the Security Items section, click **Delete** . A confirmation dialog appears. Click **Yes** and the selected security item is removed from the system.



Make the desired security role changes and at the top-right corner of the page, click **Save and close** to save the changes.

## System Users and Service Accounts

As with any secured application out there, to access FintechOS Studio you need a user account, referred as system user accounts. These user accounts can be created in Studio, by navigating to **Security > System Users** from the main menu.

Another type of users in FintechOS are service accounts. These are a type of privileged accounts that gives the user elevated rights within the platform. Such accounts are set up in the FintechOS Identity Provider and are visible in Studio as well. Service accounts can be created in the FintechOS IDP interface, as explained on the [FintechOS IDP](#) page, or from Studio, starting with v24.4.

When it comes to security roles, one or more can be associated at user level, enabling a simple process for promoting or revoking rights. You can grant access to the platform to both users within your organization, and to persons outside your organization who use company data to make decisions (external users). Users have access to the platform functionality based on the security role assignment. By default, all users are able to view and manage their account data in the My Account section.

### NOTE

Users with [security roles](#) such as Security Role Management, User Management, and User Admin can manage users.

In FintechOS, user type is a grouping for the users based on platform high level access (to not be confused with system roles).

The platform distinguishes three system user types:

- **Back Office** - The user type that all users have when created. It does not have a special access scope, it is just a category.
- **Guest** - It is just a category, it does not have any special platform access meaning. There can't be two users with this type.
- **Portal** - Used in implementation if there is a DNN Portal created that gets data from FintechOS.

## Add System Users

1. From the menu, click **Security > System Users**. The **Active Users List** page appears.
2. Click **Insert**. The **Add System User** page appears.
3. In the appropriate fields, provide the user credentials the user uses to log into the platform (Username, Password, Confirm Password).
4. From the **Business Unit** drop-down, select the business unit to which the user belongs to.

### NOTE

Root is an important business unit that comes by default with FintechOS. A user configured under the root business unit can see all the records of the entities based on granted access rights.

5. If you want the user to have full access privileges within the platform (Admin user), tick the **Is Administrator** checkbox.
6. Tick the **Is Guest** box if the user you are creating has a guest account.
7. Activate the user by ticking the **Is Authorized** checkbox. Unticking this box, deactivates a user account.

8. Select the **System User Type** by selecting from existing user types or insert new ones based on your needs.
9. Click **Save and reload**. The **Edit System User** page appears.
10. From the **Security Roles Role** section, define the role assignment by clicking **Insert existing** and selecting the desired security role. A user can have multiple role assignments. The user has the access privileges and the level of access as defined by the security items within the selected security roles.
11. Click **Save and close** to save the user updates.

You can now pass the credentials to users and recommend them to change the password at first login.

## Change Password

At first login, system users should change the default password to a different one that also abides by the strong password rules. Passwords must contain a combination of uppercase letters, lowercase letters, numbers, and symbols.

1. In FintechOS Studio, in the top right corner of the screen, click **Profile > My profile > Edit System User**.
2. In the **Edit System User** screen, click the **Change Password** button at the bottom of the screen. A modal window opens where you need to add the current and the new passwords.
3. When done, click **Save and reload**.

## Create Service Accounts

Depending on your infrastructure, you can manage service accounts from the [FintechOS IDP](#), or create accounts in Studio, as explained below. Make sure you are working in a [digital asset](#) of type resources or digital app, and it is set to context, when creating service accounts. When creating a digital package, make sure to tick the **Include Security Configuration** to export this service account and its security roles as well. At import, the client secret is not imported in the destination environment, but rather a new client secret is generated.

1. From the menu, click **Security > System Users**. Click the **Active Service Accounts** button, the **Active Service Accounts** list opens.
2. Click **Insert Service Account**, the **Add Service Account** form opens.
3. Add the **Client Id**. The name of the service account will be in the form `service-account-clientID`, for example, if the client ID is 0123, the username will be `service-account-0123`.
4. Write a description explaining the purpose of this service account.
5. From the **Business Unit** drop-down, select the business unit to which the user belongs to.

6. Click **Save and reload**. The form is reloaded and the **Client Secret** is inserted. To view the secret, click the **Show** button.
7. From the **Security Roles Role** section, define the role assignment by clicking **Insert existing** and selecting the desired security role. A user can have multiple role assignments. The user has the access privileges and the level of access as defined by the security items within the selected security roles.
8. Click **Insert** to add other **Business Units** if necessary.
9. Click **Save and close** to save the service account.

The screenshot shows the 'Service Account Details' form in FintechOS Studio. The form has a dark blue header with the title 'FintechOS Studio' and a sidebar with various icons. The main content area is white and contains the following fields and controls:

- Client Id:** A text input field containing '0123'.
- Client Description:** An empty text input field.
- Service Account Name:** A text input field containing 'service-account-0123'.
- Business Unit:** A dropdown menu showing 'root'.
- Client Secret:** A text input field with masked characters (asterisks). A blue 'Show' button is to its right.
- Security Roles:** A section with a search bar and two buttons: '+ Insert existing' and 'X Remove existing'.

At the top right of the form, there are two blue buttons: 'Save and close' and 'Save and reload'.

## Edit System Users or Service Accounts

You can edit users by changing their details, adding new security roles, editing existing ones, or removing security roles.

### IMPORTANT!

User management, more specifically manual user activation and deactivation, should be done from Studio.

To edit a user, from the **Active Users list**, double-click the active user whose details you want to edit. The **Edit System User** page opens. Make the desired changes and click **Save and close**.

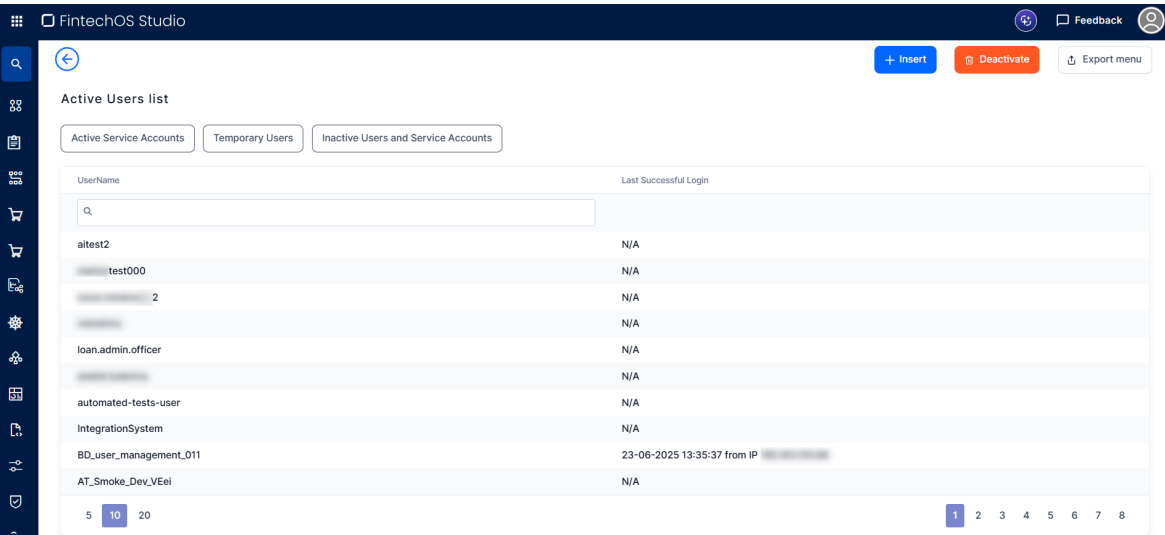
This users view also displays the last successful login date, hour, and IP address.

NOTE

The username field is read-only, you cannot edit it.

Click **Active Service Accounts**, **Temporary Users** or **Inactive Users and Service Accounts** to display lists of these types of users. If you have admin rights, you can delete temporary users by clicking the **Delete Temporary Identity** button.

You can also export lists of users by clicking the **Export menu** button.



## Service Accounts in Digital Assets

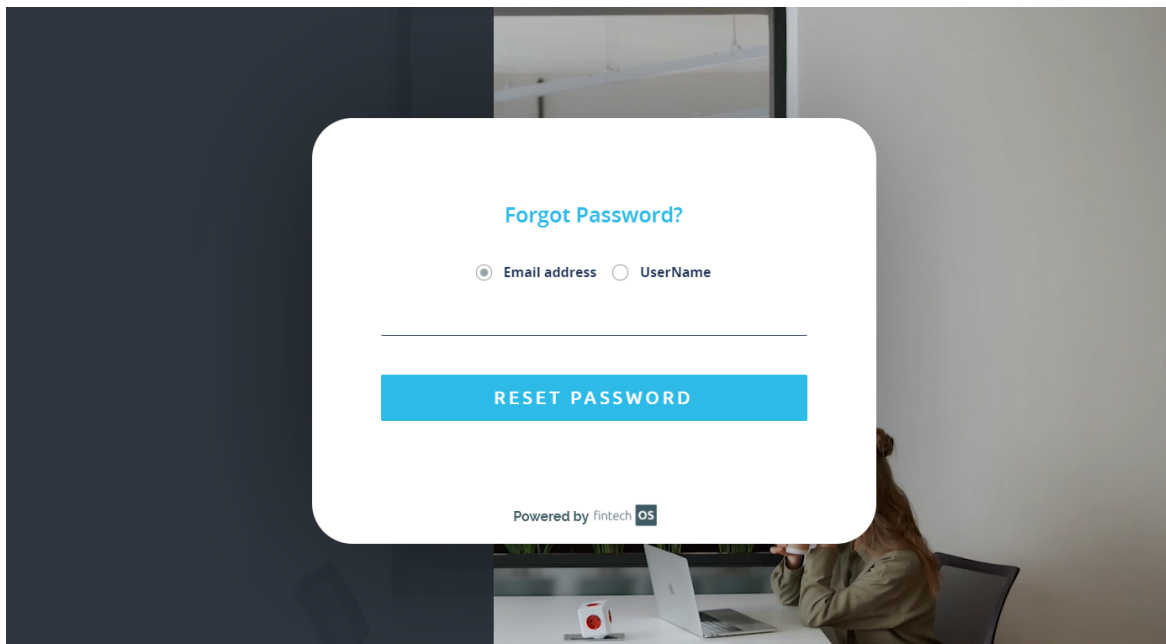
Service accounts can be included in **digital assets** of type resources and digital apps. When adding a configuration item, choose **System User** from the drop-down and pick your service accounts from the list. Security roles linked to the service accounts are also included in the digital asset. When exporting a digital solution package the configuration rules related to security are the same as for all metadata.

# Recover Password (for users)

## NOTE

In order for users to recover their FintechOS Platform password, the forgot password feature should be activated. If you are not able to reset your password, please contact your system administrator or the person who manages you FintechOS Platform app.

To reset your password, on the login page click the [Forgot password?](#) link. You will be prompted to enter your e-mail address or your username. In the Recover password screen, enter the email address or the username associated with your FintechOS Platform account.



Click **Send**.

You will receive an email with instructions on how to change your password.

If multi-factor authentication has been activated, following the instructions received by email, when providing a new password and confirming it, you will have to enter the security pass code received via SMS on your mobile phone.

**NOTE**

The forgot password functionality does not work for LDAP or Azure AD authentication. If you forgot your domain password, contact your system administrator to reset it.

If you haven't received an email for password reset:

- Check your email spam/junk folder.
- Make sure that the email address: [noreply@fintech.com](mailto:noreply@fintech.com) is not blocked or make sure that all emails from this email address are always delivered.
- If the above do not work, try contacting your email service provider. They are most likely blocking emails from FintechOS Platform from being delivered.

## Share Access to Uploaded Files

Some users may need to access files that were uploaded by somebody else, but haven't been associated to an entity record yet.

For instance, in a co-browsing scenario, an operator helps an applicant with an online onboarding. The applicant uploads a file, such as an identity document, but hasn't reached a point in the journey where the entity record is saved in the database (for instance, hasn't reached the next step in the flow). The operator, however, needs to be able to see the file to provide assistance. For this purpose, he needs access to the **FileEntry** entity.

The FileEntry entity, tracks all files as soon as they are uploaded, even before they are assigned to an entity record. Users with access to the FileEntry entity, can access these files under the following conditions:

- Non-B2C users need read access to the FileEntry entity with a security scope of business unit or above. This will grant access to files uploaded by users in the corresponding security scope (the same business unit for the business unit scope, the same business unit or its children for the Parental scope, or the entire organization for

the Organization scope).

- For B2C users, the impersonated user account needs read access to the FileEntry entity with a security scope of Organization.

**IMPORTANT!**

The process described above applies only to files that were uploaded, but haven't been associated to an entity record yet. Once a corresponding entity record is saved, the usual "[Security Roles](#)" on [page 1245](#) based rules apply.

# Admin Configuration

The **Admin** section in FintechOS Studio is a set of administrative configurations that allow you to manage the FintechOS Platform.

This section covers the following topics:

# Email Templates

This feature helps the users build creative emails for campaigns to promote, or remind the clients about a new element of the campaign.

1. Go to **Main Menu > Admin > Email Templates**.
2. Click **Insert** to add a new template.
3. Fill in the Template Name, Subject, and Body. Then click **Save and reload**.

Edit Email Template

Email Template

Template Name

Subject

Body

ResetPasswordEmail

Reset your FintechOS password

Release

Check Out

Current version number: none, CHECKED OUT

History

File - Edit - Insert - View - Format - Table - Tools -

Formats - B / A - [Text formatting icons]

Hello,

No need to worry, you can reset your password by clicking the link below:

[Reset password](#)

Your username is: {userName}.

If you didn't request a password reset, feel free to delete this email.

Thanks,

FintechOS Team

# Entity Cloning and Versioning

**Cloning** allows you to replicate entity records for a variety of operations such as:

- Reducing redundant manual entries when creating similar records.
- Using standard records as templates with pre-filled attribute values.
- Duplicating data for various purposes such as testing, training, or demos.

**Versioning** is useful when you need a thorough validation and tracking of your record updates, such as when storing information about security roles or customer documents. Versions can be used to preserve snapshots of the a record's values at different points in time, aiding in auditing or other regulatory requirements. The versioning process uses ["Business Workflows" on page 418](#) to assign different states to a record indicating if it is:

- a draft that hasn't been approved yet
- the currently active version of the record
- a discontinued prior version
- a substitute version that is a candidate to replace the currently active version.

The record versioning process is illustrated below.

1. The original record starts out in a **Draft** state.
2. Once the draft is validated, the record goes into the **Approved** state, indicating that this it is the currently active version of the record.

## NOTE

Only one version of the record can be active (in the Approved) state at one time.

3. To modify an active record, you need to create a secondary version of it. The secondary version starts out in the **Version Draft** state, while the original record remains in the **Approved** state.
4. When the version draft is ready for approval, the original record transitions into the **Version Closed** state and the secondary version becomes the **Approved** currently active entity record.
5. The process can then be repeated from step 3 whenever a new version of the record is necessary.

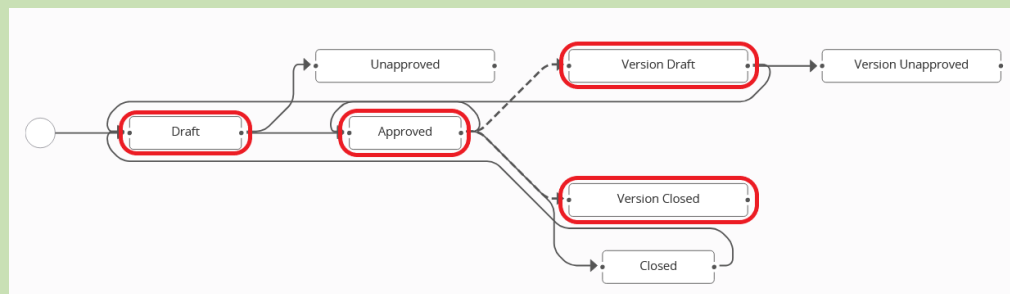
## How to Set Up Cloning and Versioning on an Entity

### 1 Set Up the Entity Workflow

Assign a business workflow to the versioned entity. You may use any workflow that has at least 4 states which you can map to the **Draft**, **Approved**, **Version Draft**, and **Version Closed** statuses with the corresponding transitions.

#### HINT

FTOS\_VersioningWorkflowExample is an example of a preset business workflow that meets the above criteria.



### 2 Configure the Clone and Version Settings

Use the **FTOS\_Versioning\_Setting** system entity to configure sets of clone and version settings that track your versioned entities, statuses, naming convention suffixes, etc. For this purpose, you can expose one of the entity's ["Data Views"](#) on page 112 or

"Data Forms" on page 142 in one of theFintechOS Portal instance's "Dashboards" on page 362 or "Menu Items" on page 371.

Edit FTOS Versioning Setting

Clone and Version Settings

Name

test\_et\_trei

Versioned Entity

test\_et\_trei

Is Versionable

☒

Naming Suffix Keys

s1,s2

Status Draft

s1

Status Approved

s2

Status Version Draft

s3

Status Version Closed

s1

FTOS Versioning Settings Items

+ Insert

× Delete

📄 Export menu

🔄 Refresh

<input type="checkbox"/> Parent Versioned Entity	Related Versioned Entity	Versioning Attribute
<input type="checkbox"/> test_et_parinte	test_et_trei_test_et_parinte	test_et_parinteid

For instance, you can use the FTOS\_Versioning\_Setting entity's default form to populate the following fields:

- **Name** - Assign a name to the set of clone and version settings.
- **Versioned Entity** - Name of the entity you are making available for cloning or versioning.
- **Is Versionable** - Toggles if the entity is available for versioning or only for cloning.

TABLE OF CONTENTS

1263

- **Naming Suffix Keys** - Comma separated list of placeholders which will be replaced with suffixes to be appended to the cloned/versioned records' primary attributes.
- **Status Draft** - State in the entity's business workflow you wish to map to the Draft status. Mandatory if the entity is versionable.
- **Status Approved** - State in the entity's business workflow you wish to map to the Approved status. Mandatory if the entity is versionable.
- **Status Version Draft** - State in the entity's business workflow you wish to map to the Version Draft status. Mandatory if the entity is versionable.
- **Status Version Closed** - State in the entity's business workflow you wish to map to the Version Closed status. Mandatory if the entity is versionable.

#### (Optional) Set Up Related Versioned Entities

When you clone/version an entity record, you may also want to automatically clone/version some of its related entity records. These records belong to entities that have lookup attributes pointing to the versioned entity either directly or indirectly (through a stream of intermediate related entities).

This allows you to propagate your cloning/versioning to any relevant dependencies. For instance, you may have an insurance product record with a number of related insured items records, each of which has multiple related covered risks records. When you create a new version of an insurance product, you will also be able to create new versions of all its insured items and all of their covered risks at the same time.

For this purpose, scroll down to the FTOS Versioning Settings Items section and add the desired dependencies:

- **Parent Versioned Entity** - If the versioning attribute points to the main versioned entity (is linked directly), leave empty. Otherwise (if it is linked indirectly), select the entity the versioning attribute points to.
- **Related Version Entity** - Name of the related versioned entity.

- **Versioning Attribute** - Lookup attribute pointing to the main versioned entity (either directly or indirectly through a stream of intermediate related entities).
- **Disable Version or Clone** - Allows you to temporarily disable cloning and versioning for the related entity.
- **Apply Naming Convention** - Propagates the naming suffix keys to the related entity.

[Edit FTOS Versioning Setting Item](#)

#### FTOS Versioning Setting Item

Parent Versioned Entity	<input type="text" value="test_et_parinte"/>
Related Versioned Entity	<input type="text" value="test_et_trei_test_et_parinte"/>
Versioning Attribute	<input type="text" value="test_et_parinteid"/>
Disable Version or Clone	<input type="checkbox"/>
Apply Naming Convention	<input type="checkbox"/>

## How to Clone an Entity Record

Use the [ftos.data.clone](#) Server SDK method to create a clone and retrieve its record ID.

## How to Version an Entity Record

To clone or version an entity record:

1. Use the [ftos.data.version](#) Server SDK method to create a draft version of the desired entity record.

2. Make your edits on the draft version.
3. Use the [ftos.data.closeLastApprovedVersion](#) Server SDK method to close the approved record version and redirect its references to the version draft.
4. Approve the draft using the corresponding "[Business Workflows](#)" on page 418 transition.

For example:

```
var myEntity = ftos.context.entityName;
var currentRecord = ftos.context.id;

// Create a new draft version of the current record based on the
// version_config clone and versioning settings
var myDraft = ftos.data.version(currentRecord, version_config);

// Redact the first name and last name in the draft
ftos.data.update(myEntity, myDraft, { FirstName: 'Redacted',
LastName: 'Redacted' });

// Close the previously approved version and redirect references to
// the draft
ftos.data.closeLastApprovedVersion(myDraft, version_config);

// Approve the draft
ftos.businessStatusWorkflow.update(myEntity, myDraft,
ftos.businessStatusWorkflow.getId(myEntity, 'Approved'));
```

## Legacy Entity Versioning (Deprecated)

Starting with v24.3, the entity versioning mechanism has been updated based on the model described above. For information on the legacy entity versioning mechanism, see the [v24.2 documentation](#).

# Settings

Settings allow you to customize the FintechOS Portal look and feel. From the menu, click **Admin > Settings**.

1 General

2 Custom Theme

Portal

Portal Company Logo

Add file or Drop file here

Portal background image

Add file or Drop file here

☒ Use floating style for Portal

☐ Use full width forms

FintechOS Studio

☐ Use the floating style for FintechOS Studio

Default Form Template Type

1 Column

☐ Use Custom Home Page

☒ Show Shortcuts Tab on Home page

☒ Use Global Dashboard

☐ Use Custom Styles

☐ Use deployment package permissions

Upload your own company logo and background image to be shown in the **Digital Experience Portal**.

To show the Portal UI container in full screen width, tick the **Use full width forms** checkbox.

You can also add custom actions (custom flows) on the home page by ticking the **Use Custom Home Page** checkbox and selecting the desired custom flow. A new custom flow can be inserted directly in the **Settings** page, by clicking the arrow next to the **Use Custom Home Page** field. The list of available custom flows will be displayed. Click the **Insert** icon and provide the custom flow information. For more information on custom flows, see [Creating Form Driven Journeys](#).

Choose if the shortcuts are displayed as the first tab on the homepage by selecting the checkbox next to the **Show Shortcuts Tabs on Home page** label. Unselect the checkbox to disable the user shortcuts and not have them displayed as the first tab on the home page, but available through a new icon on the top menu bar.

Multiple dashboards can be added to the Portal UI if the **Show on Home page** option is selected on the dashboards configuration page.

To force the default Dashboard, named Main Dashboard, to be displayed first on the user interface, tick the **Use Global Dashboard** checkbox.

By default, the color palette of the Digital Experience Portal UI is inherited from the FintechOS Platform brand. To apply your own brand, follow these steps:

1. Select the **Portal background image**.
2. Set the transparent top menu bar and a floating feeling for all elements by ticking the **Use floating style for Portal** checkbox.

**NOTE** The global dashboard has higher priority than the floating dashboard; therefore, if both checkboxes are selected, only the global dashboard will be visible in the user interface.

## Custom Themes

Custom themes allow you to adjust the default fonts and color palette of your FintechOS Portal user interface. For more info, see ["Custom Themes" on page 373](#).

## Localization

Localization allows you to adapt a product to the language and culture of a target audience. It comprises the following elements:

- Translation
- Design and layout to properly display translated text
- Converting to local requirements (such as currencies and units of measure)
- Using proper local formats for dates, addresses, and phone numbers
- Addressing local regulations and legal requirements

FintechOS Platform supports the localization of static and dynamic elements, as well as the localization of customized messages and metadata.

### NOTE

Before localizing in a new language, you need to prepare your environment to easily identify the resources to be localized resources. For information on how to set up the localization, see the [FintechOS Platform Administration Guide](#).

FintechOS Platform supports the localization of static and dynamic elements, as well as the localization of customized messages and metadata.

### NOTE

Currently, application languages and their linked items cannot be deployed via Digital Assets.

## Available Languages

To view the list of available languages, from the menu, click **Admin > Application Languages**.

<input type="checkbox"/>	Name	Culture	Currency	Date Format	Default Language	ISO Code	Disabled
<input type="checkbox"/>	English	en-GB	£/\$	dd/MM/yyyy	<input checked="" type="checkbox"/>	626	<input type="checkbox"/>
<input type="checkbox"/>	Romanian	ro-RO	RON	dd.MM.yyyy	<input type="checkbox"/>	642	<input type="checkbox"/>

By default, the page lists the standard system languages: English (EN) and Romanian (RO). To use any of them, on the right-side of the navigation bar, next to your name, click the language code icon, and select the language that you want to use.



### SELECT LANGUAGE

 English

 Romanian

Save

The page refreshes and the user interface text is displayed in the selected language.

#### NOTE

The **applicationLanguage** entity stores information and configuration related to the supported languages.

## Add Languages

**NOTE** When adding a new language, the system automatically creates all the additional database fields for the new language, for all the attributes marked as localizable.

To add a new language, follow these steps:

1. From the **Main Menu**, click **Admin > Application Languages**.
2. In the **Application Language List** page, click **Insert**, and enter the language details:

Field	Description
Name	The name of the language you want to add. The field is mandatory.
Culture	The .NET language culture. Make sure that you enter one from the available <a href="#">.NET cultures</a> . The field is mandatory. By default, two cultures are available: en-GB and ro-RO.
ISO Code	<p>The ISO country code. View the list with <a href="#">ISO country codes</a>.</p> <div> <p><b>NOTE</b> You must enter the ISO Numeric Code UN M49 Numerical Code, otherwise you will receive an error message when saving the new language.</p> </div> <p>Based on the country ISO Code, the corresponding flag and language name appear in the user settings.</p>
Currency	The default currency for the specified language.
Default language	The user default language after login.

Field	Description																
Date Format	<p>The format of attributes of type Date Time. You can define a different format per language:</p> <table> <tr> <th>Date Attribute Format</th><th>Example Output</th></tr> <tr> <td>dd/MM/yyyy</td><td>05/10/2022</td></tr> <tr> <td>dd/MMM/yyyy</td><td>05/Oct/2022</td></tr> <tr> <td>dd/MMMM/yyyy</td><td>05/October/2022</td></tr> <tr> <td>d/MMMM/yy</td><td>5/October/22</td></tr> <tr> <td>MMM dd, yyyy</td><td>Oct 05, 2022</td></tr> </table>	Date Attribute Format	Example Output	dd/MM/yyyy	05/10/2022	dd/MMM/yyyy	05/Oct/2022	dd/MMMM/yyyy	05/October/2022	d/MMMM/yy	5/October/22	MMM dd, yyyy	Oct 05, 2022				
Date Attribute Format	Example Output																
dd/MM/yyyy	05/10/2022																
dd/MMM/yyyy	05/Oct/2022																
dd/MMMM/yyyy	05/October/2022																
d/MMMM/yy	5/October/22																
MMM dd, yyyy	Oct 05, 2022																
Date Time Format	<p>The format of attributes of type DateTime. You can define a different format per language:</p> <table> <tr> <th>DateTime Attribute Format</th><th>Example Output</th></tr> <tr> <td>dd/MM/yyyy HH:mm</td><td>05/10/2022 15:25</td></tr> <tr> <td>dd/MM/yyyy hh:mm</td><td>05/10/2022 03:25</td></tr> <tr> <td>dd/MM/yyyy hh:mm tt</td><td>05/10/2022 03:25 PM</td></tr> <tr> <td>dd/MMM/yyyy h:mm</td><td>05/Oct/2022 3:25</td></tr> <tr> <td>dd/MMMM/yyyy HH:mm</td><td>05/October/2022 15:25</td></tr> <tr> <td>d/MMMM/yy HH:mm</td><td>5/October/22 15:25</td></tr> <tr> <td>MMM dd, yyyy - HH:mm:ss</td><td>Oct 05, 2022 - 15:25:00</td></tr> </table>	DateTime Attribute Format	Example Output	dd/MM/yyyy HH:mm	05/10/2022 15:25	dd/MM/yyyy hh:mm	05/10/2022 03:25	dd/MM/yyyy hh:mm tt	05/10/2022 03:25 PM	dd/MMM/yyyy h:mm	05/Oct/2022 3:25	dd/MMMM/yyyy HH:mm	05/October/2022 15:25	d/MMMM/yy HH:mm	5/October/22 15:25	MMM dd, yyyy - HH:mm:ss	Oct 05, 2022 - 15:25:00
DateTime Attribute Format	Example Output																
dd/MM/yyyy HH:mm	05/10/2022 15:25																
dd/MM/yyyy hh:mm	05/10/2022 03:25																
dd/MM/yyyy hh:mm tt	05/10/2022 03:25 PM																
dd/MMM/yyyy h:mm	05/Oct/2022 3:25																
dd/MMMM/yyyy HH:mm	05/October/2022 15:25																
d/MMMM/yy HH:mm	5/October/22 15:25																
MMM dd, yyyy - HH:mm:ss	Oct 05, 2022 - 15:25:00																
Disabled	True, then the language is disabled and will not appear in the list of available languages in the UI.																

Edit Application Language

Application Language

Name

Romanian

Culture

ro-RO

ISO Code

642

Currency

RON

Date Format

dd.MM.yyyy

Date Time Format

dd.MM.yyyy HH:mm

Disabled

☒

Default Language

☐

3. Click **Save and close** to save the language. If you want to save the new language and stay on the Add Application Language page, click **Save and reload**.

**NOTE**

To have your app localized in a language other than default languages, you should localize all the application resources. For information on how to localize generic resources, see "[Localization Resources](#)" below.

## Localization Resources

By default, FintechOS Studio includes ro-RO and en-EN localizations for the generic UI elements, such as fields (password, username, etc.), buttons (Save and reload, OK, Delete), and others.

This section describes how to add a new localization resource, view the list of generic language resources, and how to localize them.

### Localize Language Resources

To view the list of UI generic resources, go to **Main Menu > Admin > Localization Resources**. The list displays all UI localization resources sorted automatically by module, key and culture.

Localization Resources list

Module Name	Resource Key	English Value	Romanian Value	Culture Name	Value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Action_Login	Login	Conectare	en-GB	Login
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Action_Login	Login	Conectare	ro-RO	Conectare
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Password	Password	Parola	en-GB	Password
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Password	Password	Parola	ro-RO	Parola
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Title	Login	Autentificare	en-GB	Login
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_Title	Login	Autentificare	ro-RO	Autentificare
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_UserName	UserName	Utilizator	en-GB	UserName
EBS.Core.Web.Mvc.LocalizationResources...	Account_Login_UserName	UserName	Utilizator	ro-RO	Utilizator
EbsCore.JavaScript.Static	ActionHandler_Advanced_Find	Advanced find	Cautare avansata	en-GB	Advanced find
EbsCore.JavaScript.Static	ActionHandler_Advanced_Find	Advanced find	Cautare avansata	ro-RO	Cautare avansata

51020

1234...

The **Value** field supports inline editing, so you do not have to open each record to edit it. To easily localize the UI resources, click the **Value** field corresponding to the resource to be localized and type in the localized text.

## Export-Import Localization Resources between FintechOS Studio Instances

To exchange localizations between FintechOS Studio instances, you need to export the localization resources from the source instance and import them in the target instance:

1. On the source FintechOS Studio instance, go to **Main Menu**, click **Admin > Localization Resources**.
2. Click **Export** and select **Export Data set**. All localizable resources within the list are exported to an Excel file. For details, see [Data Exports](#).
3. On the destination FintechOS Studio instance, use the "[Data Import Templates](#)" on [page 165](#) feature to set up an import template for the **LocalizationResource** entity.
4. Use the template to import the Excel data set file.

## Localize Metadata

For data localization, some of the most visible metadata entities have an additional **DisplayName** field, that is used when rendering the user interface instead of the **Name** field.

FintechOS Studio comes with two predefined languages: en-GB and ro-RO; where en-GB is the default language.

The following attributes are marked localizable by default:

entity	displayName
entity	displayCollectionName
attribute	displayName
action	displayName
actiongroup	displayName
optionset	displayName

entity	displayName
optionsetitem	displayName
entityformheaderitem	label
entityformsection	displayName
entityviewcolumn	label
customAction	displayName
relationship	displayName
report	displayName

At metadata attribute level, there is a new property **IsLocalizable**. If the checkbox is ticked on Text attributes, the system automatically creates additional fields in the database for each application language except for the default one (that is, en-GB).

#### NOTE

The primary attributes are not localizable.

To localize metadata, follow these steps:

1. Provide the value for the display name in en-GB.
2. Switch to the language in which you want to localize resources. If the Debug mode has been activated, the value to be localized is marked with the question mark.
3. Replace the en-GB value displayed in the **Display Name** field with the one corresponding to the language you want to localize.
4. Click **Save and close**.

The localization updates are saved in the database for the field corresponding to the current language selected from the **User Settings**. Inserts will save the data in all additional fields for each localizable attribute.

## Localize HTML Templates

FintechOS Platform allows you to localize the content of HTML template embedded within the entity data forms, data form sections and custom actions.

**NOTE**

- Localization applies to the text part of the HTML element and it does not support children mixed with text. When mixing text with other HTML elements, split the text in spans.
- The metadata displayed on forms is localized using the DisplayName, you do not have to insert resources for them. If you want to translate the same metadata with different text, depending on the context, on different forms with different rules, then add the localization resources separately from the forms.

## Localize HTML elements on data forms

To localize HTML elements that will be displayed on data forms, on the HTML Editor, click **Tools > Source code**, and add the attributes data-resource-key and data-culture to the element.

The data-culture attribute should be a valid .NET culture name. For details on the valid .NET culture names, see: [MSDN Table of Language Culture Names, Codes, and ISO Values Method \[C++\]](#).

When not specified, the data-culture attribute is set by default to English UK (en-GB).

You can add additional translations for the same label by using the custom 'ebs-resource' tag.

**NOTE** The HTML editor does not allow the ebs-resource tag inside tr and td elements, so you need to add it to the div level.

```
<div>
```

```

<!-- localization for myAttr specific for italian language,
augments the localization inside td tag -->
<ebs-resource data-resource-key="myAttr" data-culture="it-
IT">TestCinci-IT</ebs-resource>
<table>
<tr>
<!-- localization by resource key -->
<td data-resource-key="myAttr" data-culture="ro-
RO">Attributul meu</td>
<td>{MyOptionSetAttribute}</td>
</tr>
</table>
</div>

```

On save, the resource keys are automatically inserted within the LocalizationResource and available for edit in the Localization Resources editor. The resource keys will be saved with the module name following these naming conventions:

## Entity data form template naming convention

*entities / entiyName / forms / formName / html*

Example: *entities/ProductPromotion/forms/default/html*

## Form section template naming convention

*entities / entiyName / forms / formName / sections / sectionName / html*

Example: *entities/ProductPromotion/forms/default/sections/Section 2/html*

## Custom action naming convention

*customActions / customActionName / html*

Example: *customActions/Product Promotion/html*

## Localize from Metadata

To get localization from metadata, use piped arguments inside the bracket expressions:

- `{ xxx | entity }` will be replaced with the display name of the entity named xxx. The entity name is case sensitive.
- `{ xxx | entities }` will be replaced with the display collection name of the entity named xxx. The attribute name is case sensitive.
- `{ myAttribute | attribute }` will be replaced with the display name of the attribute named myAttribute. The entity is the data form owner. The attribute name is case insensitive.
- `{ xxx_yyy | relationship }` will be replaced with the display name of the relationship between entity xxx and entity yyy. The relationship name is case insensitive.

### HTML Template Localization

```
<div style="padding: 20px; background-color: white; border:
solid 1px #E0E0E0;">
  <table style="width: 100%;">
    <tbody>
      <tr>
        <td colspan="2">
          <!-- entity pipe: display name of entity
test1 (case sensitive) -->
          <h4>{test1 | entity }</h4>
          <!-- entities pipe: display collection
name of entity test1 (case sensitive) -->
          <h4>{test1 | entities }</h4>
        </td>
      </tr>
      <tr>
        <!-- localization by resource key -->
        <td data-resource-key="myAttr" data-
culture="ro-RO">Attributul meu</td>
        <td>{MyOptionSetAttribute}</td>
      </tr>
```

```

        <tr>
            <!-- localization by resource key -->
            <td data-resource-key="name" data-
culture="ro-RO">label pentru name</td>
            <td>{Name}</td>
        </tr>
        <tr>
            <td>{Name | attribute }</td>
            <td>{Name}</td>
        </tr>
        <tr>
            <!-- attribute pipe: shows display name of
attribute Field1 (case insensitive)-->
            <td>{Field1 | attribute }</td>
            <td>{Field1}</td>
        </tr>
        <tr>
            <td colspan="2">
                <hr />
            </td>
        </tr>
    </tbody>
</table>
</div>

```

## Localize Relationship Labels

To render relationships inside the HTML template, use a syntax similar to:

```
{#sys_entity_sys_attribute,collapse:Attributes#}
```

Where:

- `sys_entity_sys_attribute` is the name of the relationship.
- `Attributes` is a user-defined label to be rendered as title.

To automatically localize this construct, use the localizable `displayName` attribute from the Relationship by specifying `$displayName` as the label.

```
{#sys_entity_sys_attribute,collapse:$displayName#}
```

## Localize Option Set Items

The `dxSelectBox.option` function is used in customization code to access the name of the option set directly from the control, so localization of `optionSetItem` would return the localized `DisplayName` instead of the expected `Name`.

The rendering for the select box has been modified so that it uses the **DisplayName** attribute of the option set item as display name instead of the **Name** attribute.

The below will return the option set item name, and not the translated text:

```
$("#ebsContainerContent_xxx_list").dxSelectBox("instance").option("text")
$("#ebsContainerContent_xxx_list").dxSelectBox("instance").option("displayValue")
```

## Localize Views

To localize view columns, follow these steps:

1. On the right-side of the navigation bar, next to your name, click the language code icon and select English). The page refreshes and the UI texts are displayed in English.
2. On the desired entity (Edit Entity page), scroll-down to the **Data Views** section and click it. The Data Views section expands.
3. Double-click the desired view. The View configuration page appears.
4. Click the **Data** tab, scroll-down to the **Entity View Columns** section and click **Insert**.
5. Fill-in the following fields: Attribute Name and Label.
6. Go to **User Settings** again and switch the language (e.g., fr-FR). The page refreshes and the Label field is marked with the question mark (?) indicating that it has not been localized yet.

7. In the **Label** field, enter the localized value then at the top-right corner of the page click one of the save icons (based on your needs).

#### NOTE

- Localization is backward compatible with previous implementations, in which view columns have been added in the Data field, instead of the View columns grid.
- Once localized, the resource will be displayed in the language selected, regardless of the display option used within the view.

## Client-side and Server-Side Localization

### Client-side

You can specify any "design-time" language specific messages. The resources are automatically exported when the metadata is saved and are available for edit in other defined languages in the Localization Resources editor.

It is not mandatory to define message values for all languages using JavaScript. You should define at least one language (en-GB or ro-RO) using JavaScript. For other languages, go to the Localization Resource entity and fill-in the **Value** field corresponding to the resources to localize.

At run-time, the application will resolve the translations from the database with fallback to the explicit values defined in the script.

#### Code Snippet

```
var rsMyMessage = new EbsResource({
  key : "myMessage",
  "en-GB" : "message in English",
  "ro-RO" : "mesaj in romana",
  "de-DE" : "Meldung auf Deutsch"
});
```

```
var rsMyMessageFmt = new EbsResource({
    key : "myMessageFmt",
    "en-GB" : "Field {0} is empty",
    "ro-RO" : "Campul {0} nu este completat",
    "de-DE" : "Der Feld {0} ist leer"
});
console.log( rsMyMessage.getString()); // when culture is ro-RO
outputs: mesaj in romana
console.log( rsMyMessageFmt.getString("Field1")); // when culture
is ro-RO outputs: Campul Field1 este necompletat
```

### ebs.showMessage

The existing method for ebs.showMessage has been modified to accommodate localization.

You can verify the code snippet either in the 'Developer Tools' (in Chrome) or in the **After Events** field.

Code snippet from Developer Tools:

```
> ebs.showMessage
< f (englishMessage, type, romanianMessage) {
    //type:success, warning, info, error
    //englishMessage can be a resource identifier

    var Localized = null;
    if (englishMessages && englis...
```

Code snippet for After generate events:

```
ebs.showMessage(englishMessage, type, romanianMessage);
```

**NOTE** New localization is backwards compatible, it passes both English and Romanian message strings; however, when using EbsResource it is not mandatory that you pass the Romanian message.

Code snippet from Developer Tools:

```
> EbsResource
< f EbsResource(data) {
    if (data) {
```

```

        for (var prop in data) {
            this[prop] = data[prop];
        }
    }
}

```

ebs.showMessage

```

var rsMyWarning = new EbsResource({
    key : "myWarning",
    "en-GB" : "Warning!",
    "ro-RO" : "Atentie!",
});

// other code
ebs.showMessage(rsMyWarning.getString(), 'warning'); //new style
ebs.showMessage("english message", 'warning', "mesaj in romana"); // still works for backwards compatibility

```

When saved, the resource keys are automatically inserted in the database and are available for edit in the **Localization Resource** entity. The resource keys will be saved with the module name following these naming conventions:

## Entity data form aftergeneratejs naming convention

*aftergeneratejs: entities / entiyName / forms / formName / aftergeneratejs*

Example:

*entities/Product/forms/default/aftergeneratejs*

## Form section aftergeneratejs naming convention

*entities / entiyName / forms / formName / sections / sectionName / aftergeneratejs*

Example:

*entities/Product/forms/default/sections/Section 1/aftergeneratejs*

## Entity views aftergeneratejs naming convention

*entities/ entiyName/ views/ viewName / aftergeneratejs*

Example:

`entities/Product/views/default/aftergeneratejs`

## Form attribute change aftergeneratejs naming convention

*entities / entityName / forms / formName. / attributes / attributeName / attributeChangeEventsJs*

Example:

*entities/Product/forms/default/attributes/Name/attributeChangeEventJs*

## Custom actions aftergeneratejs naming convention

*customActions / actionName / aftergeneratejs*

Example:

*customActions/Product Promotion/aftergeneratejs*

## Server-side

JavaScript localization support is available in workflows and workflow libraries. The pattern is similar to the one used in Form AftergenerateJS, with a small difference at call time:

```
var resource1 = new EbsResource({
    key : "myKey",
    "ro-RO" : "mesaj romana",
    "en-GB" : "English message"
});

throwException( getString(resource1), 1 ); //the resource must be
passed as parameter to getString

//support for message formatting
```

```

var resource2 = new EbsResource({
    key : "myKey",
    "ro-RO" : "Atributul {0} din entitea {1} nu poate fi null",
    "en-GB" : "Attribute {0} in entity {1} cannot be null"
});

// getString can be called as string.Format in .NET, passing the
// formatting args as an array argument

throwException( getString(resource2,
    ["MyManadatoryAttribute", "MyEntity"]) , 1 );

```

Upon save, the resource keys are automatically inserted within the LocalizationResource and available for edit in the Localization Resources editor. The resource keys will be saved with the module name following these naming conventions:

## Workflow naming convention

*workflows / workflowname / js*

Example:

*workflows/Product wf/js*

## Workflow library naming convention

*workflowlibraries / workflowlibname / js*

Example:

*workflowLibraries/Product wf library/js*

# System Parameters

System parameters are used to apply default settings or requirements that are treated as rules in your environment. For example, you can set a parameter to show/hide the company logo, or enable/disable the message shown after an insert/update/delete record.

To add a new system parameter:

1. Open the main menu, and go to **Admin > System Parameters**.
2. Click **Insert** and fill in the following fields:
  - **Name** - Insert a name for the parameter.
  - **Parameter value** - Insert the configuration needed e.g. 12:00; false; 3.
  - **Description** - Insert an appropriate description.
3. Click **Save and reload**.

From here, if you want to add your system parameter to a [portal profile](#):

1. Scroll down to **System Parameters On Portal Profiles** and select **Insert**.
2. Choose the desired Portal Profile from the drop-down list.
3. Provide the value in the **Provide Value** field.
4. Select one of the **Save** options.

#### NOTE

Changes made to system parameters are cached and require at least 30 seconds to propagate and take effect.

If you have scenarios requiring real-time information and not cache based, you can create [an entity](#) and read the attribute's value directly from the database.

## Default System Parameters

Name	Description	Default Value
sys-hide-footer	Set it to true (1) to hide the page footer. It only has effect on the Portal.	0
sys-hide-form-title	Set this parameter to true (or 1) to hide all form titles.	0

Name	Description	Default Value
sys-feeds-heartbeat-interval	Set this parameter in order to activate feeds. Its value should be an integer representing the interval (in seconds) on how often to execute automatic count. The minimum value for this parameter is, by default, 30.	
sys-hide-user-settings	Set this parameter to true (or 1) to hide User Settings (top-right corner, <b>Select Theme</b> and <b>Select Palette</b> ). This parameter is ignored in Designer/Studio.	0
sys-remove-exif-metadata	Set this parameter to true (or 1) to delete the metadata related to security and privacy concerns from uploaded files (e.g. EXIF geolocation metadata from images). This parameter is ignored in Designer/Studio.	0
sys-hide-shortcuts-navbar-left	This parameter hides only the shortcut links from left navbar. Set it to true (1) to hide the shortcut links from left navbar. It only has effect on the Portal.	0
sys-disable-logging-request-params-for-urls	Set this parameter in order to prevent logging requests parameters (containing sensitive information) for specified URLs. Each URL must be placed on a separate line. E.g.: api/Authorize/GetToken. If the parameter's value is empty, the requests' parameters will be logged.	
sys-disable-record-insert-update-delete-message	Set this parameter to true (or 1) to disable the message shown after insert/update/delete record. This parameter is ignored in Designer/Studio.	

Name	Description	Default Value
sys-documentreport-should-read-barcode-from-userfiles-storage	If this parameter's value is false (0), the {barCode128} token will be resolved by getting the image from the disk using the specified absolute/relative path. If its value is true (1), the {barCode128} token will be resolved by getting the image from the configured user files storage using the specified file name.	0
sys-allow-multiple-BU-per-user	This parameter enables the selection of a Business Unit that the user is assigned to.	0
sys-hide-my-profile-link	Set this parameter to true (or 1) to hide the MyProfile link (top-right corner, under User Profile panel). This parameter is ignored in Designer/Studio.	0
sys-hide-app-dashboard	Set this parameter to true (or 1) to hide the application dashboard and also the TV shown when shortcut links are deactivated in the settings. This parameter is ignored in Designer/Studio.	0
sys-minimal-css	Set this parameter to true (or 1) to force the Portal to load only the necessary style sheets required for the application to run normally. This parameter is ignored in Studio. Generate color palette from the background will not be triggered, and Choose theme and Choose palette will not be available.	0
sys-hide-search-in-menu	Set this parameter to true (or 1) to hide the search in the application menu. This parameter is ignored in Designer/Studio.	

Name	Description	Default Value
sys-right-footer-text	Set this parameter in order to display a custom text in the right footer. Its value must be set as a JSON object with a property for each supported language. E.g. {'en-GB':'Custom text', 'ro-RO':'Text personalizat'}. If the value is empty, the default text "ADVANCED DIGITAL LAUNCHPAD" is displayed.	
sys-minimal-login-css	Set this parameter to true (or 1) to force the Portal to load only the necessary style sheets required for the login page to run normally. This parameter is ignored in Designer/Studio.	0
sys-hide-company-logo	Set this parameter to true (or 1) to hide the company logo. This parameter is ignored in Designer/Studio.	
sys-hide-menu	Set this parameter to true (or 1) to hide the application menu. This parameter is ignored in Designer/Studio.	
sys-compatibility-url-params-router	This parameter allows the new routes (no-path params) to function in compatibility mode with the old routes, created by API functions: e.g. <code>ebs.goToUrl</code> , <code>ebs.getNavigation</code> , and parse the links to route state and respectively, call <code>EbsRouter</code> in <code>ebs.goToUrl()</code> . It also blocks legacy routes from being parsed to a route state, or work in compatibility mode and treat the link as an error. This parameter is evaluated only in Studio, and only if <code>sys-single-page-app-router</code> is true.	1

Name	Description	Default Value
sys-load-custom-style-sheet	This parameter always loads the provided style sheets. Its value must be an enumeration of names of Style Sheet entity records (comma separated). It is available in B2C mode too, and it is not available in the login page.	
sys-log-anonymization-settings	The parameter is used for <a href="#">log anonymization</a> . The value represents the properties to be anonymized when a JSON object or array is logged using the log SDK method. The value is used only when <code>feature-log-anonymization-enabled</code> is enabled.	
sys-hide-migration	Set this parameter to true (or 1) to hide the migration module from the application.	0
sys-ftos-cmb-activity-heartbeat-interval	Set this parameter to complement the <code>show-ftos-cmb-activity-shortcut</code> field in order to refresh the list dynamically and regularly. Its value should be an integer representing the interval (in seconds) on how often to execute an automatic count. The minimum value for this parameter is, by default, 30.	
sys-document-title	Set this parameter in order to display a custom text as the webpage title. Its value must be set as a JSON object with a property for each supported language. E.g.: <code>{ 'en-GB': 'Custom text', 'ro-RO': 'Text personalizat' }</code> . If the value is empty, the default text "FintechOS", "DigitalOS" or "DCX360" is displayed.	

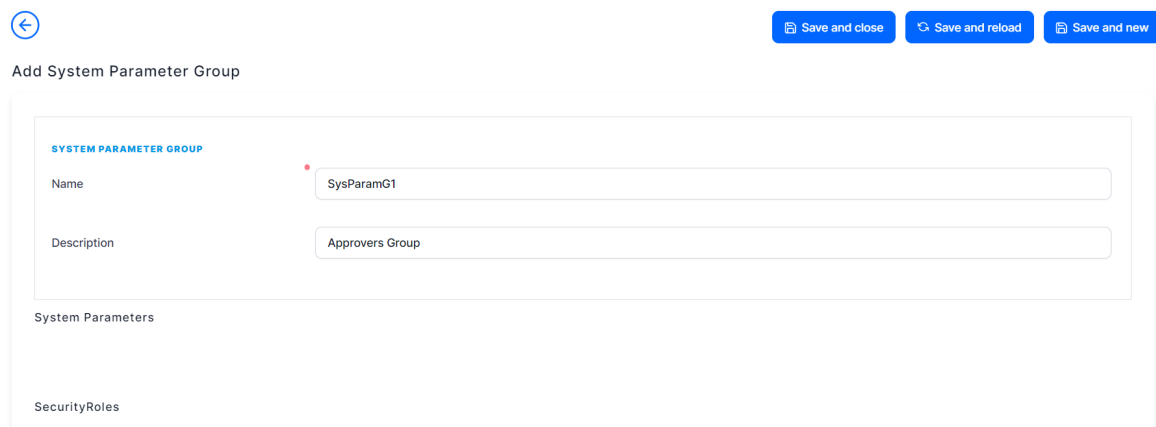
Name	Description	Default Value
sys-hide-header-navbar-default	This parameter hides the page header default navigation bar that contains components like: logo, menu, feed and approvalTask shortcuts, account dropdown. Set it to true (1) to hide the page header default navigation bar. It only has effect on the Portal.	0
sys-deny-altered-content-import	Set this parameter to true (or 1) to deny import of files that have been modified after the export generated by the application.	0
sys-single-page-app-router	This parameter enables SPA EbsRouter which uses navigation paths that do not expose information in the URL. Set it to false (0) to enable legacy navigation using links with complete information.	1
sys-redirect-on-session-expiry	Set this parameter to true (or 1) to force the portal to log-off and redirect to login when the session token expires. This parameter is ignored in Studio.	0
sys-hide-dashboard-tabs	This parameter hides the page dashboard-tabs. Set it to true (1) to hide them. It only has effect on the Portal.	0
sys-move-language-to-user-profile	Set this parameter to true (or 1) to display the selected language and language selection in the User Profile panel. This parameter is ignored in Studio.	0
sys-approval-tasks-heartbeat-interval	Set this parameter to show the active tasks icon in the top-right corner, and to enable the automatic count of active tasks. Its value should be an integer representing the interval (in seconds) on how often to execute an automatic count of active tasks. The minimum value for this parameter is, by default, 30.	

Name	Description	Default Value
sys-show-ftos-cmb-activity-shortcut	Set this parameter in order to (de)activate the activities menu. The entity (Activity) is installed from the Standard SysPacks. All the activities with BusinessStatus different from Done, InProgress and Cancelled match the criteria to be displayed in the menu. Its value should be true or false.	
sys-show-feed-shortcut	Set this parameter in order to show the mail-icon on the top right of the screen for the feeds notifications. Its value should be true or false.	
sys-enable-pdf-export-tooltip	This parameter controls the visibility of the tooltip that will show when PDF export is missing - this occurs when a PDF template is not configured.	0
sys-enable-in-app-experiences	Setting this flag to true will enable contextual help, flows, embed videos and more to support the users navigating the FintechOS Studio.	0
sys-workflowengine-useclearscript	This parameter determines if the workflow engine is using Jint or Microsoft.ClearScript library. Set it to true (1) to use Microsoft.ClearScript.V8.V8ScriptEngine instead of Jint.Engine".	0
sys-workflowengine-enable-debugging	This parameter determines if the debug mode is enabled when calling actions using Microsoft.ClearScript.V8.V8ScriptEngine. Set it to true (1) to enable debug mode.	0
sys-workflowengine-type	This parameter specifies the type of JavaScript engine used by the workflow. Possible values are Jint, V8, JintV3.	Jint
sys-do-not-allow-client-side-direct-data-updates	Disables the <code>ebs.insertEbs</code> , <code>ebs.insertAsync</code> , <code>ebs.updateEbs</code> , <code>ebs.updateAsync</code> , <code>ebs.deleteEbs</code> , and <code>ebs.deleteAsync</code> Client SDK functions.	1

## System Parameter Groups

In FintechOS Studio, the **System Parameter Groups** menu item allows you to set up a Security Role and add several System Parameters to a group. This adds an extra layer of security, in that only users having that specific Security Role assigned are able to view, edit, and delete parameter values.

1. Open the main menu, and go to **Admin > System Parameter Groups**.
2. To add a new system parameter group, select **Insert** and fill in the **Name** and **Description**.
3. Click **Save and reload**.



The screenshot shows the 'Add System Parameter Group' interface. At the top left is a back arrow icon. At the top right are three buttons: 'Save and close', 'Save and reload', and 'Save and new'. Below the title 'Add System Parameter Group' is a form with the following fields:

- SYSTEM PARAMETER GROUP** (Section Header)
- Name**: Input field containing 'SysParamG1'.
- Description**: Input field containing 'Approvers Group'.
- System Parameters**: A section for adding parameters.
- SecurityRoles**: A section for adding security roles.

You are now able to add system parameters and security roles to the system parameter group.