Core Policy Admin 3.4.0

User Guide

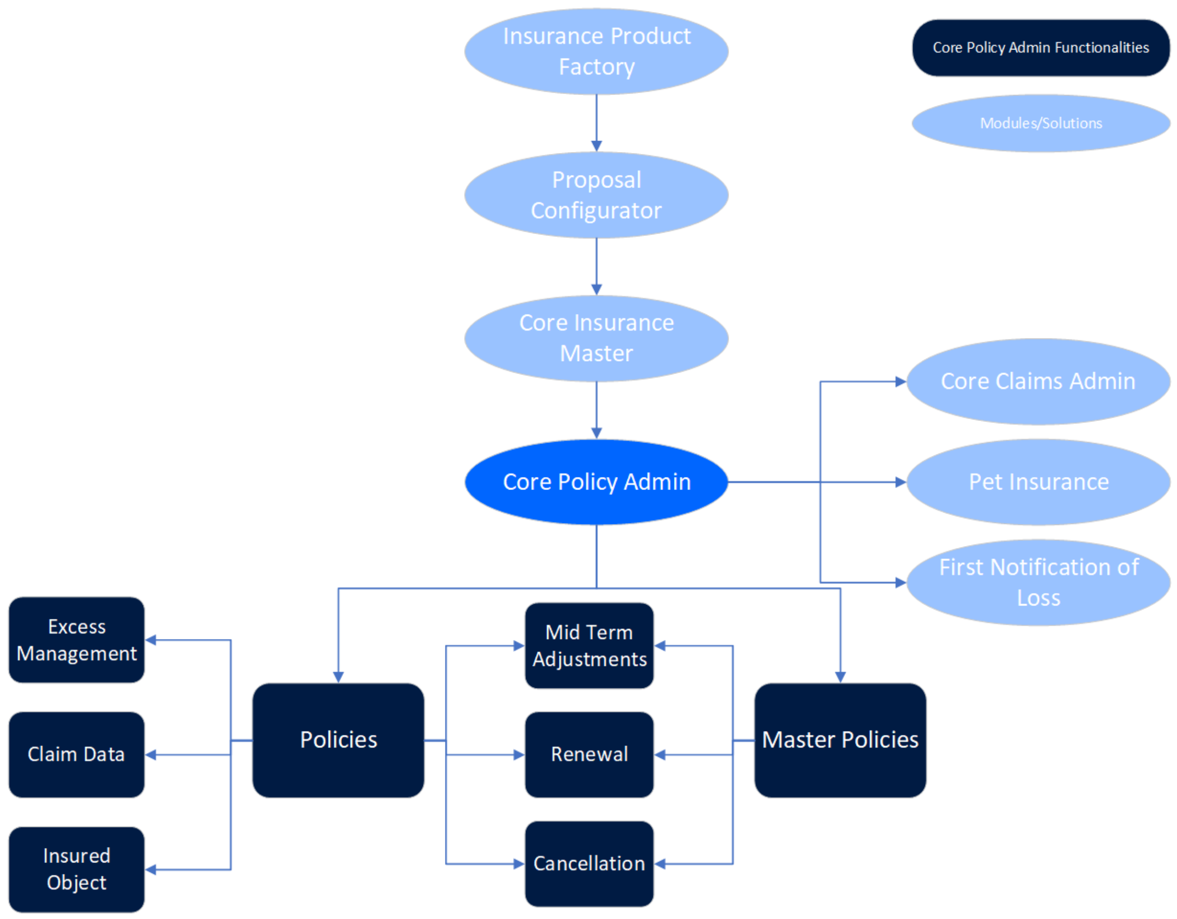
TOC

Overview

The **Core Policy Admin** module provides capabilities to support your back-office policy administration for the usual life cycle of a contract after its issuance. You can alter a policy in order to adjust the insurance contract to a customer's need, manage renewals and multi-object/person contracts, lapse, and cancel policies.

The module encompasses the registration of permitted policy alterations, such as an upgrade or downgrade in covered benefits. It automatically identifies and communicates the requirements and evidence necessary to recalculate premium and complete the alteration. Evidence and document requirements gathering is orchestrated by **FintechOS Portal** and assisted by Automation Blocks.

The diagram below presents the **Core Policy Admin** module, along with its functionalities, dependencies, and solutions that use the module.



Use **Core Policy Admin** along with other **FintechOS Northstar** solutions or accelerators to manage the life cycle of policies, Masterpolicies, and manage the claims made for them. With **Core Claims Admin**, you can create claim requests for existing policies, and create a payment proposal. Use **Core Policy Admin** with the **Pet Insurance Quote & Buy** and **First Notification of Loss** accelerators to store the policies that are generated through a Quote & Buy journey, and manage the policies that are selected to go through a First Notification of Loss journey.

This guide focuses on the use of **Innovation Studio** in the configuration of a digital journey that incorporates **FintechOS** Automation Blocks and highlights the existing opportunities to improve the process efficiency, Policy Administration automation, audit and management capabilities, and speed time to completion of administrative tasks.

The key features of the **Core Policy Admin** module are listed below:

- It enables a central point from which to initiate policy administration tasks that impact on multiple policies;
- It allows the automatic identification of business requirements pertaining to the requested changes;
- It allows the automatic initiation and completion of regular scheduled policy alterations, e.g. annual premium and benefit escalations;
- It offers high levels of automation but with the capability to determine where intervention of experts is required;
- Reverts to manual processing only where necessary for more complex or specialist cases.

Installing Core Policy Admin

Follow the guidelines below to install and configure Core Policy Admin 3.4.0.

NOTE If you already have the previous version of Core Policy Admin installed on your environment, follow the instructions from the Upgrade section.

Prerequisites

Before installing **Core Policy Admin 3.4.0**, make sure to install the following:

- **HPFI v22.1.1.0**
- **SySDigitalSolutionPackages v22.1.0003.zip**
- **Core Insurance Master v2.4.0**

Installation Steps

1. Unzip your **Core Policy Admin 3.4.0.zip** archive file.
2. Create a folder and name it **01 DeploymentPackages**.
3. Copy the **DigitalAsset** folder (if it exists) and **DigitalSolution M.N.U.P.RC.zip** file (extracted at step 1) inside the **01 DeploymentPackages** folder.

4. Locate the **FtosSysPkgDeployer** folder in the `FintechOS installation kit` (the path is `<unzipped_install_ archive>\Tools\FtosSysPkgDeployer`). You need it to install the SySDigitalSolutionPackages.
5. Select and copy the **FtosSysPkgDeployer** folder.
6. Create the **install_Syspack.bat** file needed for installation.
7. Add the following script in the file and save it next to the **FtosSysPkgDeployer** folder.

```
CD /D %~dp0
"%~dp0\FtosSysPkgDeployer\FtosSysPkgDeployer.exe" -i -a -s
"StudioLink" -u AdminStudioUser -p user_password -z
DataBaseServer -v DB_user -k DB_user_password -d
"TheNameOfTheDataBase" -r "%~dp0\01 DeploymentPackages\*.zip"
Pause
```

8. Run the **async install_SyspackDA.bat** script by double-clicking on it.
9. Follow the steps above to import and install the **Core Policy Admin Import v3.4.0**.
10. After installing, run the SQL scripts from the **Core Policy Admin Import** folder.

.bat file script parameters

- <StudioLink> - The web URL of the Innovation Studio installation, for example http://localhost/ftos_studio.
- <AdminStudioUser> - The username of the Innovation Studio user under which this import is executed. The user has to exist in Innovation Studio prior to this operation.<user_password> - The password for the Innovation Studio user.
- <DataBaseServer> - The name of the database server where the FintechOS installation database was created.
- <DB_user> - The username of the SQL Server user with administration rights on the FintechOS installation database.
- <TheNameOfTheDataBase> - The name of the database.

- <syspack_path>- The physical path to the unzipped Core Policy Admin v3.4.0 previously downloaded.
- <DB_user_password>- The password for the above mentioned SQL user.

Upgrade

If you already have the Core Policy Admin v3.3.0 package installed, upgrade to v3.4.0 by performing the below steps:

1. Check if near the **01 Deployment Packages** folder the **Upgrade** folder exists.
2. Run the SQL scripts located in the **Upgrade** folder.
3. Perform the installation steps above, steps 1-8, to install **Core Policy Admin** v3.4.0.
4. Perform the same to install **Core Policy Admin Import Upgrade** v3.4.0.
5. After installing the **Core Policy Admin Import Upgrade** v3.4.0, perform the post-installation setup listed below.

Post-Installation Setup

After installing, perform the following configurations:

1. Check the system parameter by going into Innovation Studio and setting the context to **Core Insurance Master Digital Asset** as described here.
2. Go to **Main Menu > Admin > System Parameters**, search `PolicyAdminUsed` parameter and open it for edit. Set the value to 1.

IMPORTANT!

- Enable the PolicyAdminUsed parameter to be able to enable Policy admin automatic transitions.
- Core Policy Admin Import v3.4.0 is mandatory to be imported after Core Policy Admin v3.4.0 otherwise the policy automatic transitions might not work as expected.

3. Add Vault keys:

For Portal app-settings:

```
{
"baseUrlApi": "PORTALAPI_URL "> //URL of the portal site
using EBSDefaultAuthentication = EB
"clientApi": "yourClient",
"userApi": "yourUserName",
"passwordApi": "youUserPass",
"SMTP:Port": "***",> //Your SMTP port<
"SMTP:Host": "***",> //Your SMTP host<
"SMTP:EnableSSL": "0",
"SMTP:User": "",> //Your SMTP user<
"SMTP:Password": "",> //Your SMTP password<
"DefaultFromEmail": ""> //Your SMTP default from email
address<
}
```

For **Innovation Studio** app-settings:

```
{
"SMTP:Port": "***",
"SMTP:Host":"***",
"SMTP:EnableSSL":"0",
"SMTP:User":"***",
"SMTP:Password":"***",
"DefaultFromEmail":"***"
}
```

For **Job Server** app-settings:

```
{
"UploadFolder": "yourPath:\Sites\UploadEBS",
"AttachmentPath": "yourPath:\Sites\UploadEBS",
"FileUploadWhiteList":
".pdf,.doc,.docx,.els,.jpg,.jpeg,.xlsx,.dll,.ppt,.pptx,.txt,
.png,.ttf,.xml",
"baseUrlApi": "PORTALAPI_URL *",
"clientApi": "yourClient",
"userApi": "yourUserName",
"passwordApi": "youUserPass",
"SMTP:Port": "***",
"SMTP:Host": "***",
"SMTP:EnableSSL": "0",
"SMTP:User": "***",
"SMTP:Password": "***",
"DefaultFromEmail": "***",
}
// *  URL of the portal site using EBSDefaultAuthentication =
EBS
// *** = your SMTP information
```

IMPORTANT! The **UploadFolder** and **AttachmentPath** keys are not needed for a job server installed as an web app. Instead, use the standard configuration steps to allow the job server access to the blob storage used by the other sites (to the same UploadEBS folder).

Security Roles for Core Policy Admin

A security role is a set of privileges and levels of access to various actions/ functions within the **High Productivity Fintech Infrastructure**. Use security roles to protect sensitive data, and allow better communication, collaboration, or reporting. For more details, see also the Security Roles documentation.

The following security roles are available for **Core Policy Admin** allowing the users to only perform the actions attributed to them:

Security Role	Description
Policy info	Users with this security role only have the right to see the Policies and Masterpolicies list and form (and the Policies + Masterpolicies menu entry) without having the possibility to edit.
Policy user	Users with this security role have read-only rights for Policies and Masterpolicies. They also have rights for policy versioning process, and r ights to insert a cancellation request, without the possibility to approve it.
Policy superUser	Users with this security role have the rights to insert Policy alterations and work on them. Reinitiate payment returns and make the payment returns (last buttons on the Cancellation form). The Policy superUser has to either make a payment return or reinitiate payment return.
Policy manager	Users with this security role can make cancellation approvals and returns approvals.

The following are the defined security privileges per every role (view, insert and edit):

Functionality	Policy info	Policy user	Policy superUser	Policy manager
Masterpolicies	View	View	View	View
Policies	View	View Edit	View	View
Installments	View	View	View Edit	View
Installment payment allocation	View	View	View	View
Invoices	View	View	View	View

Functionality	Policy info	Policy user	Policy superUser	Policy manager
Policy coverages	View	View	View	View
Alterations			View Insert Edit	View
Cancellations		View Insert Edit	View Edit	View Edit
Request approval cancellation		View	View	View Edit (approval)
Premium Reimbursements		View	View Edit	View Insert
Return approvals		View	View	View Edit

The table below presents which menu items are accessible for every security role:

Menu item	Policy info	Policy user	Policy superUser	Policy manager
Masterpolicies	x	x	x	x
Policies	x	x	x	x
Alterations			x	x
Cancellations		x	x	x
Premium Reimbursements			x	x
Policy Versionning		x		

Manage Core Policy Admin

With increasing digitization, the amount of data and the number of data points is growing faster and faster – and their intelligent and fast use can be a pressure for insurers. When you use **Core Policy Admin** for collecting, storing and processing policy data, you also save time for: acquiring new customer segments, tapping into other customer needs, or creating new products and services.

The **Core Policy Admin** module keeps a traceability during the lifetime of an insurance contract and its adjustments through time. The module is comprised of the following functionalities:

- The Policies repository - for storing and managing your digital insurance policies.
    - The Policy Mid Term Adjustments flow - for making various changes on the policy with different impacts such as changes in the existing coverage on the policy, changes in the type of payment, changes in the frequency of payment and more.
    - The Policy Cancellation flow - for managing cancellation processes; from registering and validating policy cancellation requests, to calculating the amount to be returned and approving payments.
    - The Policy Automatic Renewal flow - for generating renewal offer policies for those which are due to expire.
    - The Excess Management flow - for capturing and storing the excess for coverages, sub-coverages and risks.
    - The Policy Claim Data flow - for getting claims data from an external source in order to be used in the views for Policy Claims data and other processes.
    - The Insured Object flow - for saving relevant data for each object that has a policy issued.
- The Master Policies flow - for creating bundled policy offerings, covering general insurance, healthcare and life protection.
    - The Master Policy Mid Term Adjustments flow - for making various changes on masterpolicies.

- The Master Policy Automatic Renewal flow - for generating renewal offers for the masterpolicies and the policies related to them.
- The Master Policy Cancellation flow - for managing the cancellation process for masterpolicies.

Core Policy Admin Key Steps

Core Policy Admin is designed to offer you a streamlined route for managing different types of changes affecting an insurance policy during its lifetime. To access the **Core Policy Admin** module take the following steps:

1. Go to your **FintechOS Portal** and click on the main menu.
2. From the drop-down, click to open **Core Policy Admin**. Inside the **Core Policy Admin** drop-down:
   - If you need to search inside your policy repository, click Policies;
   - If you need to register a new type of payment made on a policy, click Change Payment Type;
   - Click Policy Cancellation to trigger the cancellation journey for a policy.

Automated Flows

When they satisfy some given conditions for changing their business status, policies can automatically be handled by the solution. **Core Policy Admin** has the capability to meet the following policy administration needs, without intervention from an operator:

Policy Origination

The Policy issuance process represents the main **Core Policy Admin** functionality through which the insurance contracts are generated within the Core system, on the basis of which the other **Core Policy Admin** processes are based.

From a business perspective, this functionality involves an end-to-end process that starts from the generation of the initial policy through a specific endpoint, is updated through an update endpoint and then moves to **InForce** business status, according to the contractual **Start Date**.

Policy origination endpoint

The policy issuance endpoint brings business value to the **Core Policy Admin** component by creating a connection between various external systems and the core system. Thus, once the integration with another system in order to take over the information is achieved, the policy is generated in the system according to the information received with reference to the policy to be created.

Within this endpoint, a policy generation object is structured to contain all the information that must be taken over within an integration with an external system.

Following the made request, the response body populates the system with the new data specific to the policy, in addition to those obtained during the data collection from an external system.

Policy Lapsing

When they satisfy some given conditions for lapsing, policies can automatically be moved from **InForce** to **Lapsed** status. Lapsing occurs when there is no payment, for an agreed period of time, of the latest installment on the contracted policy. The lapsing process of a policy represents an automatic process performed by means of configuration items specific to the lapsing process: the DAUDD insurance flow parameter and the FTOS_PA_Policy Lapsed scheduled job.

Policy Maturity

When they satisfy some given conditions for termination, policies can automatically be moved from **InForce** to **Maturity** status. The solution has an automated job in place in order to help with moving the policies to **Maturity** status.

Policy Renewal

When they satisfy some given conditions for renewal, policies can automatically be moved from **Maturity** to **Renewed** status. The solution has an automated job in place in order to help with moving the policies to ready for renewal procedures.

Business Workflows

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.

Policy Workflow

The Core Policy Admin management system allows policy operators to achieve efficiency, gain flexibility and organize their insurance policy data for analysis and operational purposes in an easy way.

Policy Statuses

The **Core Policy Admin** module accommodates the following business states for a policy:

Status name	Type	Description
Proposal	Initial	The first status for every policy generated into the system. Can be tailored to apply to policies waiting for their first installment to be paid.
Issued	Initial	For policies that passed their first **Paid** premium.
InForce	Ongoing	For policies passed their **Begin Date** that also meet all their contract terms - for ex. current date is not the **End Date** of the policy, payments made on time and so on.

Status name	Type	Description
Withdraw	Final	For policies closed before reaching the **Issued** status due to the following reasons: first premium not paid in a certain period following the proposal, conflicting parameter configurations between the insurance product and the policy or a system error - for ex. duplicated policy.
Surrendered	Final	For policies closed by the insurer for reasons other than those exposed in the . This status is reached through the cancellation flow only.

Status name	Type	Description
WithdrawClientRequest	Final	For policies closed following the policy holder request. This status is reached through the cancellation flow only.
ClosedByClaim	Final	For policies which exhausted their coverage to claims made by the policy holder. This status is reached through the cancellation flow only.
DeclineScreening	Final	For policies being pursued by a high risk customer. This status is reached through the cancellation flow only.

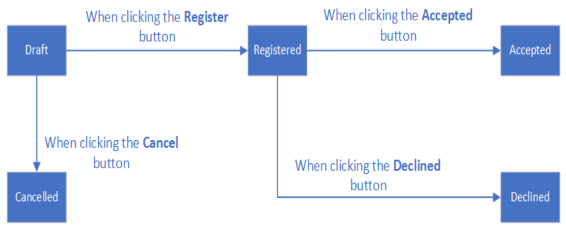
Status name	Type	Description
Lapsed	Final	For policies on which premiums were not paid, after the Payment Grace Period expired.
Maturity	Final	For policies passed their **End Date**.

Change Policy Request Business Workflow

The business statuses for a change policy request are described below:

- **Draft**: This is the initial status when inserting a new change policy request. In this status, all fields are editable, with a few exceptions.
- **Registered**: This is the intermediary status for an MTA request when the user clicks the **Register** button. In this status, all the fields become read-only. In this situation, from a business perspective, is the decision of the MTA user if continues with the MTA request or not.
- **Cancelled**: This is the final status if the user clicks the **Cancel** button in the interface. In this status, all the fields become read-only. In this situation, from a business perspective, is the decision of the MTA user if they continue with the MTA request or not.

- **Accepted**: This is the final status if the user clicks the **Accepted** button. In this status, all the fields become read-only. In this situation, from a business perspective, is the decision of the MTA user if they continue with the MTA request or not.
- **Declined**: This is the final status if the user clicks the **Declined** button. In this status, all the fields become read-only. In this situation, from a business perspective, is the decision of the MTA user if they continue with the MTA request or not.


Draft
When clicking the Register button
Registered
When clicking the Accepted button
Accepted
When clicking the Cancel button
Cancelled
When clicking the Declined button
Declined

Policy Status Transitions

Here is a description of the transitions managed through the **Core Policy Admin** module:

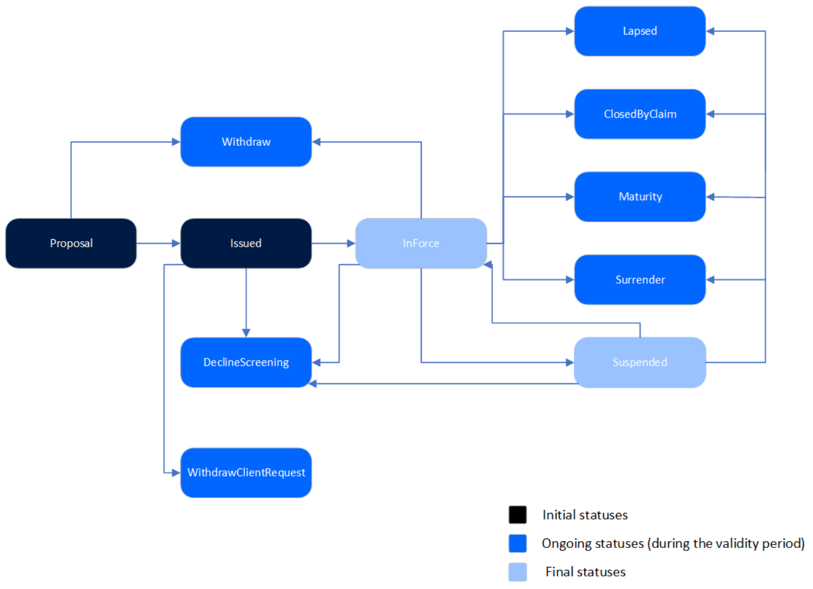
Transition	Description
_Proposal	The first status for every policy generated into the system.
Proposal_Issued	Automatic transition after policy generation in the system. Mention: This transition is properly used in a delivery project where the first paid installment triggers this status change.

Transition	Description
Proposal_Withdraw	Automatic transition before reaching the **Issued** status due to the following reasons: first premium not paid in a certain period following the proposal, conflicting parameter configurations between the insurance product and the policy or a system error - for ex. duplicated policy. This transition is triggered by FTOS_INSPA_Policy_ AutomaticallyWithdraw scheduled job using the PWDAY parameter value.
Issued_InForce	This transition can be triggered in 2 different ways: Automatically - when the policy includes an insurance Product on which **Automatic InForce** is set to **True**, at product level, the FTOS_INSPA_Policy_ IssuedToEnforced scheduled job moves the policies meeting the conditions for enforcement from **Issued** to **InForce** status. Manually - triggered by a specific request - for example Policy related endpoints, on policies that include Products with **Automatic InForce** set to **False**, at Product level.

Transition	Description
Issued_DeclineScreening	Automatic transition triggered by choosing the **Decline By Screening** reason type, during the Cancellation flow.
Issued_ WithdrawClientRequest	Automatic transition triggered by choosing the **Withdrawal** reason type, during the Cancellation flow.
InForce_Withdraw	Manual transition that can be made when the insurer decides that the policy needs to be cancelled and a Cancellation flow is not necessary - for example when policy duplication occurs due to operational mistakes.
InForce_ WithdrawClientRequest	Automatic transition triggered by choosing the **Withdrawal** reason type, during the Cancellation flow.
InForce_Surrender	Automatic transition triggered by choosing the **Surrender** reason type, during the Cancellation flow.
InForce_Maturity	Automatic transition for policies that reach their contractual **Policy End Date**. FTOS_PA_ PolicyTerminationProcess scheduled job verifies whether the condition `Policy End Date is Current Date` applies to the policy and triggers the status transition. After transitioning, the **Core Policy Admin** keeps the policy status updated.

Transition	Description
InForce_Lapsed	Automatic transition for policies on which premiums were not paid for a number of days, after the Payment Grace Period expired. FTOS_PA_PolicyLapsed scheduled job verifies each policy having this condition and triggers the status transition.
InForce_DeclineScreening	Automatic transition triggered by choosing the **Decline By Screening** reason type, during the Cancellation flow.
InForce_ClosedByClaim	Automatic transition triggered by choosing the **Closed By Claim** reason type, during the Cancellation flow.

Here is a diagram with the transitions managed through the **Core Policy Admin** module:



Click here to download the Policy Status Transitions workflow diagram in Visio format (.vsdx)

Multipolicies Contract Business Workflow

- The first status of the master policy is **Draft**, after the user clicks the **Insert** button.
- The status moves from **Draft** into **Proposal** automatically when the associated policies are generated.
- From **Proposal**, the status can move to:
    - **Issued**, through a call from the **Quote&Bind** module.
    - **Withdraw**, automatically if the master policy status does not move from **Proposal** into **Issued** within a specific number of days.
- From **Issued**, the status can move to **Inforce**, when the current date is the same as the start date of the master policy.
- From **Inforce**, the status can move to:
    - **Cancelled**, due to different aspects, such as a duplicated policy, or on a client request but after the cooling off period.
    - **Lapsed**, when the installments are not paid until the due date.
    - **Maturity**, when the current date is the same as the end date of the master policy.
    - **WithdrawClientRequest**, on a client request for personal reasons, within the cooling off period.

Master Policy Payment Schedule Business Workflow

The trigger for the Master Policy installments status transitions are the installments statuses of the policies associated with said Master Policy.

There is a specific job, for Master Policies, which runs daily and checks the installments statuses of the policies that are included in a Master Policy and have the same due date.

Master Policy Payment Schedule Status Transitions

Here is a description of the transitions for the master policy payment schedule.

Transition	Description
_on time	When generated, all the Master Policy installments are generated in this status for the entire period. They are available, but without an issued statement.
on time_ statement issued	When the statement is generated, the statement is issued and all the installments are correlated on that statement, if all the installments of the associated policies have the **Statement Issued** status, then the Master Policy installment status is also **Statement Issued**.
statement issued_paid	If all the associated policies are paid before the due date, and they have the **Paid** status, then the Master Policy installment is also **Paid**.
statement issued_ unpaid	If at least one associated policy is not paid before the due date and has the **Unpaid** status, then the Master Policy installment status is also **Unpaid**.

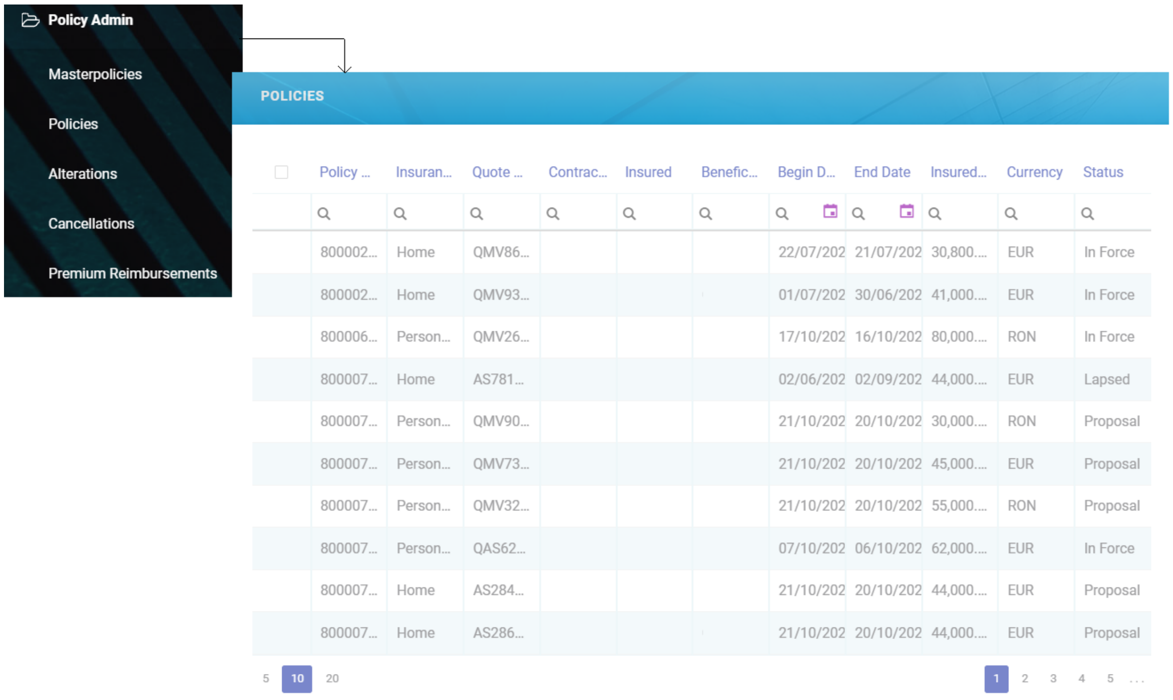
Transition	Description
paid_ statement issued	When a payment for at least an associated policy is deallocated, the statement of that policy is not covered anymore, so it becomes generated instead of closed or paid. The result being the **Statement Issued** status for the Master Policy installment also.
paid_unpaid	When a payment for at least an associated policy is deallocated, so the installment status for that policy becomes **Unpaid**, the installment status for the Master Policy also becomes **Unpaid**.

Policies

The **Core Policy Admin** module enables you to store and manage a virtually unlimited number of digital policies. Use **Policies** when you want to see all the policies stored in your environment. It is also here where you see the latest **Policies** generated into your system. However, for an in depth analysis of your portfolio, use **Policies Report** to extract the needed data.

For accessing the repository, follow the next steps:

1. In **Fintech OS Portal**, in the main menu, navigate to **Policy Admin** > **Policies**.
2. The **Policies** window opens, and you can access the repository.



Policy ...	Insuran...	Quote ...	Contrac...	Insured	Benefic...	Begin D...	End Date	Insured...	Currency	Status
800002...	Home	QMV86...				22/07/202	21/07/202	30,800....	EUR	In Force
800002...	Home	QMV93...				01/07/202	30/06/202	41,000....	EUR	In Force
800006...	Person...	QMV26...				17/10/202	16/10/202	80,000....	RON	In Force
800007...	Home	AS781...				02/06/202	02/09/202	44,000....	EUR	Lapsed
800007...	Person...	QMV90...				21/10/202	20/10/202	30,000....	RON	Proposal
800007...	Person...	QMV73...				21/10/202	20/10/202	45,000....	EUR	Proposal
800007...	Person...	QMV32...				21/10/202	20/10/202	55,000....	RON	Proposal
800007...	Person...	QAS62...				07/10/202	06/10/202	62,000....	EUR	In Force
800007...	Home	AS284...				21/10/202	20/10/202	44,000....	EUR	Proposal
800007...	Home	AS286...				21/10/202	20/10/202	44,000....	EUR	Proposal

5 10 20 — 1 2 3 4 5 ...

Term Life policies are also displayed in the list, once these are generated through the Policy Generation API.

Creating a Policy

The policy issuance process represents the main **Core Policy Admin** functionality through which the insurance contracts are generated within the core system, laying the basis for other **Core Policy Admin** flows. This functionality involves an end-to-end process that starts from the generation of an initial Policy through a specific endpoint, is updated through an update endpoint, and then reaches the **InForce** status, according to the period of validity.

NOTE A **Policy** can be introduced into the system only automatically, by the policy issuance endpoint.

The policy issuance endpoint creates a connection between the core system and various external systems - such as websites, apps and other digital channels that you use in order to reach out to customers. Once the integration with another system that sends in customer data is obtained, **Core Policy Admin**generates a policy according to the received information.

Within this endpoint, a policy generation object is structured to collect the necessary policy related information through the integration with an external system. Following the request made, the response body issues into the system the new data specific to the policy, in addition to those obtained during the data collection from a given external system.

For more details go to the Policy Generation API page.

Perform Policy Mid Term Adjustments

The Policy Mid Term Adjustments or Policy Alteration functionality allows you to make changes to the active policies, according to the customer's request. Through this functionality, you can make various changes on the policy with different impacts such as changes in the existing coverage on the policy, changes in the type of payment, changes in the frequency of payment and more. Each change in the policy also generates a new version of the policy so that the information is up to date and correct, according to customer requirements. The Mid Term Adjustments functionality has an impact on the value of the insurance premium, as through changes in coverage, the insurance premium may increase or decrease.

Updating Policies

After you register the Mid Term Adjustment (MTA) request, the new policy version is displayed in a new tab, so you can see the updated information.

1. Click the **Register** button to create a new MTA request.
2. The MTA request tab is displayed next to the first tab. This displays the updated policy after MTA changes, showing the entire policy form, as it is for the **Core Policy Admin** module, and all the fields are read-only.

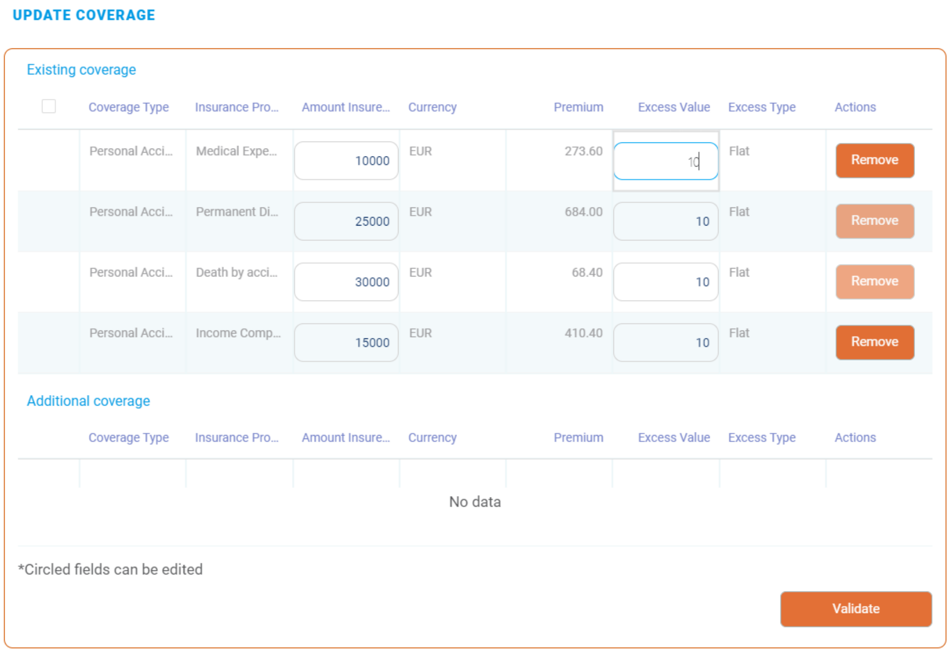
The displayed updated policy is the current policy version:

- In Version **Draft** status for versioning before approving or declining the MTA request.
- In Version **Unapproved** when declining the MTA status.

- After approving the MTA request, the **Approved** policy version is displayed.

Alternatively, you can initiate the MTA flow from the policy form. The **Actions** section at the end of the form enables you to trigger this journey directly from the policy.

1. On the policy form, click the **Issue MTA** button to initiate the flow. You have 3 adjustment options:
    - Update Coverage;
    - Change Payment Type;
    - Change Frequency.
2. When you click the **Update Coverage** button, you can edit both the **Amount Insured** and the **Excess Value** fields.



UPDATE COVERAGE

Existing coverage

	Coverage Type	Insurance Pro...	Amount Insure...	Currency	Premium	Excess Value	Excess Type	Actions
	Personal Acci...	Medical Expe...	10000	EUR	273.60	10	Flat	Remove
	Personal Acci...	Permanent Di...	25000	EUR	684.00	10	Flat	Remove
	Personal Acci...	Death by acci...	30000	EUR	68.40	10	Flat	Remove
	Personal Acci...	Income Comp...	15000	EUR	410.40	10	Flat	Remove

Additional coverage

Coverage Type	Insurance Pro...	Amount Insure...	Currency	Premium	Excess Value	Excess Type	Actions
No data							

*Circled fields can be edited

Registering and Canceling an MTA Request

After registering an MTA request, you can register or cancel it, either after performing the alterations on it and validating them, or before attempting to work on the request.

1. Click the **Register** button to trigger the following actions:
   - The change policy request transitions from **Draft** to **Registered**.
   - All fields become read-only.
   - The **Updated Policy** tab becomes available displaying the entire form of the updated policy.
   - Core Policy Admin calculates the additional premium. For premium reimbursements, a negative amount is displayed, as per the formula explained below.
   - The policy alteration type is determined. If at least one included alteration requires the issuance of an MTA, the value is MTA needed.
2. After updating a policy alteration request, the coverages on a policy by editing the **Amount Insured**, or removing or adding new coverages, the impact in the premium is calculated sending a `getPrices` API to receive the new annual premium amount calculated for each coverage (item). The request is sent after the validation of each alteration type (**Register** button).

NOTE
In order for the request to be sent, you must create a policy data type mapping on the product.

View the calculations for the premium amounts after a cancellation by accessing the Core Policy Admin Formulas page.

The adjustments are made according to the new amount, the rest of the installments which are unpaid and having the status **OnTime**. Any rounded amounts are aggregated on the closest unpaid installment that does not have a statement issued.

The premium amount may be updated in case of policy changes done through an MTA, like :

- Removing or adding a coverage;
- Adjusting the insured amounts;
- Changing the payment frequency.

Approving and Declining an MTA Request

You can approve or decline an MTA request according to the customer's decision regarding the modifications on the policy.

Beside the displayed policy information, the **Change Policy Request** tab contains the **Accepted** or **Declined** buttons, which trigger specific status transitions.

- Click **Accepted** to change the policy status from **Registered** to **Accepted**. The **MTA No** is automatically updated from the policy alteration summary. The policy version status automatically changes to **Approved**.
- Click **Declined** button, to change the policy status from **Registered** to **Declined**. The policy version status automatically changes to **Unapproved**.

NOTE The actions and the buttons are displayed only for the change policy requests in **Registered** status having displayed the **Updated Policy** tab, otherwise, the buttons are not visible and the actions not possible.

MTA Refactoring According to the Prorata Type

The Premium calculations made in the Policy Alteration processes take into consideration the type of pro rata which is used in the calculations.

The type of pro rata is configured in the prorata type insurance parameter. View the calculations for the premium amounts according to the prorata type Core Policy Admin Formulas page.

Underwriting Rules

After an MTA is registered, the system not only checks the pricing, but also the underwriting (UW) rules. If the check for the UW rules has a response that is has not passed, the following pop-up is displayed: “The alterations brought through this MTA(s) did not pass the Underwriting rules check. Please decide if you want to edit the detail of the MTA or cancel the action”. In this pop-up:

- If you click **Decline the MTA(s)**, the MTA is cancelled.
- If you click **Edit MTA**, the system redirects you to the MTA Request view where all the values are reverted to previous policy values.

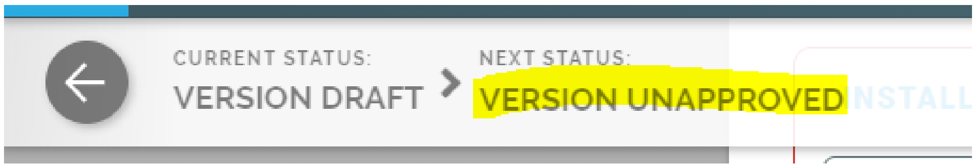
Manage Policy Versioning

There are cases, like policy renewal when you need to create a new version of a specific policy. To accommodate the versioning functionality, from an insurance business perspective, the **Core Policy Admin** solution uses a combination of the **FintechOS** standard versioning process mechanism and custom development. Consequently, you use the FTOS_VersioningHelper client side library and follow the standard procedure when configuring version settings, version settings items and entity settings. However, for policy versioning, you use the **FTOS_VersioningHelper_ Edit** client side library in order to keep some attributes **Read Only**, even when the policy is in **Version Draft** business status, with **isEditable** option enabled.

Policy Versioning Process Description

The versioning functionality is available only for **Issued**, **In Force** or **Approved** policies.

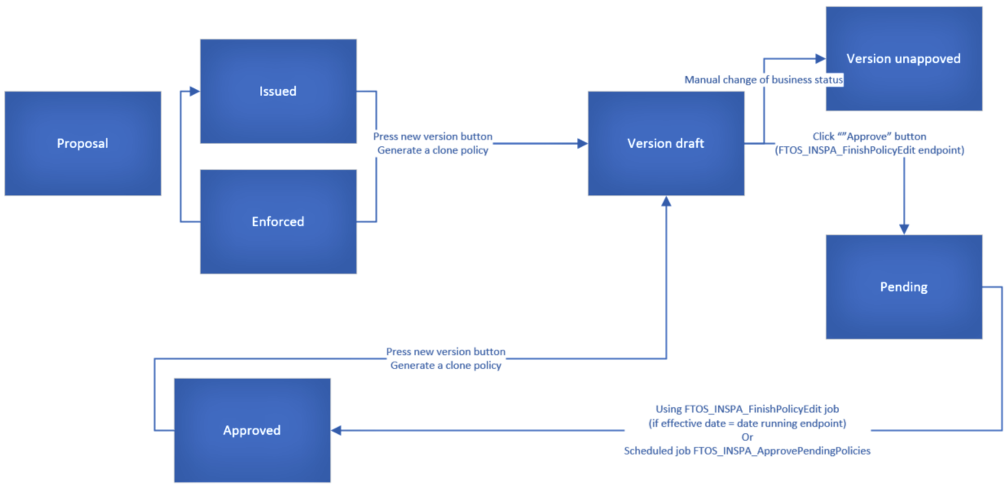
1. Click **New Version** on the initial policy, and a policy clone is registered, in **Version Draft** status.
2. You can change the status of the new policy to **Version Unapproved** by following the standard process (manually change the status of the record).



The standard approval process (changing the status from Version **Draft** to **Approved**) is an automated process, especially implemented for the policy.

3. Fill in the effective date for the change and click **Approve**. The version remains in **Version Draft** status until the effective date is reached and it isautomatically transitioned to **Approved** status, on that day.

If the effective date is also the date when you validate the changes by pressing **Approve**, then the request automatically transitions from **Version Draft** to **Pending** and also from **Pending** to **Approved**.


Proposal
Issued
Enforced
Press new version button
Generate a clone policy
Version draft
Manual change of business status
Version unappoved
Click ""Approve" button
(FTOS_INSPA_FinishPolicyEdit endpoint)
Pending
Press new version button
Generate a clone policy
Approved
Using FTOS_INSPA_FinishPolicyEdit job
(if effective date = date running endpoint)
Or
Scheduled job FTOS_INSPA_ApprovePendingPolicies

If the Policy Begin Date is greater than the current date, then the policy status is transitioned to the **Issued** status.

If the Policy End Date is lesser than the current date, then the policy status is transitioned to the **Maturity** status.

If the conditions above are not respected, then the policy is transitioned to the previous status of the policy.

The following processes automatically generate newly approved versions of the policies:

- **Cancellation** (Ex-Surrender) - for the updates of end date and premium;
- **Lapsing** - or the updates of end date and premium.

- **Cancellation** without an update of the premium (eg. Cancellation with Claims), the schedule has to be updated with a schedule including just the paid installments in the paid status and the future unpaid installments in Canceled status (new status for PaymentScheduleDetail);
- **Lapsing** - same as above;

- **Cancellation** with premium returned the schedule has to be updated with a schedule including the paid installments, the future unpaid installments in **Canceled** status plus an additional installment which is equal with :minus: the value of the returned amount. The due date is equal to the approval date of the cancellation and the status of this payment schedule detail is On Time (TBD).

After the automatic version approval process, the status for policy must be changed according to the process which triggered the versionning:

- If the version approval has been made after the **Lapsing** process - the policy status is transitioned from **Approved** to **Lapsed**.
- If the version approval has been made after the **Cancellation** process - the policy status is transitioned from **Approved** to the specific status for **Cancellation**: Closed by claim, Decline by screening etc.

Business Workflow Configuration Actions

Pending - Approved - When you click **Approve**, the script checks the effective approval date. If it is less than or equal to the current date, it moves the policy version record into the **Approved** status. If not, the record remains in **Pending** status until the **FTOS_INSPA_ApprovePendingPolicies** job moves it to **Approved**, according to the effective chosen date.

Version Draft - Pending - If the necessary validations are met, clicking **Approve** changes the status of the cloned policy from **Version Draft** to **Pending**.

A new version of the policy can be made from the **Issued** and **InForce** statuses.

The **Approved** status is not visible in the interface, and the transition is automatically made to the business status of the policy.

For advanced versioning configurations, endpoints, and server side scripts, check the Advanced Versioning Configurations page.

Cancel a Policy

When a termination of an insurance policy is initiated either by insured or by you, the insurer, you can check in what way the cancellation request falls under the policy scope, and fulfills the request under the agreed terms of the policy. Once approved, you issue a payment, if the case, to the insured, or to an approved third party on behalf of the insured. **FintechOS** clients use the **Core Policy Admin** module to organize and automate routines of this typical scenario, in order to increase the efficiency and accuracy of their operations.

NOTE **Core Policy Admin** makes sure that the same policy is not available inside two cancellation flows, simultaneously. In turn, a policy can be accessed sequentially, by different operators, inside the same cancellation flow - for example, when a user manages the cancellation processing but a super user must approve the **Returned Premium** payment.

A policy is terminated and is not considered active if:

- The first installment wasn't paid. If so, after the prescribing period, the policy becomes **Withdraw**.
- The customer wants to terminate the policy before 14 days from issuance. If so, the policy becomes **Withdraw on Client Request**.
- The policy reached **Maturity**.
- The customer didn't pay the installment. If so, after the grace period, the policy becomes **Lapsed**.

- The customer wants to terminate the policy even the payments are up to date. If so, the policy becomes **Cancelled**.
- The customer's risk class was changed into unacceptable risk. If so, all the customer's active polices are terminated with **Decline by Screening** status.
- The insurer reserves the freedom to decline the policy for any **Other** reason.

The above conditions currently shape the policy termination flows but **Core Policy Admin** is a highly customizable solution and you are able to change it according to your needs. For doing so read the Policy Configurations page.

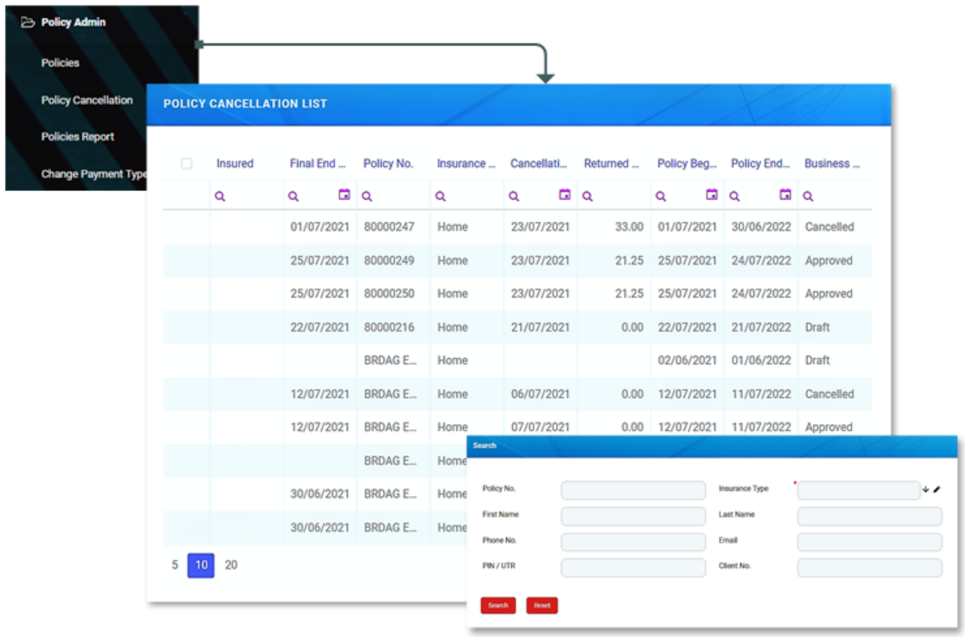
Manage Policy Cancellation Requests

In order to process a policy cancellation request, you must access the **Policy Cancellation List** that displays all the available **Policy Cancellation** records from the database.

In order to do so, follow the instructions from below:

1. In the main menu, navigate to **Policy Admin > Cancellations**. The **Cancellation List** is displayed.
2. In the **Policy Cancellation List** page:
   - **Add** a new **Policy Cancellation** record, by clicking **Insert**.
   - **Edit** a **Policy Cancellation** record from the grid, by double-clicking it. When opening an existing **Cancellation request**, the **Edit form** becomes available, with the information previously introduced. Use the form to update the desired fields. Click **Save and Close**.
   - **Delete** a **Policy Cancellation** record from the grid by selecting it and

clicking **Delete**.



POLICY CANCELLATION LIST

	Insured	Final End ...	Policy No.	Insurance ...	Cancellati...	Returned ...	Policy Beg...	Policy End...	Business ...
		01/07/2021	80000247	Home	23/07/2021	33.00	01/07/2021	30/06/2022	Cancelled
		25/07/2021	80000249	Home	23/07/2021	21.25	25/07/2021	24/07/2022	Approved
		25/07/2021	80000250	Home	23/07/2021	21.25	25/07/2021	24/07/2022	Approved
		22/07/2021	80000216	Home	21/07/2021	0.00	22/07/2021	21/07/2022	Draft
			BRDAG E...	Home			02/06/2021	01/06/2022	Draft
		12/07/2021	BRDAG E...	Home	06/07/2021	0.00	12/07/2021	11/07/2022	Cancelled
		12/07/2021	BRDAG E...	Home	07/07/2021	0.00	12/07/2021	11/07/2022	Approved
			BRDAG E...	Home					
		30/06/2021	BRDAG E...	Home					
		30/06/2021	BRDAG E...	Home					

5 10 20

Search

Policy No.		Insurance Type	
First Name		Last Name	
Phone No.		Email	
PIN / UTR		Client No.	

Search Reset

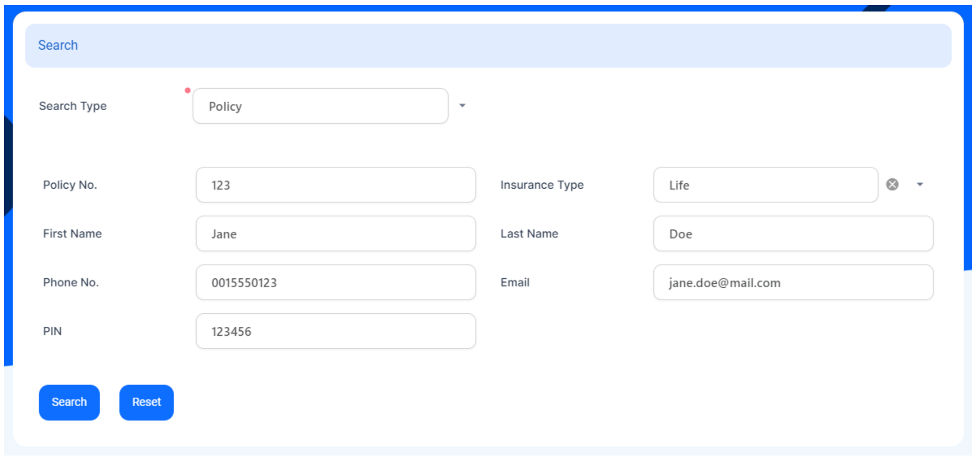
Alternatively, you can initiate the cancellation flow explained above from the policy form. The **Actions** section at the end of the form enables you to trigger this journey directly from the policy. Click the **Cancel Policy** button to initiate the flow.

Search for Cancelled Policies

1. To find a specific **Policy Cancellation** record, click **Insert** on the **Cancellation List** page.
2. The **Policy Search** form is displayed. Use it to input data about the policy you need to cancel and hit the **Search** button to find it.

NOTE

The same **Search** form is used for both type of customers: individuals and companies. Only the policies that are in the status of **InForce** are displayed in the list of policies.

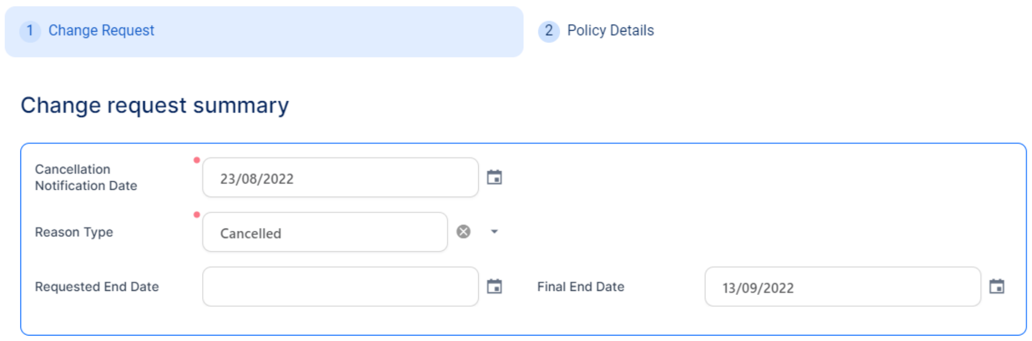


Search Type	Policy		
Policy No.	123	Insurance Type	Life
First Name	Jane	Last Name	Doe
Phone No.	0015550123	Email	jane.doe@mail.com
PIN	123456		

3. Click **Search**. A new list of policies is displayed. Choose the policy to be closed.
4. Click **Refresh** to reload the existing search results. If needed, reset the search process by clicking **Reset**. All the information displayed in the previous search is erased.
5. From the search results, you can choose only one policy for the **Policy Cancellation**. Click **Choose option** near the policy record. A new form related to the cancellation process is displayed.
6. At this stage, a new cancellation record is registered by default in **Draft** status. This record is later transitioned to other statuses according to the Policy Cancellation process steps. You cannot save a request if there is no owner or no policy.
7. Proceed forward with the change request form.

View the Change Request and Policy Details

This is where the registration of the request for the policy's cancellation begins. The **Edit** form of a new Policy Cancellation record initially opens with only 2 tabs: **Change Request** and **Policy Details**. The other tabs are displayed after completing the minimum information required. Consequently, completing the **Change Request Summary** fields subsequently triggers the actual calculations for the payment amounts to be returned to the client - if the case and the calculations are displayed inside the Policy section specific fields.


1 Change Request
2 Policy Details

Change request summary

Cancellation Notification Date	23/08/2022		
Reason Type	Cancelled		
Requested End Date		Final End Date	13/09/2022

1. Fill in the **Change Request Summary** fields.
2. Choose a **Reason Type**. This triggers the calculation for the **Requested End Date** and the **Final End Date** fields as per the table below:

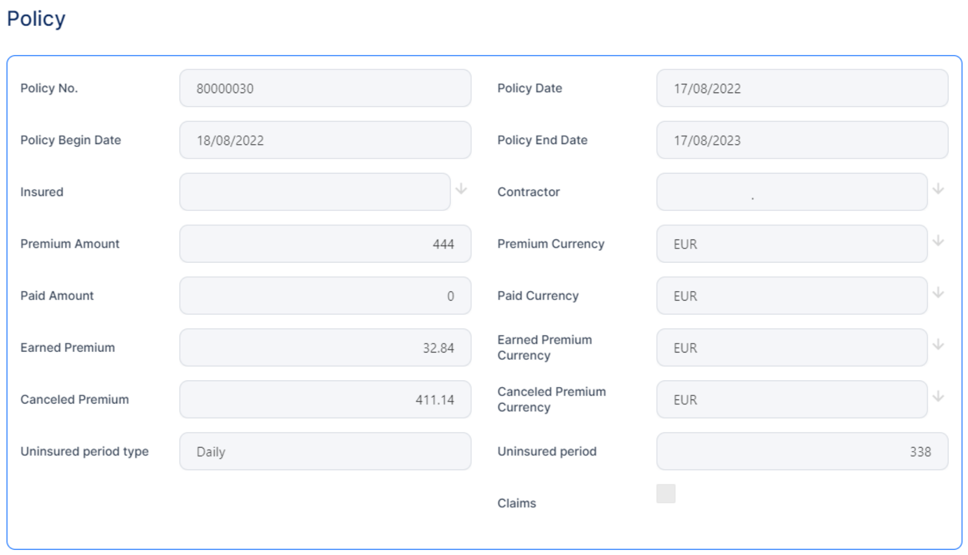
Reason Type	Requested End Date	Final End Date
Withdrawal	In this case, it is automatically completed with the **Policy Begin Date**.	In this case, it is automatically completed with the **Policy Begin Date**.
Property sold	In this case, the **Requested End Date** is set to any day before or including the day of the request for cancellation.	**Final End Date** is filled with the value from **Requested End Date** but with the possibility to edit.

Reason Type	Requested End Date	Final End Date
Cancelled by Client	You can manually complete the field with any date.	If Notification Date -Policy (Issued) Date <= 15 then this field is automatically completed with the Policy Begin Date with the possibility to be adjusted. Otherwise this is automatically completed with the Notification Date + 21days to go
Decline by screening	In this case, you can manually complete the field with any date.	You can fill the field with the value from **Requested End Date** but with the possibility to edit.
Cancelled	Here you can manually complete the field with any date, by the following rule: If the **Notification Date** - **Policy (Issued) Date** <= 15 days, then the **Requested End Date** is filled with the **Policy Begin Date**.	Here it is automatically completed with the Notification Date + 21 days to go. If the **Requested End Date** meets the rule explained above, the **Final End Date** is filled with the **Policy Begin Date**.

Reason Type	Requested End Date	Final End Date
Closed by Claim	It is pre-filled with Notification Date + 21 days from the present on.	It is filled with the value from the **Requested End Date** but with the possibility to edit.

In case of the Pet Passed Away value for insured object Pet, this reason triggers the **Cancelled** status for the policy. The effective date is the final end date for both cancellation reason types, Cancelled by Client and Pet Passed Away.

3. Following the input of the above data, on the next section with details regarding the chosen policy, the fields are automatically filled. The pre-filled details are extracted from Policy basic information section and from the results of the calculation made by selecting the information in the **Change Request Summary** section.


Policy

Policy No.	80000030	Policy Date	17/08/2022
Policy Begin Date	18/08/2022	Policy End Date	17/08/2023
Insured		Contractor	
Premium Amount	444	Premium Currency	EUR
Paid Amount	0	Paid Currency	EUR
Earned Premium	32.84	Earned Premium Currency	EUR
Canceled Premium	411.14	Canceled Premium Currency	EUR
Uninsured period type	Daily	Uninsured period	338
		Claims	

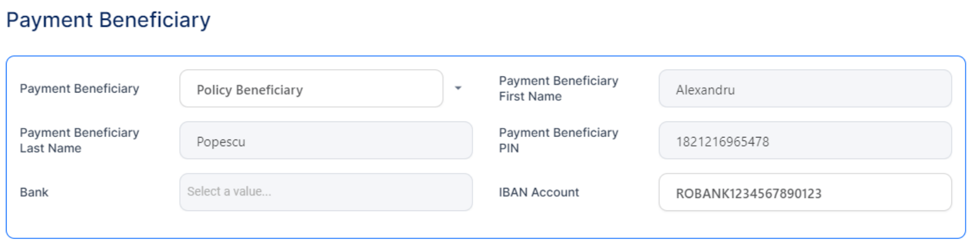
Field	Description
Policy No.	The number of the policy when issued.
Policy Date	The date when the policy was issued.

Field	Description
Policy Begin Date	The day when the policy becomes enforced.
Policy End Date	The day when the policy is no longer be available, according to the contract.
Insured	First and last name of the insured person on the policy.
Contractor	First and last name of the contractor on the policy.
Premium Amount	The premium amount of the policy.
Premium Currency	The currency of the policy.
Paid Amount	The total amount of the payments made by the customer on a contract.
Paid Currency	The currency in which the amount has been paid.

Field	Description
Earned Premium	The earned premium on the Cancellation process: The amount is calculated after the following formula: • If there are no paid or opened claims and 1. If [Cancellation Date – (Policy Begin Date+1)] <= 15, then the Earned Premium is equal to 0; 2. If [Cancellation Date – (Policy Begin Date+1)] > 15, then the amount is filled in with the following: Premium Amount/ 12 * (12 – No. of uninsured months); • If there are no paid or opened claims the Earned Premium is equal with the Premium Amount. The amount calculated should be returned to the client.
Earned Premium Currency	The specific currency of the earned premium is, by default, the premium currency.
Canceled Premium	The cancelled premium.
Canceled Premium Currency	The currency of the cancelled premium.
Uninsured Period Type	The values on a monthly or daily basis according to the Prorata type configured in the system.
Uninsured Period	Calculated as the number of months or number of days from the **Policy Begin Date** to the **Policy End Date** according to the uninsured period type set.

Field	Description
Claims	Check the box for existing claims - it should be checked if there are any claims on the policy.

4. Fill in the information regarding the customer to whom the payment must be made, in the Payment Beneficiary section. Provide the information for all the fields in this section, in order for the payment to take place. However, leave this section blank when there is no amount to be returned to the customer.



Payment Beneficiary

Payment Beneficiary	Policy Beneficiary	Payment Beneficiary First Name	Alexandru
Payment Beneficiary Last Name	Popescu	Payment Beneficiary PIN	1821216965478
Bank	Select a value...	IBAN Account	ROBANK1234567890123

5. Click **Register request**, in order for it order to go for approval.

You can also cancel this request, provided you offer a cancellation reason.

1. If you continue with the current cancellation request, click the **Register Request** to open a tab with a new **In Progress** status, while the system calculates the amount to be returned to the beneficiary. Also, the **Comments** field becomes available if there is any observation regarding the request.

Field	Description
Comments	Complete this field with relevant information regarding the cancellation process. This field is optional.

Field	Description
Resolution Reason	When choosing to cancel the registration of the policy termination request, provide relevant information about doing so. This field is mandatory. The values of the option set are: Product reasons, Other reasons, Company reason, Withdraw.

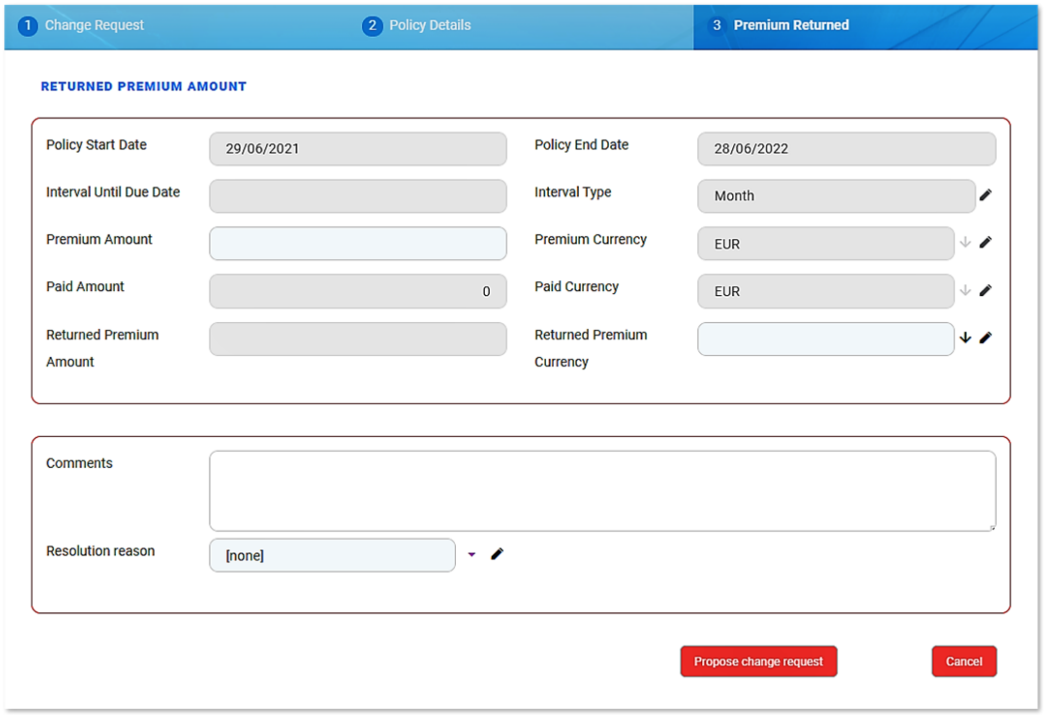
2. Click **Register request** to switch to the next tab, which represents the third step of the process. This is where you find two predefined lists that contain the valid installments and the claims that have been opened on the current policy. On this tab there is no possibility to insert or delete records.
3. Edit a specific record by double- clicking the desired row, which redirects you to the specific **Edit** form of the installment or claim.

View the Calculated Premium Returned

This is a step where all details are automatically filled in as follows:

Field	Description
Policy Start Date	The day when the policy becomes enforced.
Policy End Date	The day when the policy is no longer be available according to the contract.
Last Payment Date	The date when the last installment payment was made.
Interval Until Due Date	The remaining days until the next installment payment.
Interval Type	The type of the interval between the installment payments.
Premium Amount	The premium amount of the insurance policy.
Premium Currency	The currency of the insurance policy.
Paid Amount	The total amount of the payments made by the client on a contract.
Paid Currency	The currency of the paid amount.

Field	Description
Returned Premium Amount	The amount to be returned to the customer. Is equal to the Unearned Premium Amount result, but with the possibility for the user to edit the amount with a desired value if necessary.
Returned Premium Currency	The currency of the returned premium amount.



Policy Start Date	29/06/2021	Policy End Date	28/06/2022
Interval Until Due Date		Interval Type	Month
Premium Amount		Premium Currency	EUR
Paid Amount	0	Paid Currency	EUR
Returned Premium Amount		Returned Premium Currency	

Comments	
Resolution reason	[none]

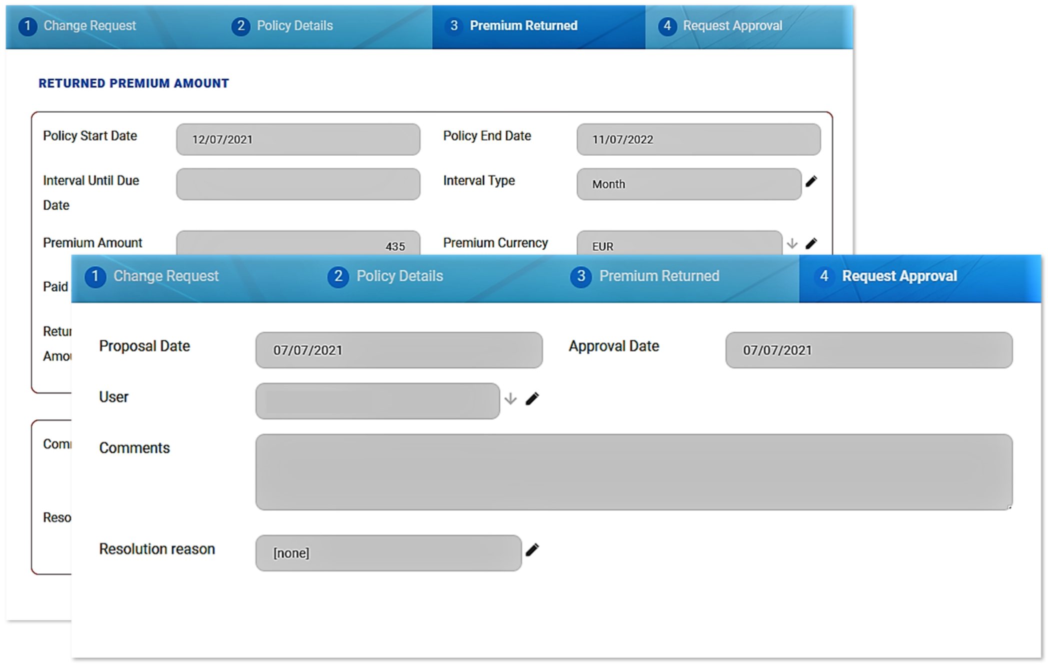
When the information regarding the value to be returned to the customer is displayed, decide whether the process continues towards the approval phase or towards closing the request.

1. Click **Propose change request** to continue with the cancellation process and send the request to approval or click **Cancel** and complete the **Resolution Reason** field. This field is mandatory.
2. Requesting approval opens a new tab, while automatically triggering the transitioning from **In Progress** status to **In Approval** status. Also, you can use the **Comments** field to transmit any observations regarding the current request.

Request Approval

Approving the request for policy cancellation requires the intervention of a **Policy Admin Super User** who holds the necessary security rights to approve or reject such cases.

Field	Description
Proposal Date	The date when the cancellation request was scheduled for approval. It is completed with the current date.
Approval Date	The date when the request was approved. It is completed with the date the cancellation application is approved.
User	It is automatically completed with the name of the Policy Admin Super User who is logged in at the approval moment.
Comments	You can fill the field with relevant information regarding the approval process or the cancellation flow.
Resolution Reason	In case of declining, the Policy Admin Super User must offer information on why the cancellation application was not approved.



Policy Start Date	12/07/2021	Policy End Date	11/07/2022
Interval Until Due Date		Interval Type	Month
Premium Amount	435	Premium Currency	EUR

Proposal Date	07/07/2021	Approval Date	07/07/2021
User			
Comments			
Resolution reason	[none]		

When the case is **Approved**, the flow continues on the approved branch as follows:

- If the **Returned Premium Amount** is equal to 0 then the process ends and the status changes from **In Approval** to **Approved**.
- If the **Returned Premium Amount** is not equal to 0 then the process continues with the **Payment Return** flow, explained in the Billing and Collection user guide. In addition, the status changes from **In Approval** to **Approved**. The system redirects the user to the tab where the date of the scheduled payment is displayed.

When the case is **Declined**, the status changes from **In Approval** to **Declined**. The process reaches the end without the possibility of starting again.

Policy Cancellation Status Transitions

The **Policy Cancellation** statuses are as follows:

Status name	Description
Draft	When you open the policy, the **Cancellation** record is by default in the draft status.
In Progress	When you input the notification date. You can move forward by pressing **Register request**.
In Approval	Following the registration of the request, for approval use **Propose change request** button.
Approved	The cancellation request is approved by the **Policy Admin** super user.
Declined	The cancellation request is declined by the **Policy Admin** super user.
Cancelled	The cancellation request is cancelled.

View the Cancellation Notifications

The following notifications are automatically sent to the customer after a policy is cancelled or lapsed:

Policy Cancellation Notification

The system automatically informs the customer of the Policy Cancellation process for their insurance contract through a specific notification sent to them.

The policies which are included in the notification are those which have the **Policy Status** = Decline by screening/ Closed by Claim/ Withdraw on client's request/ Cancelled, transitioned due to a Cancellation process.

The notification is sent when the policy status is changed to a specific cancellation status, on the effective date.

The following are the tokens used in the notification:

- Contractor name;
- Policy number;
- Insurance type;
- Policy End Date (new End Date) - the Effective date from the Cancellation process which is updated on the last active version of the policy;
- Email template name : HomeInsurance_ ClientPolicyCancelledNotification.

Lapsed Policy Notification

Similarly to the cancellation process, the system informs the client of the Policy Lapsing process for their insurance contract through a specific notification sent to them.

The Policies which included in the notification are those which have the **Policy Status** = Lapsed.

The following are the tokens used in the notification:

- Contractor name
- Policy number
- Insurance type
- Email template: homeInsurance_ClientPolicyStatusLapsed

Automatically Renew the Policies

The Policy Automatic Renewal solution helps insurers to continue providing coverage to the customer once the initial policy period has passed, at the end of the term period, so the beneficiary never goes without coverage in any field of insurance. You can opt for this automatic renewal once you initiates your first insurance policy, so as to get rid of the worry of manual renewal once the policy reaches maturity.

NOTE
Such automatic renewals may also include changes in the premium insured by rate increases or decreases.

Understanding the Automatic Renewal Process

The daily scheduled job, `FTOS_PA_PolicyRenewal`, is running in order to find all the policies from the system which have to be renewed according to their end date and the parameter set for the number of days before renewal mentioned above.

The job verifies all the policies which have the *Policy End Date (policy to be renewed Initial End Date) - No. of days before renewal (parameter set) = Current Date*, and applies the renewal process to them.

The policy automatic renewal types are described below:

New Policy/Same Validity/Actual Tariff

The system generates renewal offer policies for those which are due to expire and have the following product configuration:

- **Types of Renewal**: New Policy;
- **Renewal Validity**: Same Validity;
- **Renewal Tariff**: Actual Tariff.

The policies which have to be renewed, need to generate insurance offers which include the following information:

- Status of newly renewed policy - Proposal;
- Validity - Renewed Policy validity;
- Begin Date = Renewed policy End Date + 1;
- End Date is calculated according to the Begin Date + Validity;
- Same parties;
- Same insured object;
- Same agent/broker and distribution channel;
- Same quote number;
- Same payment type and the same payment frequency;
- Same coverages and indemnity limits;
- The premium has to be calculated based on the currently approved product version;
- The Payments Schedule is generated according to the standard logic;
- The old policy has to be mapped as renewed and have completed the Renewed by attribute with the new Policy number;
- For the new policy, save the Renewed Policy number of the old policy.

Renewal Offers/Same Validity/Actual Tariff

The system generates renewal offer policies for those which are due to expire and have the following product configuration:

- **Types of Renewal**: Renewal Offers;
- **Renewal Validity**: Same Validity;

- **Renewal Tariff**: Actual Tariff.

The policies found through the `FTOS_PA_PolicyRenewal` job which have to be renewed, need to generate insurance offers which include the following information:

- Renewed policy;
- Same Product and Insurance Type;
- Validity: Renewed Policy validity;
- Begin Date: Initial Policy End Date + 1;
- End Date is calculated according to the Start date + Validity;
- Same parties;
- Same insured object;
- Same agent or broker and distribution channel;
- Same payment type and the same payment frequency;
- Same coverages and indemnity limits;
- The total premium has to be calculated based on the currently approved product version.

New Policy/Same Old Tariff/Same Validity

The system automatically renews policies having the same premium amount. These are updated as new policies to update the system's contract. The configuration is given below:

- **Renewing Policy** = New Policy;
- **Renewal Validity** = Same Validity;
- **Renewal Tariff** = Same Tariff.

The policies found through the `FTOS_PA_PolicyRenewal` job which have to be renewed, for products with the above configuration are renewed as follows:

- Status: Proposal;
- Validity: Renewed Policy validity;
- Begin Date: Initial Policy End Date + 1;
- End Date is calculated according to the Start date + Validity;
- Same Parties;
- Same insured object;
- Same agent or broker and distribution channel;
- Same Quote;
- Same payment type and the same payment frequency;
- Same coverages and indemnity limits;
- The premium has to be calculated based on the product version valid on the issuance date of the initial policy;
- The **Payment Schedule** is generated according to the standard logic;
- The old policy has to be mapped as renewed (existing attribute) + have completed the Renewed by attribute with the new policy no;
- For the new policy, we need to save the Renewed Policy no of the policy (existing attribute);
- for the new policy, Start Date = Renewed policy End Date + 1 and End Date is calculated according to the Start date + Policy validity (to be taken from Renewal Validity attribute).

Renewal Offers/Same Validity/Same Tariff

The system generates renewal offer policies for those which are due to expire and have the following product configuration:

- **Types of Renewal**: Renewal Offers;
- **Renewal Validity**: Same Validity;
- **Renewal Tariff**: Same Tariff.

The policies found through the `FTOS_PA_PolicyRenewal` job, which have to be renewed, need to generate insurance offers including the following information:

- Renewed policy;
- Same Product and Insurance Type;
- Validity - Renewed Policy validity;
- Begin Date = Initial Policy Begin Date + 1;
- End Date is calculated according to the Start date + Validity;
- Same Parties;
- Same insured object;
- Same agent/broker and distribution channel;
- Same payment type and the same payment frequency;
- Same coverages and indemnity limits;
- The total premium is calculated based on the product version valid at the issuance date of the renewed policy.

Manage Policy Excesses

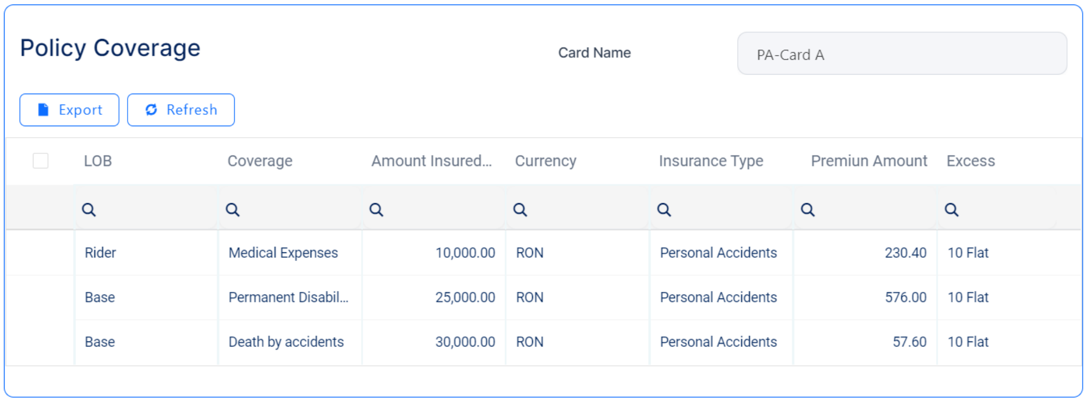
Core Policy Admin can capture and store the excess (deductibles) for coverages, sub-coverages and risks (perils). The excess represents the part of the loss that is paid by the Insured. The insurer's liability starts after the deductible or is in excess of that, hence the name.

The excess is set at coverage level and is automatically inherited (sub-coverages) and also all the risks that are bound to them. Either the system or an elevated user can modify the excess (the deductible that is set at risk level at a different value than the one that is set at coverage level). The deductibles inherit the policy currency.

On different policies from the same product, the same peril can have different excess types. For example, fire can have a 5% of loss on one policy and 5000 excess on another policy. Deductibles have the same currency as the policy they are bound to.

There are three steps where the system shows if there are any excesses.

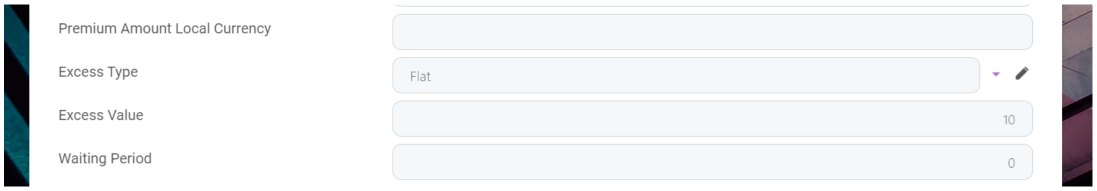
- Firstly, in the **Policy Coverage** grid of a policy record, in the **Excess** column.



LOB	Coverage	Amount Insured...	Currency	Insurance Type	Premiun Amount	Excess
Rider	Medical Expenses	10,000.00	RON	Personal Accidents	230.40	10 Flat
Base	Permanent Disabil...	25,000.00	RON	Personal Accidents	576.00	10 Flat
Base	Death by accidents	30,000.00	RON	Personal Accidents	57.60	10 Flat

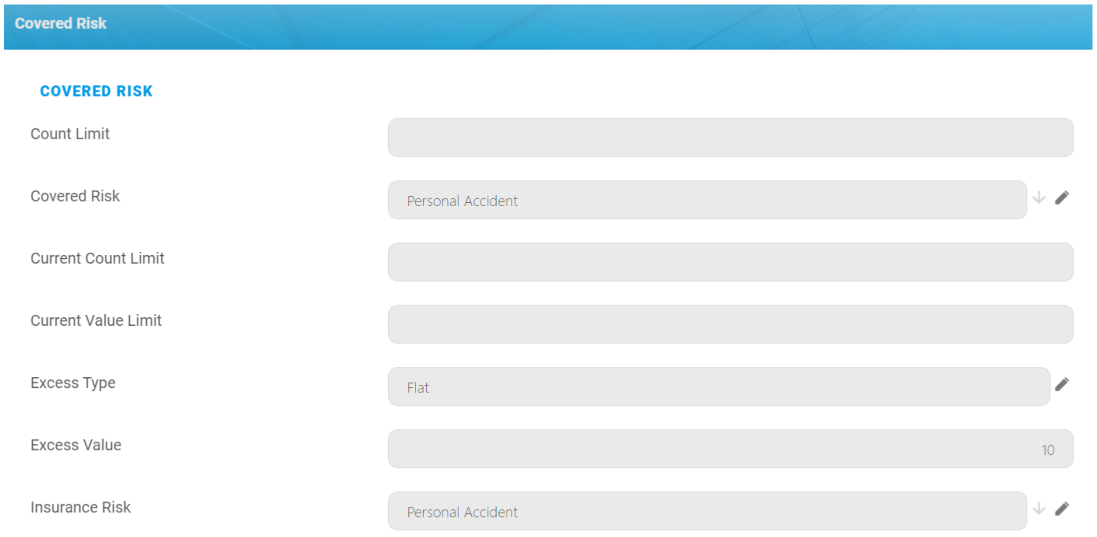
  The Excess types can be: Flat amount, Percentage of Sum Insured, and Percentage of Loss.

- Secondly, when you access the policy insurance item, you can edit the **Excess Type** and **Excess Value** fields.



Premium Amount Local Currency	
Excess Type	Flat
Excess Value	10
Waiting Period	0

- Thirdly, the actual excess value for each peril as it is saved, is displayed in the **Covered Risk** section, as per below.



Covered Risk

COVERED RISK

Count Limit	
Covered Risk	Personal Accident
Current Count Limit	
Current Value Limit	
Excess Type	Flat
Excess Value	10
Insurance Risk	Personal Accident

Manage Policy Claim Data

Claim data is used in decision process when writing a new risk or when modifying existing risks during underwriting, adjusting indemnity limits if no automatic reinstatements are in place, or renewing a policy. It can determine a set of KPIs at policy level that can be further aggregated at portfolio levels.

The system is able to get claims data from an external source in order to be used in the views for Policy Claims data and other processes (e.g. Update Sum Insured after a claim is settled).

The Policy Claim Data API is used to log the policy claim data. The API is called each time an update is made at reserve level and/or payment level. Find out more about the API by accessing the Policy Claim Data API page.

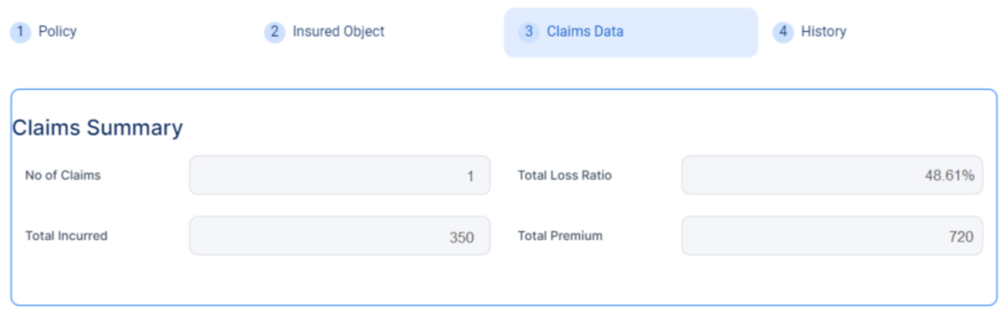
View the Loss Ratio

The loss ratio (LR) is calculated at coverage level, and takes into account the gross written premium (GWP).

In order to view the loss ratio, access a policy and click on the **Claims Data** tab.

The tab displays three sections:

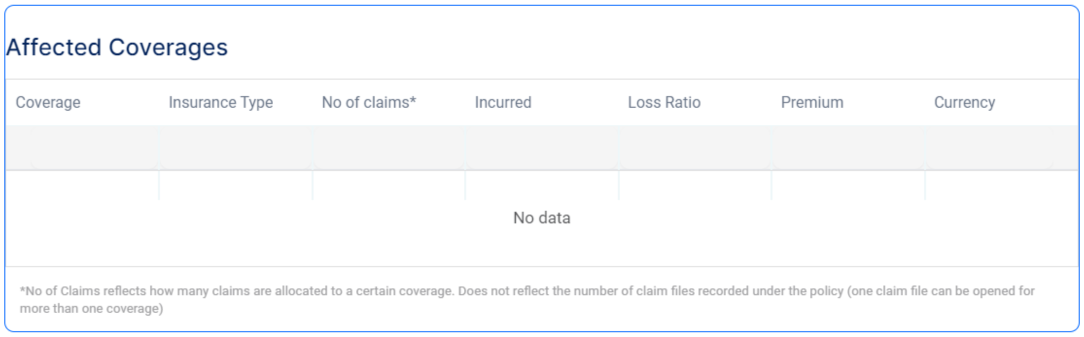
- **Claims Summary**: This section contains a grid with the following fields:
    - **No of Claims**: All claim files opened on the coverage;
    - **Total Loss Ratio**: Calculated as *Incurred Total/Premium Amount Total * 100*. It is displayed in percentages.
    - **Total Incurred**: Calculated as the sum of all incurred values for all coverages. The incurred is calculated as: *SUM (all resevres on the coverage) + SUM (all indemnities paid on that coverage)*.
    - **Total Premium**: Calculated as the sum of all the premium values for all coverages.
    - **Currency**.



Claims Summary

No of Claims	1	Total Loss Ratio	48.61%
Total Incurred	350	Total Premium	720

- **Affected Coverage**: Keeps and reflects the tabular view for claims recorded at coverage level.

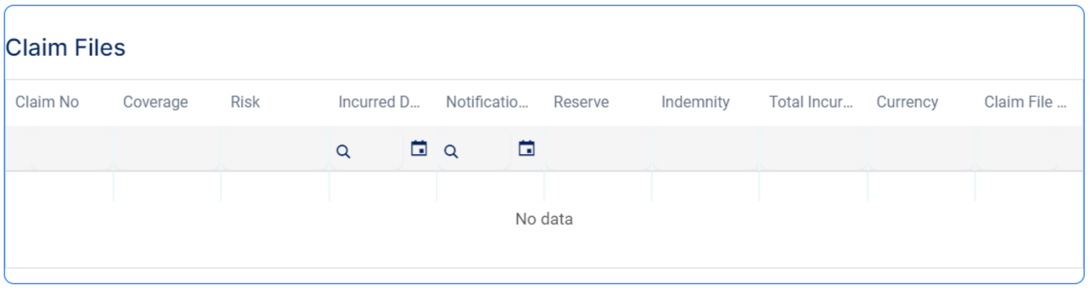

Affected Coverages

Coverage	Insurance Type	No of claims*	Incurred	Loss Ratio	Premium	Currency

No data

*No of Claims reflects how many claims are allocated to a certain coverage. Does not reflect the number of claim files recorded under the policy (one claim file can be opened for more than one coverage)

- **Claim Files**: This section displays a list with all the claim files recorded to that policy. The claim file data is updated after each API call for said claim file with the latest data. The values are updated each time the API is received.


Claim Files

Claim No	Coverage	Risk	Incurred D...	Notificatio...	Reserve	Indemnity	Total Incur...	Currency	Claim File ...

No data

View the Indemnity Limits Updates After a Claim

After a claim, the **Sum Insured** of the policy diminishes with a value corresponding to the amount paid for that loss, when the policy is not subject to an automated reinstatement of Sum Insured. In the Coverage view, the **Available Indemnity Limit** contains the updated **Sum Insured** after a loss is paid partially or final. The **Update Indemnity Limit** field is introduced to state if the indemnity limit needs to be updated from a policy after each paid claim. The default value is No. The first time a policy is saved, unless otherwise modified by the Formula Engine or an API, the **Available Indemnity Limit** is equal to the **Sum Insured/Indemnity Limit**. The update of the **Available Indemnity Limit** is triggered when at the Policy Claim Data level the following two rules apply at the same time, for a single claim:

- The claim reserve value is 0;
- The claim payment's value is greater than 0.

The source for this data is always `FTOS_INSPA_ClaimFiles` if the initial rule is validated updateSumInsuredafterClaim = True.

The **Available Indemnity Limit** in **Core Policy Admin** is equal to the **Indemnity Limit** in **Core Claims Admin**.

The **Indemnity Limit** consumes the recorded value limit and the risk level for the peril that has the claim file attached to it.

Indemnity Limit = MIN (Value Limit, Available Indemnity Limit).

The **Available Indemnity Limit** is calculated as per below:

Available Indemnity limit = Previous value for the Available Indemnity Limit - Current Indemnity Paid for the coverage.

In the remote case in which the result has a negative value, the updated **Available Indemnity Limit** becomes 0. In case the claims settles a total loss, the **Available Indemnity Limit** becomes 0. The values of the claims fees (e.g. legal expenses, handling costs, experts costa) are not included in the process of updating the **Sum Insured**.

Example:

At 01.01. 2021 a policy is issued with the **Sum Insured** for the building being 100.000 EUR, so all the coverages and the perils for the building have a sum insured of 100.000 EUR.

The content is insured at 20.000 EUR, so all the coverages and the perils for the content have a sum insured of 20.000 EUR.

TPL is covered for 10.000 EUR. So the coverages and the perils for TPL have an Indemnity limit of 20.000 EUR.

On the 27th of February a fire ensues that affects both the building and the content. The fire does not spread or affect any neighboring areas and no individuals other than the insured’s family is affected.

One month later, the file is settled and a claim of 30.000 EUR is paid, 25.000 EUR for the building and 5.000 EUR for the content.

If an agent searches for this policy on the 28th of March, the system displays the following values :

- **Available Indemnity Limit** for the Building and all subsequent coverages = 75.000 Eur;

- **Available Indemnity limit** for the Content and all subsequent coverages = 15.000 Eur;
- **Available Indemnity Limit** for TPL = 10.000 Eur.

View Insured Objects

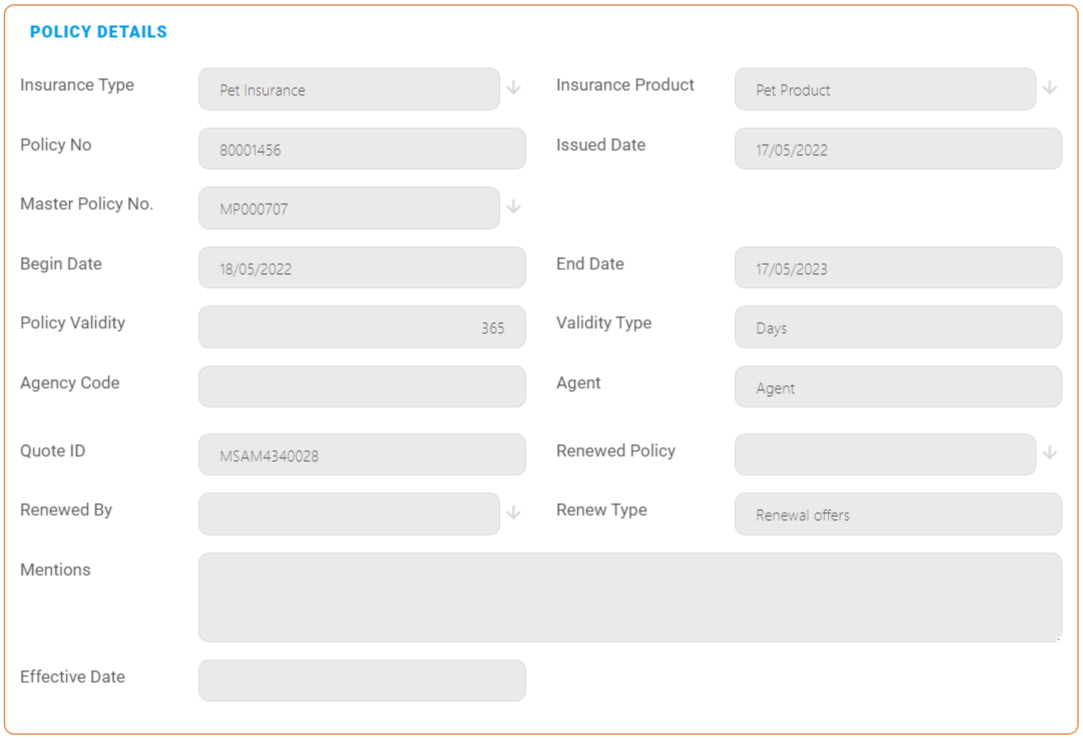
The system has the ability to save relevant data for each object covered by a policy, linked to a masterpolicy or stand alone.

The relevant data that combined can uniquely identify an object with its specific characteristics used to drive the correct price for the coverages offered for the pet's protection, represents the Insured Object Insurance Products.

NOTE
The Insured Object saved when a policy will be issued is the same as the insured object structure used in the Q&B journey.

An example of a pet insurance policy is given below. The policy presents the following sections:

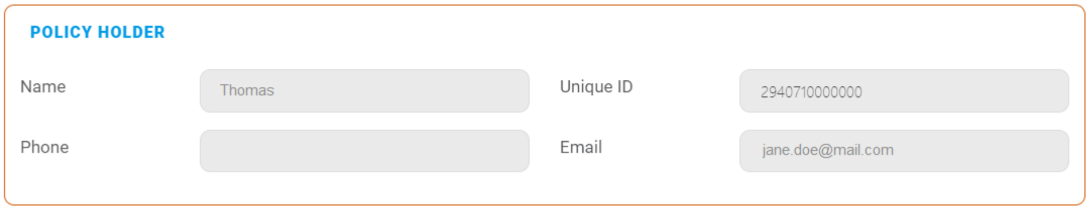
- The **Policy Details** section:



POLICY DETAILS

Insurance Type	Pet Insurance	Insurance Product	Pet Product
Policy No	80001456	Issued Date	17/05/2022
Master Policy No.	MP000707		
Begin Date	18/05/2022	End Date	17/05/2023
Policy Validity	365	Validity Type	Days
Agency Code		Agent	Agent
Quote ID	MSAM4340028	Renewed Policy	
Renewed By		Renew Type	Renewal offers
Mentions			
Effective Date			

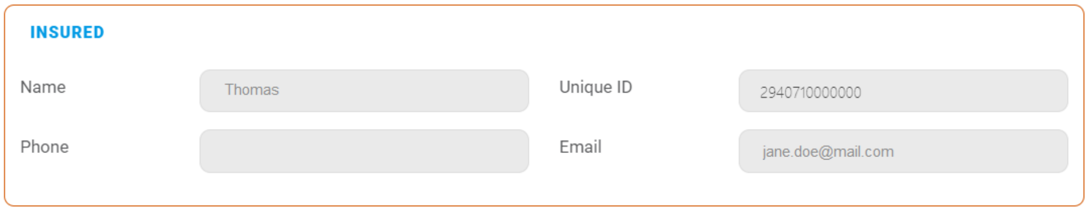
- The **Policy Holder** section:



POLICY HOLDER

Name	Thomas	Unique ID	2940710000000
Phone		Email	jane.doe@mail.com

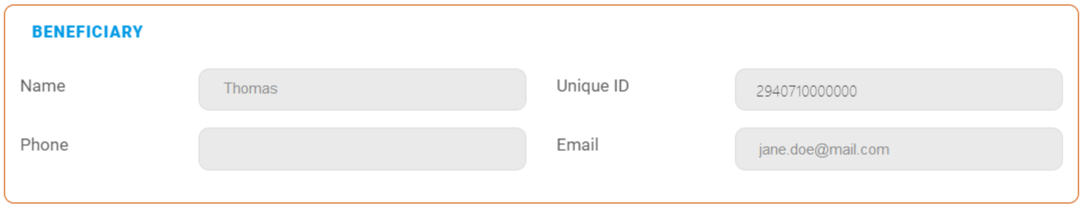
- The **Insured** section:



INSURED

Name	Thomas	Unique ID	2940710000000
Phone		Email	jane.doe@mail.com

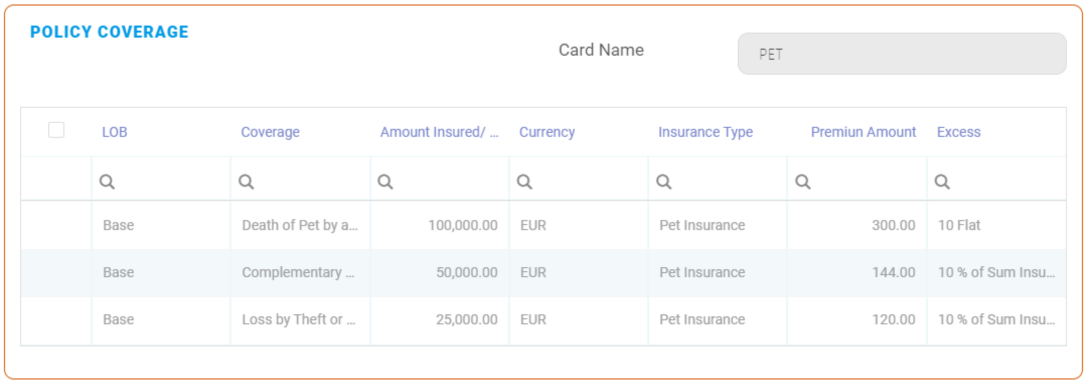
- The **Beneficiary** section:



BENEFICIARY

Name	Thomas	Unique ID	2940710000000
Phone		Email	jane.doe@mail.com

- The **Policy Coverage** section. The **Card Name** field displays what package is covered through the policy:

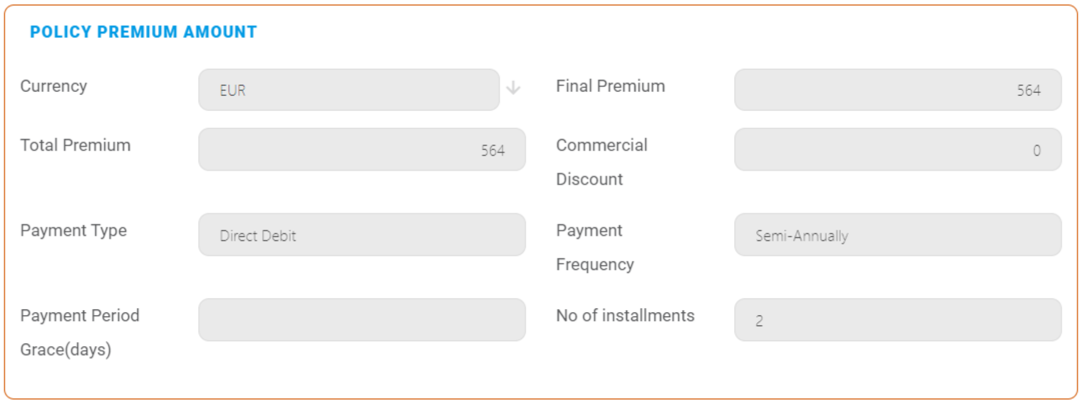


POLICY COVERAGE

Card Name: PET

LOB	Coverage	Amount Insured/ ...	Currency	Insurance Type	Premiun Amount	Excess
Base	Death of Pet by a...	100,000.00	EUR	Pet Insurance	300.00	10 Flat
Base	Complementary ...	50,000.00	EUR	Pet Insurance	144.00	10 % of Sum Insu...
Base	Loss by Theft or ...	25,000.00	EUR	Pet Insurance	120.00	10 % of Sum Insu...

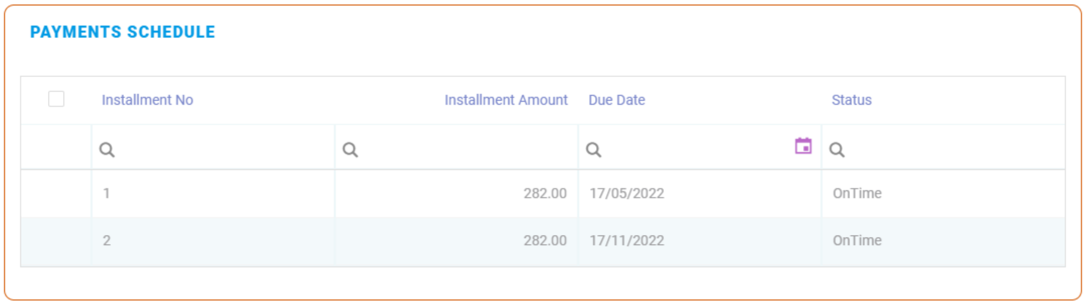
- The **Policy Premium Amount** section:



POLICY PREMIUM AMOUNT

Currency	EUR	Final Premium	564
Total Premium	564	Commercial Discount	0
Payment Type	Direct Debit	Payment Frequency	Semi-Annually
Payment Period Grace(days)		No of installments	2

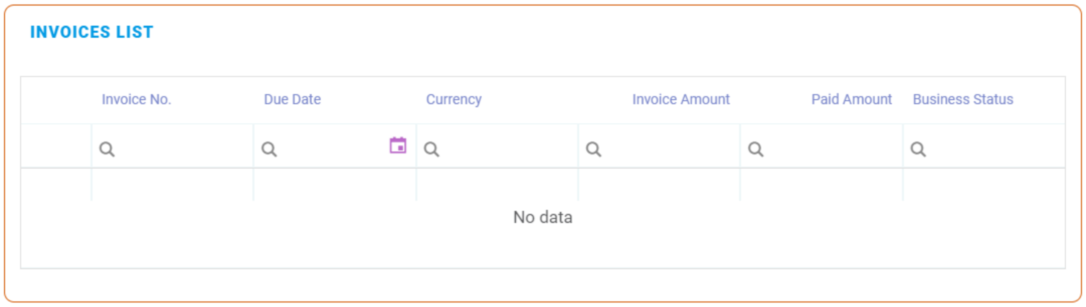
- The **Payments Schedule** section:



PAYMENTS SCHEDULE

Installment No	Installment Amount	Due Date	Status
1	282.00	17/05/2022	OnTime
2	282.00	17/11/2022	OnTime

- The **Invoices List** section:

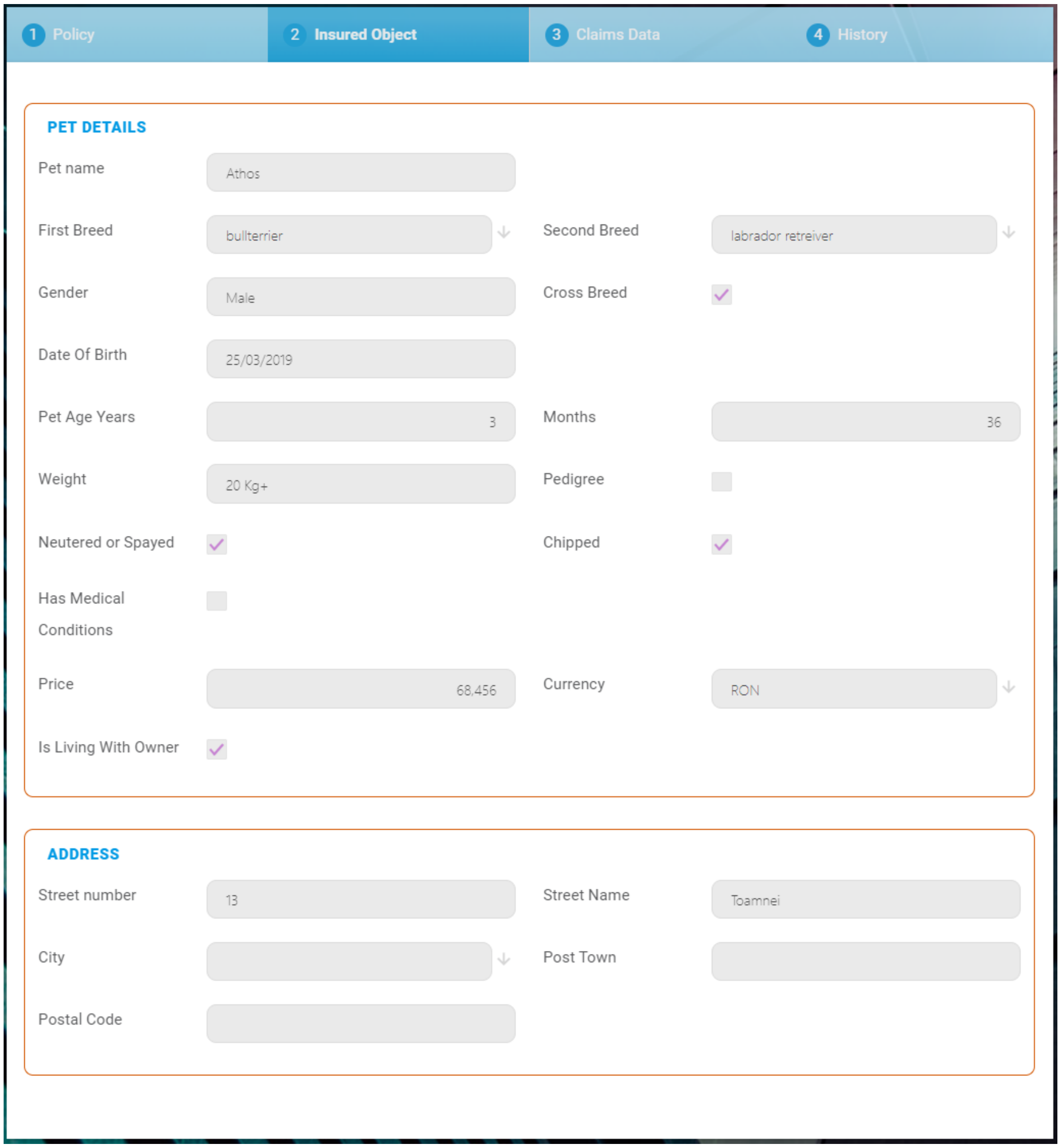


INVOICES LIST

Invoice No.	Due Date	Currency	Invoice Amount	Paid Amount	Business Status
No data					

At the bottom of the policy you can find the **Issue MTA** and **Cancel Policy** buttons. When clicked, the MTA, respectively the policy cancellation journeys are initiated.

Once you save an insurance policy (linked to a Masterpolicy or stand alone), the **Insured Object** tab is displayed, which contains both pet details and the insured's address, as shown below in a pet insurance policy.



PET DETAILS

Pet name	Athos		
First Breed	bullterrier	Second Breed	labrador retreiver
Gender	Male	Cross Breed	✓
Date Of Birth	25/03/2019		
Pet Age Years	3	Months	36
Weight	20 Kg+	Pedigree	
Neutered or Spayed	✓	Chipped	✓
Has Medical Conditions			
Price	68,456	Currency	RON
Is Living With Owner	✓		

ADDRESS

Street number	13	Street Name	Toamnei
City		Post Town	
Postal Code			

Master Policies

This functionality enables insurers to create their bundled policy offerings, covering General Insurance, Life & Health protection, but underlining the fact that the Mastepolicy has the distinct characteristic for being the way through which multi-object, multi-location is managed for Property & Casualty business or for example multi-person (group) contracts for Life & Health.

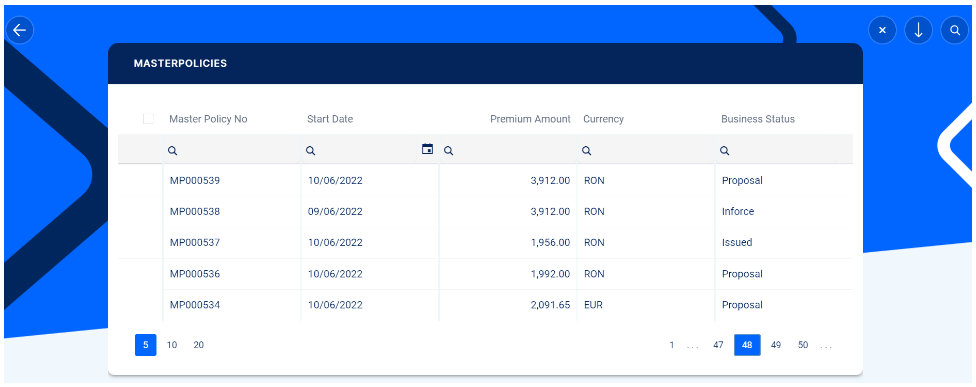
The Masterpolicy and its linked Policies have some common details. For example they share the same contractor, intermediary, currency, policy end date, number of installments, installments due dates, payment types.

Master Policy Generation

A Masterpolicy is issued by using the GenerateMasterpolicyAPI:

- When generating a Masterpolicy with no policy attached, it is automatically set in the **Draft** status, with several editable fields described further;
- When generating a Masterpolicy with its attached policy at the same time, it is automatically set in the **Proposal** status, and its fields are read-only.

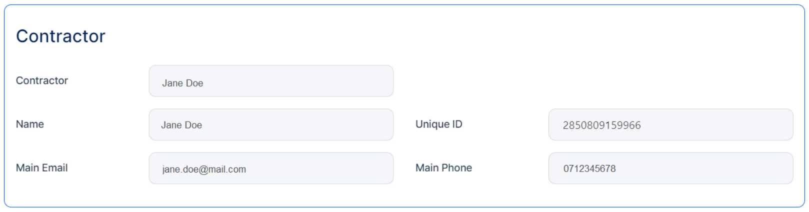
1. To view the list with all the existing masterpolicies in **FintechOS Portal**, navigate the following menu: **Policy Admin > Masterpolicies**.
2. In the **Masterpolicies** subsection, the **Master Policies List** grid is displayed, which contains all the created master policies.


MASTERPOLICIES

Master Policy No	Start Date	Premium Amount	Currency	Business Status
MP000539	10/06/2022	3,912.00	RON	Proposal
MP000538	09/06/2022	3,912.00	RON	Inforce
MP000537	10/06/2022	1,956.00	RON	Issued
MP000536	10/06/2022	1,992.00	RON	Proposal
MP000534	10/06/2022	2,091.65	EUR	Proposal

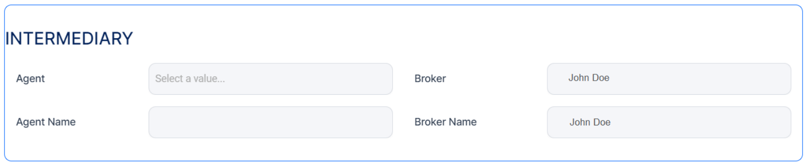
3. Open a Masterpolicy and view the displayed tabs. In case of a Masterpolicy in **Draft** status, you can edit some of the fields, as presented below:

   - The **Master Policy** tab contains the following sections:
     - **Contractor**: Choose a **Contractor** from the drop-down to automatically fill in the fields with their details.


Contractor

Contractor	Jane Doe		
Name	Jane Doe	Unique ID	2850809159966
Main Email	jane.doe@mail.com	Main Phone	0712345678

     - **Intermediary**: Only one of the **Agent** or **Broker** name fields can be filled. If you try to fill both, the following message is displayed: "A Master Policy can have either an Agent or a Broker".

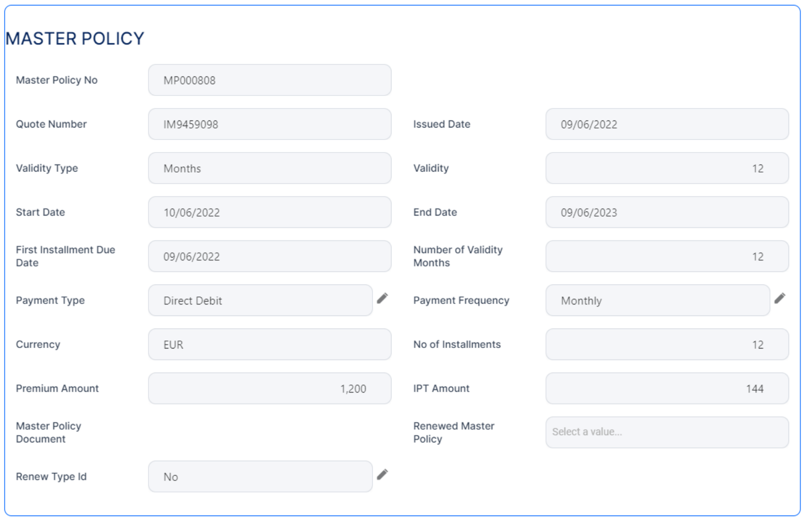

INTERMEDIARY

Agent	Select a value...	Broker	John Doe
Agent Name		Broker Name	John Doe

     - **Master Policy**, containing the following fields:
       - Master Policy No: The unique number of the master policy;

- Quote Number: The unique number of the quote. The field is editable in the **Draft** status;
- Validity Type: Drop-down field with the following possible values: Days, Months, or Years. The field is editable in the **Draft** status;
- Validity: The validity of the master policy expressed in the number of days, months, or years. The field is editable in the **Draft** status;
- Start Date: The start date of the policy. The field is editable in the **Draft** status;
- End Date: The end date of the policy;
- Payment Type: Choose from the following possible values: Bank Transfer or PayU. The field is editable in the **Draft** status;
- Payment Frequency: Choose from the following possible values: annually, semiannually, quarterly, monthly. The field is editable in the **Draft** status;
- Currency: The currency used for the policy;
- No of Installments: The number of installments for the policy;
- Premium Amount: The amount of the premium;
- IPT Amount: The amount of the insurance premium tax;
- Master Policy Document: The master policy file to be uploaded;
- Renewed Master Policy: Option set to choose the master policy to be renewed. The field is editable in the **Draft** status;

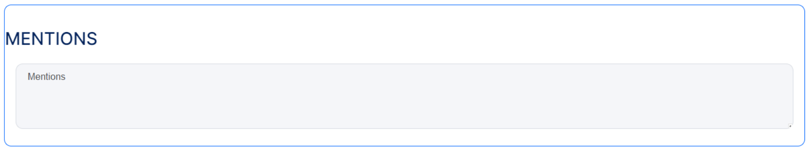
- Renew Type Id: Drop down field to choose the renew type ID, the possible values being None, No, Automatic Renewal and Renewal Offers. The field is editable in the **Draft** status.



MASTER POLICY

Master Policy No	MP000808		
Quote Number	IM9459098	Issued Date	09/06/2022
Validity Type	Months	Validity	12
Start Date	10/06/2022	End Date	09/06/2023
First Installment Due Date	09/06/2022	Number of Validity Months	12
Payment Type	Direct Debit	Payment Frequency	Monthly
Currency	EUR	No of Installments	12
Premium Amount	1,200	IPT Amount	144
Master Policy Document		Renewed Master Policy	Select a value...
Renew Type Id	No		

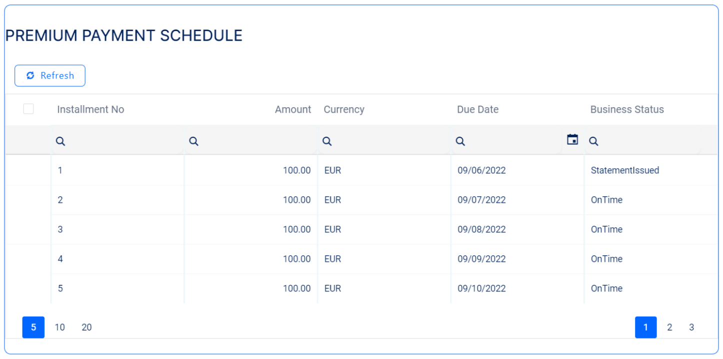
- **Mentions**, where you can add some free text, with mentions regarding the Master Policy. You cannot edit this field after the offer is made.



MENTIONS

Mentions

- Fill in the mandatory details and save the record. Two more tabs are displayed in the form:
  - The first tab, first called **Master Policy** and now called **Master Policy Summary**, containing the above fields, plus, below them, 2 new sections called **Premium Payments Schedule** and **Invoices**.
    - These sections are empty at first, and populated with values after the associated policies are generated:
      - The **Premium Payments Schedule** section, the list with the installments:



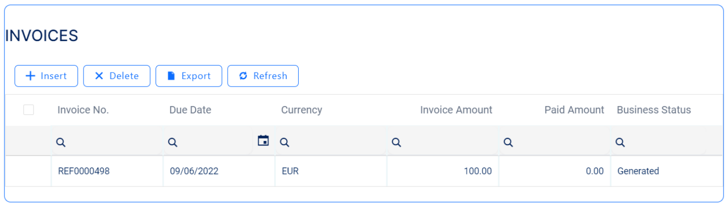
PREMIUM PAYMENT SCHEDULE

Refresh

Installment No	Amount	Currency	Due Date	Business Status
1	100.00	EUR	09/06/2022	StatementIssued
2	100.00	EUR	09/07/2022	OnTime
3	100.00	EUR	09/08/2022	OnTime
4	100.00	EUR	09/09/2022	OnTime
5	100.00	EUR	09/10/2022	OnTime

5 10 20 1 2 3

- The **Invoices** section, the list with the statements:

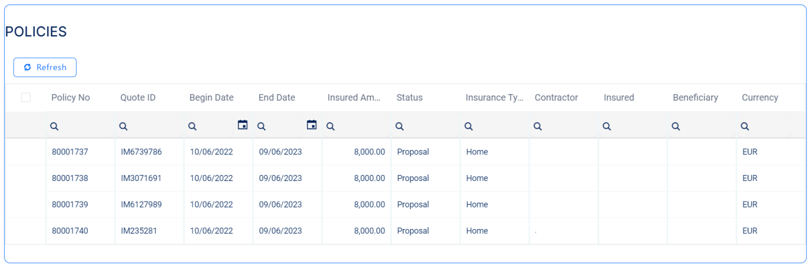


INVOICES

Insert Delete Export Refresh

Invoice No.	Due Date	Currency	Invoice Amount	Paid Amount	Business Status
REF0000498	09/06/2022	EUR	100.00	0.00	Generated

- The second tab, called **Policies**, is empty at first, and is populated with values after the associated policies are generated through the PolicyGenerationAPI endpoint. This tab contains 2 grids:
  - The **Policies** grid, as shown below:

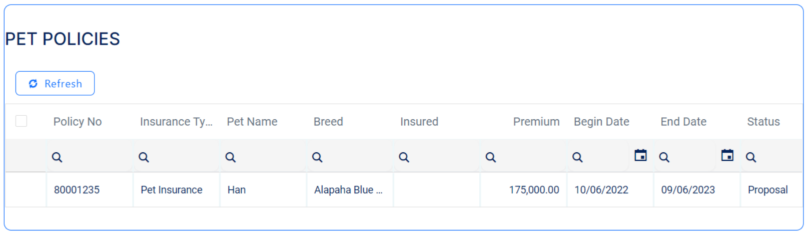


POLICIES

Refresh

Policy No	Quote ID	Begin Date	End Date	Insured Am...	Status	Insurance Ty...	Contractor	Insured	Beneficiary	Currency
80001737	IM6739786	10/06/2022	09/06/2023	8,000.00	Proposal	Home				EUR
80001738	IM3071691	10/06/2022	09/06/2023	8,000.00	Proposal	Home				EUR
80001739	IM6127989	10/06/2022	09/06/2023	8,000.00	Proposal	Home				EUR
80001740	IM235281	10/06/2022	09/06/2023	8,000.00	Proposal	Home				EUR

  - The **Pet Policies** grid, as shown below:



PET POLICIES

Refresh

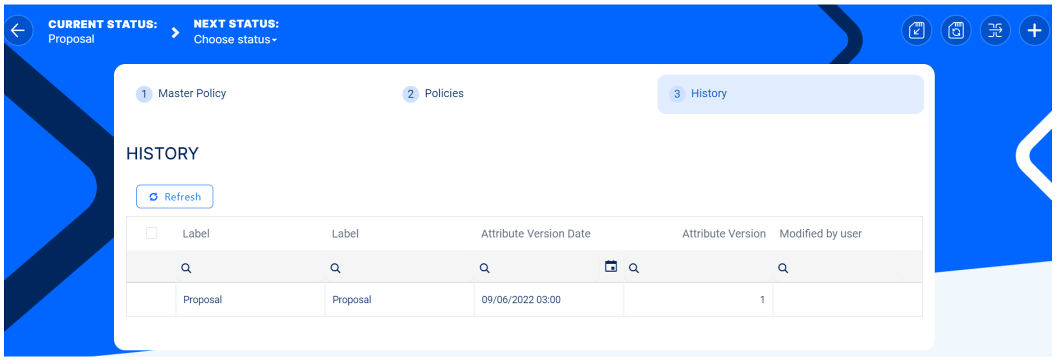
Policy No	Insurance Ty...	Pet Name	Breed	Insured	Premium	Begin Date	End Date	Status
80001235	Pet Insurance	Han	Alapaha Blue ...		175,000.00	10/06/2022	09/06/2023	Proposal

All the above fields are mandatory, except Renewed Master Policy and the **Mentions** section.

Master Policies Versioning History

A **History** tab is available for each Master Policy with all the versions related to it, in order to keep track of the version number, starting from 1.

The **History** tab is displayed for the first time when the Master Policy is in the **Proposal** status, and the first version status of the Master Policy displayed in the **History** tab is **Proposal**.



HISTORY

	Label	Label	Attribute Version Date	Attribute Version	Modified by user
	Proposal	Proposal	09/06/2022 03:00	1	

After the Master Policy is issued, the version displayed in the **History** tab is **Issued**. The previous version is not displayed anymore. When the Master Policy is in **Inforce** status, this is the version displayed in the **History** tab. The previous versions are not displayed.

When a version is approved, the previous version is closed , having the **Version Closed** status. This automatically starts with the effective date of the approved version. Unapproved versions are also displayed in the **History** tab. There cannot be 2 or more **Draft** or **Pending** versions at the same time. Also, there cannot be a **Pending** and a **Draft** version at the same time.

The **History** tab contains a grid with the following columns: Label, Attribute Version Date, Attribute Version, Modified by User.

Perform Master Policy Mid Term Adjustments

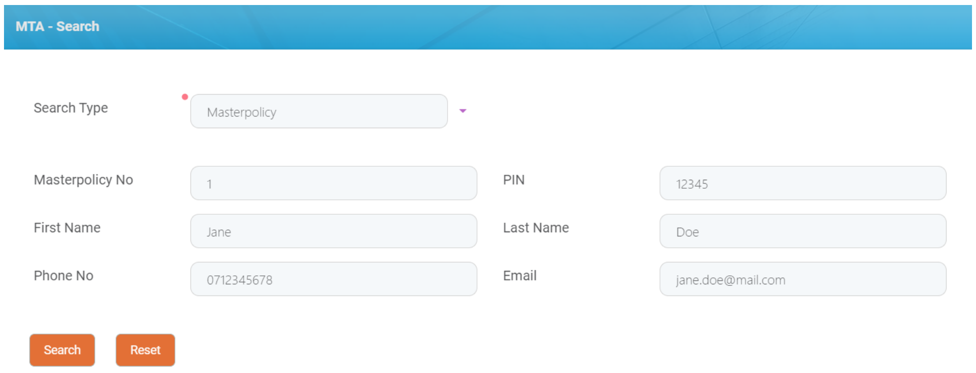
Masterpolicy MTA's comprise of a collection of alterations that apply in the same way for all the policies that are part of said Masterpolicy. These MTA types are: Change frequency, Change payment type, Change due date.

You can make alterations either to a Masterpolicy or to a single policy that is part of a Masterpolicy.

Search for Masterpolicies

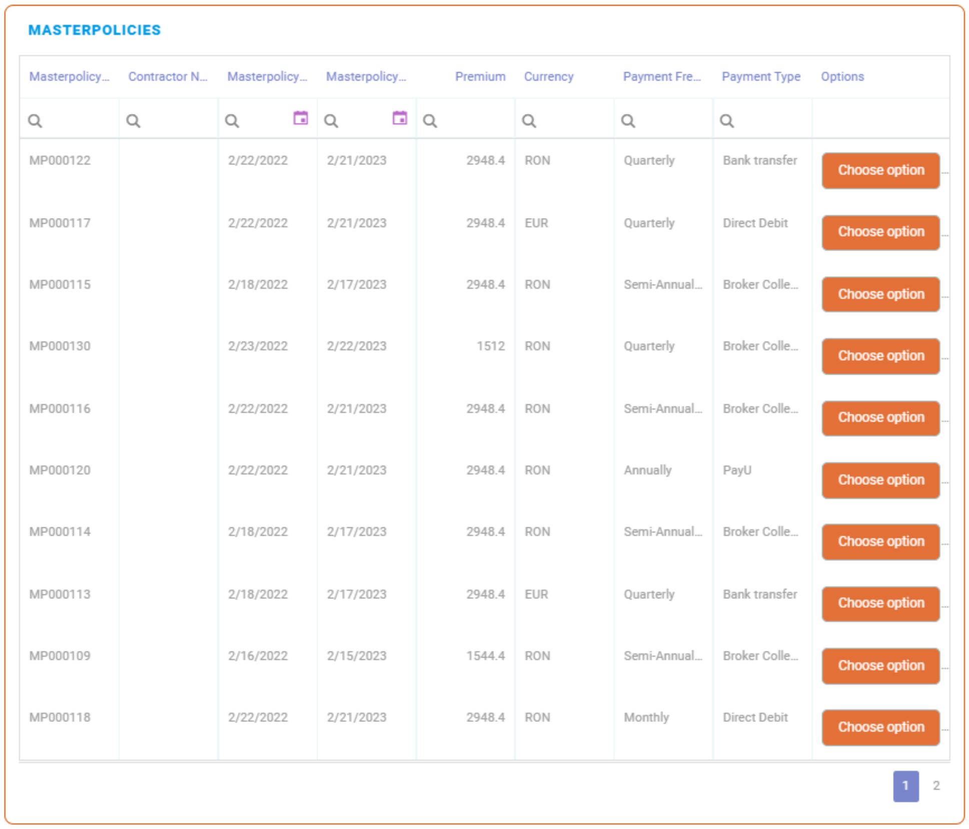
Prior to opening a change policy request, you have to look for the masterpolicy on which the MTA Masterpolicy adjustments needs to made. The Masterpolicy searching form is the first step of opening an MTA.

1. Choose from the **Policy Admin** menu, the **Masterpolicy Alterations** menu entry. This opens the specific MTA view with the following columns: Request No, Notification Date, Master Policy No, Contractor Name, Policy Alteration Type, MTA No, Request Status.
2. By double-clicking a record from the view, the system opens the existing change Masterpolicy request in its specific form and status. Otherwise, a new change Masterpolicy request is registered by clicking the **+** button above the MTA view, which opens the **MTA - Search** form.
3. In the **Search Type** field, you can select to alterate either a Masterpoilcy or a single policy that is part of a Masterpolicy.


MTA - Search

Search Type	Masterpolicy		
Masterpolicy No	1	PIN	12345
First Name	Jane	Last Name	Doe
Phone No	0712345678	Email	jane.doe@mail.com

4. Click **Search**, and the **Masterpolicies** grid unfolds, as shown below:

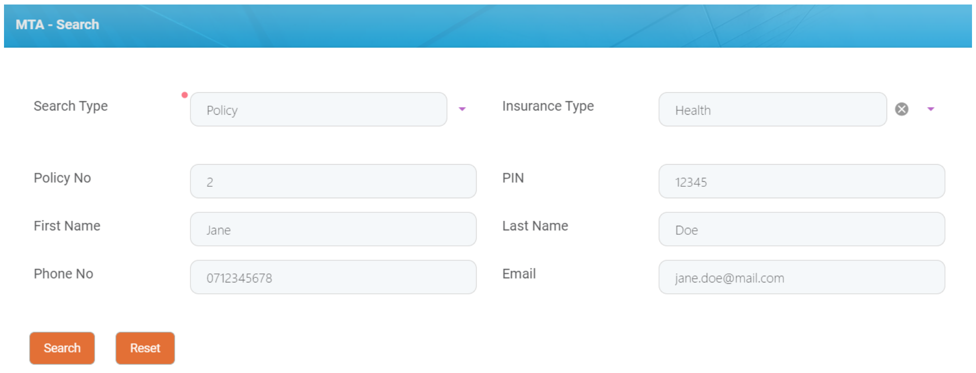


MASTERPOLICIES

Masterpolicy...	Contractor N...	Masterpolicy...	Masterpolicy...	Premium	Currency	Payment Fre...	Payment Type	Options
MP000122		2/22/2022	2/21/2023	2948.4	RON	Quarterly	Bank transfer	Choose option
MP000117		2/22/2022	2/21/2023	2948.4	EUR	Quarterly	Direct Debit	Choose option
MP000115		2/18/2022	2/17/2023	2948.4	RON	Semi-Annual...	Broker Colle...	Choose option
MP000130		2/23/2022	2/22/2023	1512	RON	Quarterly	Broker Colle...	Choose option
MP000116		2/22/2022	2/21/2023	2948.4	RON	Semi-Annual...	Broker Colle...	Choose option
MP000120		2/22/2022	2/21/2023	2948.4	RON	Annually	PayU	Choose option
MP000114		2/18/2022	2/17/2023	2948.4	RON	Semi-Annual...	Broker Colle...	Choose option
MP000113		2/18/2022	2/17/2023	2948.4	EUR	Quarterly	Bank transfer	Choose option
MP000109		2/16/2022	2/15/2023	1544.4	RON	Semi-Annual...	Broker Colle...	Choose option
MP000118		2/22/2022	2/21/2023	2948.4	RON	Monthly	Direct Debit	Choose option

Alternaitvely, choose to modify a single policy belonging to a Masterpolicy.

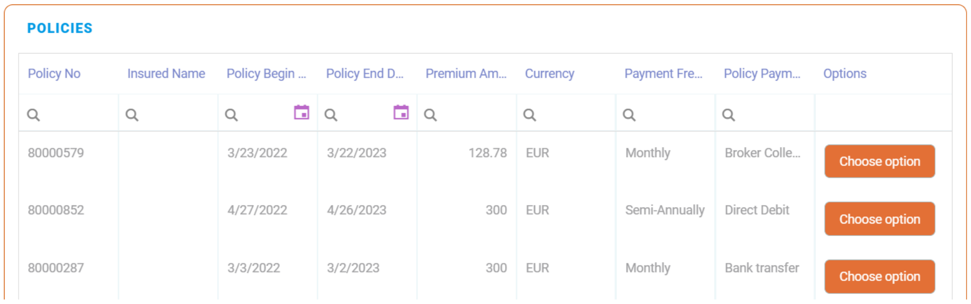
1. In the **MTA - Search** form, from the **Search Type** drop down, choose the Policy value.



MTA - Search

Search Type	Policy	Insurance Type	Health
Policy No	2	PIN	12345
First Name	Jane	Last Name	Doe
Phone No	0712345678	Email	jane.doe@mail.com

2. Click **Search**, and the **Policies** grid unfolds, containing the policies with the matching criteria.



POLICIES

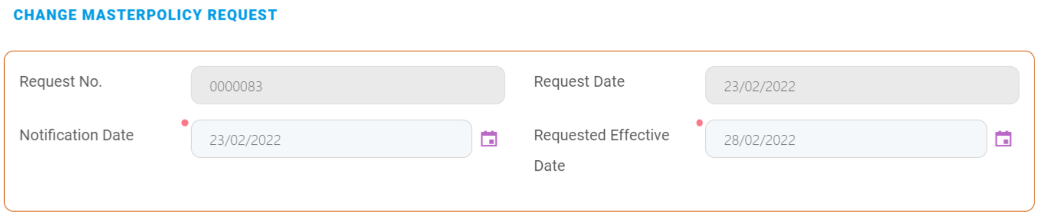
Policy No	Insured Name	Policy Begin ...	Policy End D...	Premium Am...	Currency	Payment Fre...	Policy Paym...	Options
80000579		3/23/2022	3/22/2023	128.78	EUR	Monthly	Broker Colle...	Choose option
80000852		4/27/2022	4/26/2023	300	EUR	Semi-Annually	Direct Debit	Choose option
80000287		3/3/2022	3/2/2023	300	EUR	Monthly	Bank transfer	Choose option

Request a Mid Term Adjustment

After searching for a specific Masterpolicy the process of completion for the open change Masterpolicy request can start.

The first two sections on the **Change Masterpolicy Request** tab are related to general information regarding the request:

- **Masterpolicy No.** instead of **Policy No.**;
- Without beneficiary and insured;
- The **Paid Amount** is the total of payments for all policies included in a Masterpolicy.



CHANGE MASTERPOLICY REQUEST

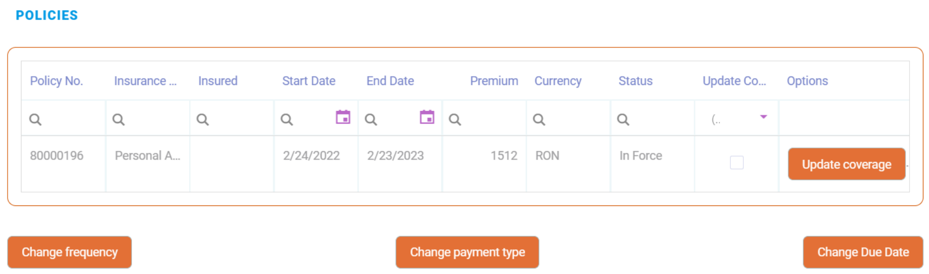
Request No.	0000083	Request Date	23/02/2022
Notification Date	23/02/2022	Requested Effective Date	28/02/2022

Register Alterations

Register Masterpolicy Alterations

After completing the general information regarding the change policy request, you have the possibility to choose from the available policy alternation types, a type of modification.

The **Update Coverage** button is displayed on each policy line from the **Policies** grid in the interface.



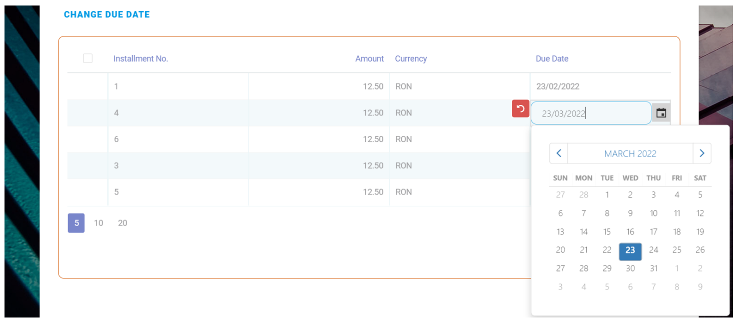
Policy No.	Insurance ...	Insured	Start Date	End Date	Premium	Currency	Status	Update Co...	Options
80000196	Personal A...		2/24/2022	2/23/2023	1512	RON	In Force	☐	Update coverage

Based on your chosen option, the specific selection is displayed. Multiple alteration types are permitted in the same MTA, so you can expand multiple selections at the same time, complete and validate them in the desired order.

After you have made all the modifications regarding the payment and/or the frequency, and validated them, all the policies related to the Master policy update their **Payment Type** according to the new value selected in the **Master Product**.

Change Due Date

1. Click the **Change Due Date** button. A grid is expanded, that contains the payment schedule associated with the Masterpolicy. The payment schedule for the MTA only contains the columns up to **Due Date**.
2. Update the due dates that you want to change inline, within the schedule. You can modify one or more installments' due dates. Only the **Due Date** can be edited, the other fields are read only.


CHANGE DUE DATE

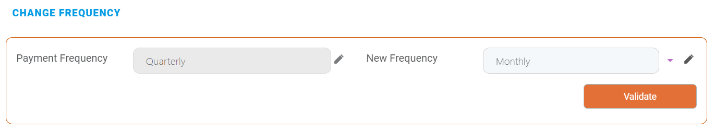
Installment No.	Amount	Currency	Due Date
1	12.50	RON	23/02/2022
4	12.50	RON	23/03/2022
6	12.50	RON	
3	12.50	RON	
5	12.50	RON	

3. Click the **Validate** button in order to save and validate the changes.

The due date in the installments grid on the Masterpolicy are updated after you click the **Register** button and reach the **Effective Date** set on the alteration request.

Change Frequency

1. Click the **Change Frequency** button. The **Change Frequency** grid expands, containing the **Payment Frequency** and **New Frequency** fields.
2. Choose the **New Frequency** from the field drop-down. This contains the same values as for new payment frequency existing on the policy alterations.


CHANGE FREQUENCY

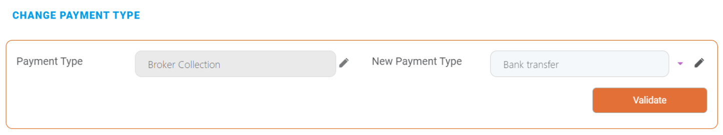
Payment Frequency: Quarterly
New Frequency: Monthly

3. Click the **Validate** button.

The **Payment Frequency** field is retrieved by default from the Masterpolicy and is not editable. The **New Payment Frequency** field is the new payment frequency selected.

Change Payment Type

1. Click the **Change Payment Type** button. The **Change Payment Type** grid expands, containing the **Payment Type** and **New Payment Type** fields.
2. Choose the **New Payment Type** from the field drop-down.



3. Click the **Validate** button.

The **Payment Type** field is retrieved by default from the Masterpolicy and is not editable.

After you make the modifications, the **Masterpolicy Alteration Summary** section is available with the following details:

Field Name	Description
Effective Date	Defaulted with the **Requested Effective Date** from the first section. Editable and mandatory. Effective Date >= Policy Begin Date.
End Date	Not editable and mandatory.
Additional Premium	Not editable and mandatory. The additional premium is calculated. For the discounted amount, a negative amount is displayed upon clicking the **Register** button.
Currency	Not editable and mandatory.
Policy alteration type	Not editable and mandatory. The policy alteration type is determined. If at least one alteration included requires the issuance of an MTA, the value is the MTA needed upon clicking the **Register** button.
MTA No	Not editable and mandatory. Sequencer automatically generated.
Policy mentions	Editable and not mandatory.

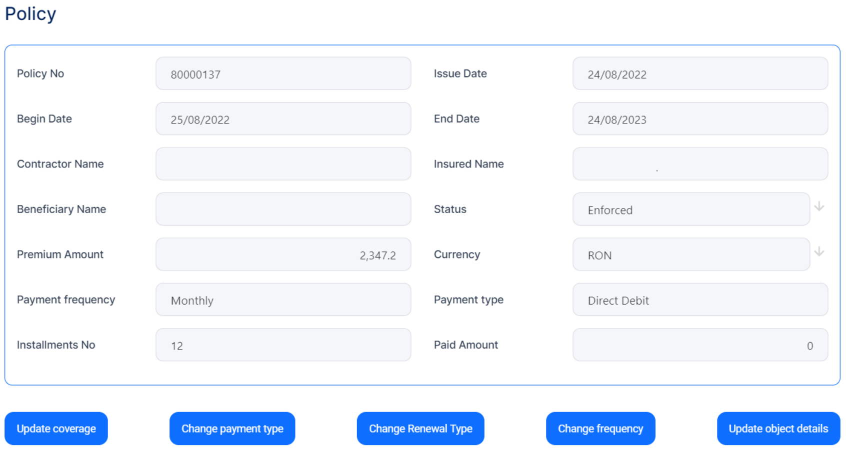
Field Name	Description
Comments	Editable and mandatory only if you cancel the request by clicking the **Cancel** button form the bottom of the page.

After reviewing the request made, you have the possibility to either **Cancel** or **Register** the request.

Register Policies Alterations

Similarly to registering altertions on a Masterpolicy, after in the **MTA - Search** section you have selected the intended policy, you are redirected to said policy, in the form for the MTA used for a single policy.

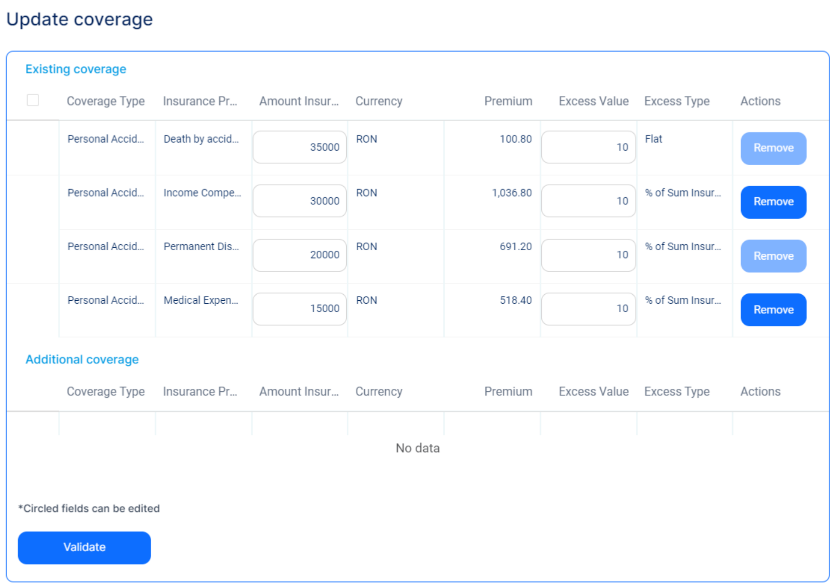
1. Choose from the available policy alternation types, a type of modification. The following buttons are available:



Policy

Policy No	80000137	Issue Date	24/08/2022
Begin Date	25/08/2022	End Date	24/08/2023
Contractor Name		Insured Name	
Beneficiary Name		Status	Enforced
Premium Amount	2,347.2	Currency	RON
Payment frequency	Monthly	Payment type	Direct Debit
Installments No	12	Paid Amount	0

2. Click **Update coverage**, and the **Update coverage** grid expands, where you can edit the **Amount Insured** and **Excess Value**.


Update coverage

Existing coverage

Coverage Type	Insurance Pr...	Amount Insur...	Currency	Premium	Excess Value	Excess Type	Actions
Personal Accid...	Death by accid...	35000	RON	100.80	10	Flat	Remove
Personal Accid...	Income Compe...	30000	RON	1,036.80	10	% of Sum Insur...	Remove
Personal Accid...	Permanent Dis...	20000	RON	691.20	10	% of Sum Insur...	Remove
Personal Accid...	Medical Expen...	15000	RON	518.40	10	% of Sum Insur...	Remove

Additional coverage

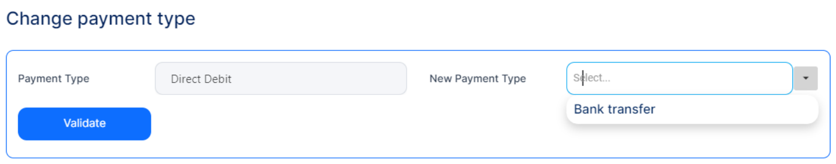
Coverage Type	Insurance Pr...	Amount Insur...	Currency	Premium	Excess Value	Excess Type	Actions

No data

*Circled fields can be edited

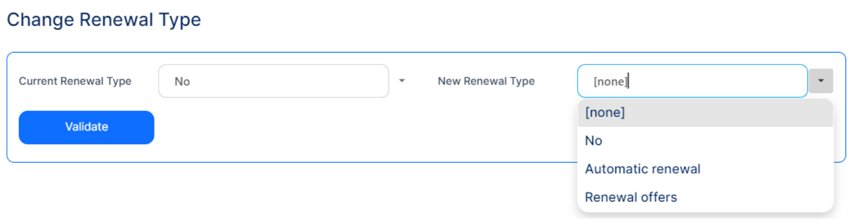
Validate

3. Click **Validate**, and the policy is now updated with the new Amount Insured And Excess Value.
4. Click **Change payment type**, and the **Change payment type** grid is unfolded, where you can select a new payment type, as per below.

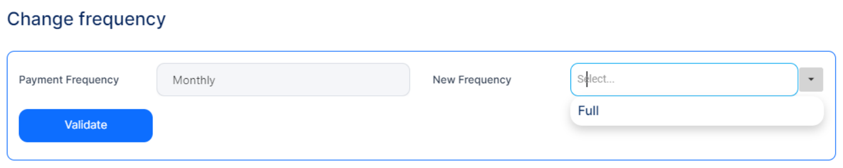

Change payment type

Payment Type	Direct Debit	New Payment Type	Select...

5. Click **Validate** to save the changes, and the policy is updated with the new payment type.
6. Click **Change Renewal Type**, and the **Change Renewal Type** grid is unfolded, where you can select if you want the policy to be automatically renewed, or not. You can also change the type of renewal.



7. Click **Validate**, and the policy is updated with the new renewal type.
8. Click **Change frequency**, and the **Change frequency** grid is unfolded, where you can select a new payment frequency, as per below.



9. Click **Validate**, and the policy is updated with the new payment frequency type.

Total Premium Adjustments

Each policy has a premium adjustment that can have a positive or a negative value. You can operate multiple coverage adjustments for multiple related policies.

In the **Policy Alteration Summary** section, a grid is displayed that only shows the related policies that had a change in premium for the current master MTA flow that you have initiated. The following fields are displayed: Insured, Policy No, Insurance Product Item, Premium Adjustment Value, Adjustment Type.

The **Master MTA Premium Adjustment** field is also displayed, showing the sum total of the Premium Adjustment Values for all the policies displayed in the grid, and affected by the current master MTA through coverage update.

Update Coverage

Through this MTA, you are able to update coverages for a specific policy included in a masterpolicy. This means that for each policy included in the policies grid, the **Update Coverage** grid is displayed. By clicking this grid, you are redirected to a new Policy MTA request for the selected policy.

The new Policy MTA is opened in **Draft** status. This contains all the information regarding the already completed request with the same information as for the Masterpolicy request, without the possibility to adjust these information. Also, the user is able to already see the **Update Coverage** grids centered.

All the initial fields needed in the normal Policy MTA take over the values already collected and available in the Master policy, such as the **Request Date**, **Notification Date**, and **Request Effective Date**. The **Request No** is automatically incremented.

The affected policies take into account the Masterpolicy MTA number.

The workflow for the **Update Coverage** grid is the following:

- This section allows you to **Edit**, **Remove** or **Add** coverages on the current policy;
- For the first grid, by clicking the **Edit**/**Remove** button:
    - The **Edit** action makes the **Amount Insured/Indemnity Limit** field available;
    - The **Remove** action removes the item from the existing coverage grid, and it is available in the **Additional Coverage** grid.
- For the second grid, you have the possibility to add a new coverage for the current policy:
    - The **Amount insured/Indemnity Limit** for existing coverages on product level is equal to the value set on product factory, but if a new coverage is added, the **Amount insured/Indemnity Limit** is editable;
    - The **Amount Insured/Indemnity limit** field is mandatory for adding a new coverage;

- When adding the **Amount Insured/Indemnity limit** for new coverages, the **Premium Amount** for each new coverage is auto completed in the grid

NOTE
By updating the **Amount Insured/Indemnity Limit** when **Editing/Removing/Adding** an insurance coverage, the **Premium Amount** and **Premium Percentage** on the policy is also updated according to the modifications. The update is triggered when registering the request.

The **Return to Masterpolicy MTA** button redirects you to the **Master Policy MTA Change Request** view. This button is displayed instead of the **Register** button on the actual Policy MTA request.

Beside it, the **Cancel** button is also displayed in the form.

- Click the **Return to Masterpolicy MTA** button to be redirected back to the Masterpolicy MTA request. The status for the Policy MTA request is changed from **Draft** to **Registered**.
- Click the **Cancel** button, to change the Masterpolicy's status from **Draft** to **Cancelled**. Once cancelled, you can re-initiate the update coverage alteration flow on a Masterpolicy MTA. A validation checks the existing Policy MTA request opened in **Draft** status from a Masterpolicy MTA. If it is already an existing one, then you are redirected to that one, otherwise a new one is registered.

All changes made in the policy MTA, even if not validated, are saved, and when you return to finish the flow or modify, update the changes done, you don't have to input the data again.

In case an coverage update done at policy level is not validated, you cannot validate the overall Master policy MTA. If this case occurs, the system displays the following error message “There is still an MTA pending for policy no. <'policy no of the MTA that is not validated;>”.

The **Update Coverage Open** column marks if for the correlated policy an update coverage alteration request is opened or not:

- If the value is Yes (checked), then an existing update coverage request is already opened by clicking the near **Update Coverage** button, and you are redirected to the existing **Draft**, **Registered**, **Accepted Update Coverage** request;
- If the value is No (not checked), then the policy does not have an update coverage request opened on it by clicking the near **Update Coverage** button, and you are redirected to a new update coverage request in **Draft** status.

By clicking the **Update Coverage** button for a policy, the **Update Coverage Open** is marked and remains checked until the update coverage request is **Cancelled** or **Declined**.

- **Update Coverage Open** checked = request in **Draft**, **Registered** or **Accepted** status;
- **Update Coverage Open** not checked = request in **Cancelled** or **Declined** status.

Accept/Decline a Masterpolicy MTA Request

You are able to approve or decline a MTA request according to the customer's decision regarding the modifications on the policy.

On the **Change Master Policy Request** tab, beside the displayed policy information, you can find the **Accepted** and **Decline** buttons which trigger its specific status transitions. These buttons are displayed only for change policy requests which are in **Registered** status.

1. When you click the **Accepted** button the following actions are triggered:
    - The status transition is made from the **Registered** to the **Accepted** status;
    - The **MTA No** is updated from the first tab in the **Master Policy Alteration Summary**;

- The transition specific for policy versioning is made to the **Approved** policy version.

2. When you click the **Declined** button the following actions are triggered:
    - The status transition is made from the **Registered** to the **Declined** status;
    - The transition specific for policy versioning is made to the **Unapproved** policy version.

Change the Masterpolicy Renewal Type

Core Policy Admin gives you the ability to change the renewal type that is set for a Masterpolicy, through an MTA.

1. Select an **Alteration Type** and the subsequent **Masterpolicy** to be modified.
2. Click the **Change Renewal Type** button. The section in which you can select the new renewal type is displayed. The **New Renewal Type** option set displays any of the available renewal types within the system.
3. Select the new renewal type for said Masterpolicy.
4. The flow for any MTA is applied (e.g. register, accept/decline, impact in the Masterpolicy's status).
5. The **Renewal Type** recorded on the all the policies related to said Masterpolicy display the newest renewal type option selected through the approved Masterpolicy MTA.

NOTE
When a Change Renewal Type MTA is accepted and the alteration flow is finished, the Masterpolicy Automatic Renewal process takes into account, for said policy, the new renewal type and ignores what is set at product level.

The rest of the configurationss for renewal at main product level continue to apply (i.e. Renewal Validity, Renewal Tariff, No of Days Before Renewal, Renewal Validity)

Manage Master Policy Versioning

There are cases, like Master Policy renewal when you need to create a new version of a specific Master Policy. To accommodate the versioning functionality, from an insurance business perspective, the **Core Policy Admin** solution uses a combination of the **FintechOS**standard versioning process mechanism and custom development. Consequently, you use the FTOS_VersioningHelper client side library and follow the standard procedure when configuring version settings, version settings items and entity settings. However, for policy versioning, you use the **FTOS_VersioningHelper_ Edit** client side library in order to keep some attributes **Read Only**, even when the policy is in **Version Draft** business status, with **isEditable** option enabled.

Master Policy Versioning Process Description

1. Access an existing Master Policy and click the **Insert** button, called **New Version**.
2. An edit form of that Master Policy opens, with a new version in the **Version Draft** status, so you are able to change and update the information related to that Master Policy.
    - When the Edit form opens, the **Master Policy Summary** tab is automatically selected.
    - Both tabs, **Master Policy Summary** and **Policies List**, are displayed.
3. The status of the newly added version of the Master Policy is **Version Daft** and the next status is also displayed, **Version Unapproved**.

Once the edit form opens, you are able to change the following:

- The values from any field of the **Master Policy** section, except the **Master Policy No** and the **Currency** fields. The **Master Policy No** remains the same like the one before versioning;
- the text from the **Mentions** section;
- the values from the **Premium Payment Schedule** section.
    - You are able to only update the installments in the **On Time** status;
    - Once the values here change and a new version of the Master Policy is added, a new version is added for the **Premium Payment Schedule** section.

NOTE In the manual versioning phase, the changes made to the Master Policy are not subject to validation. Also no changes are triggered for the associated policies. This is done in a later phase and through an MTA process at Master Policy level for example.

After the desired changes are made in the **Version Draft** status, you are able to:

- Unapprove this version, by manually switching the status of the version from the top left corner;
- Approve the new version by filling in the **Effective Date** of the version and clicking the **Approve** button.
    - The effective date is available at the end of the Master Policy form in the **Master Policy Summary** section, and it is mandatory if you want to approve a version;
    - The **Effective Date** cannot be sooner than the beginning date of the Master Policy;

Automatically Renew a Master Policy

A renewal in insurance is the continuation of a coverage for a specified period. For the Masterpolicy a series of successive developments are planned to be made in order to provide the capability to renew all policies that are linked to a Masterpolicy, within one action. There are several ways to renew a contract, for example, to automatically generate an offer for renewal or generate for example a new policy, prior to the expiry date of the existing one.

As a principle, the Masterpolicy is automatically renewed according to the received renewal configurations in order to continue providing coverage for the insured. A renewal offer for the Masterpolicy and the linked policies is issued, once the insured period for the renewed Masterpolicy has passed.

The **Main Product** (identified through Quote Config ID) + **Renewal Type** (from Gen Master Policy API if available) defines the renewal configurations to be applied for a specific Masterpolicy.

The Masterpolicies are subjected to the automatic renewal process taking into consideration the renewal configurations available at main product level: renewing policy, renewal validity, renewal tariff and number of days before renewal. Depending on the option in the **Renewal Type**, a certain Masterpolicy renewal process is triggered.

As a first step for the Masterpolicy renewal, a method of issuing the data needed for the Renewal Offer (based on existing policy data) has been rolled out.

As such for each policy under a masterpolicy, distinct JSON objects are created containing info about the each renewed policy and tariff calculated adequately (same/actual tariff).

NOTE
After a Mastepolicy renewal each Mastepolicy renewal policy has the Masterpolicy number filled in. The same behavior applies for the related policies, related to renewed policies. Each renewal Masterpolicy only contains policies that had been renewed.

Automatic Renewal Types

A daily scheduled job is running in order to find all the policies from the system which have to be renewed according to their **End Date** and the parameter set for the number of days before renewal.

The rules for a Masterpolicy are given in the table below.

Renewal Type	Renewal Validity	Renewal Tariff	Renewing Policy
Automatic Renewal	**Same validity**	**Actual tariff + Same tarif**	**New policy**
	Renewal Masterpolicy - ProposalValidity = Renewed policy validity Start Date = Renewed Masterpolicy End date +1 End date = Start Date & Validity Same Contractor as renewed Masterpolicy Same Intermediary data as renewed Masterpolicy Same Quote Same Payment type & Payment Freq Same Currency Reflect Renewed Policy no (previous contract) Same Mentions. The tariff logic does not influence the Masterpolicy. The Masterpolicy premium is the SUM of the related policies premiums.		

The rules for the policies related to a Masterpolicy are given in the table below.

Renewal Type	Renewal Validity	Renewal Tariff
Automatic Renewal	**Same validity**	**Actual tariff**
	Renewed policy number Same Product and Insurance Type Validity - Renewed Policy validity Begin Date = Renewed policy End Date + 1 End Date is calculated according to the Start date + Validity Same parties Same insured object Same agent/broker and distribution channel Same payment type and the same payment frequency Same coverages and indemnity limits The total premium has to be calculated based on the currently approved product version	
Automatic Renewal	**Same validity**	**Same tariff**
	Renewed policy number Same Product and Insurance Type Validity - Renewed Policy validity Begin Date = Renewed policy End Date + 1 End Date is calculated according to the Start date + Validity Same parties Same insured object Same agent/broker and distribution channel Same payment type and the same payment frequency Same coverages and indemnity limits The total premium has to be calculated based on the product version valid at the issuance date of the renewed policy	

Change the Masterpolicy Renewal Type

Core Policy Admin gives you the ability to change the renewal type that is set for a Masterpolicy, through an MTA.

1. Select an **Alteration Type** and the subsequent **Masterpolicy** to be modified.
2. Click the **Change Renewal Type** button. The section in which you can select the new renewal type is displayed. The **New Renewal Type** option set displays any of the available renewal types within the system.
3. Select the new renewal type for said Masterpolicy.

4. The flow for any MTA is applied (e.g. register, accept/decline, impact in the Masterpolicy's status).
5. The **Renewal Type** recorded on the all the policies related to said Masterpolicy display the newest renewal type option selected through the approved Masterpolicy MTA.

NOTE
When a Change Renewal Type MTA is accepted and the alteration flow is finished, the Masterpolicy Automatic Renewal process takes into account, for said policy, the new renewal type and ignores what is set at product level.

The rest of the configurations for renewal at main product level continue to apply (i.e. Renewal Validity, Renewal Tariff, No of Days Before Renewal, Renewal Validity).

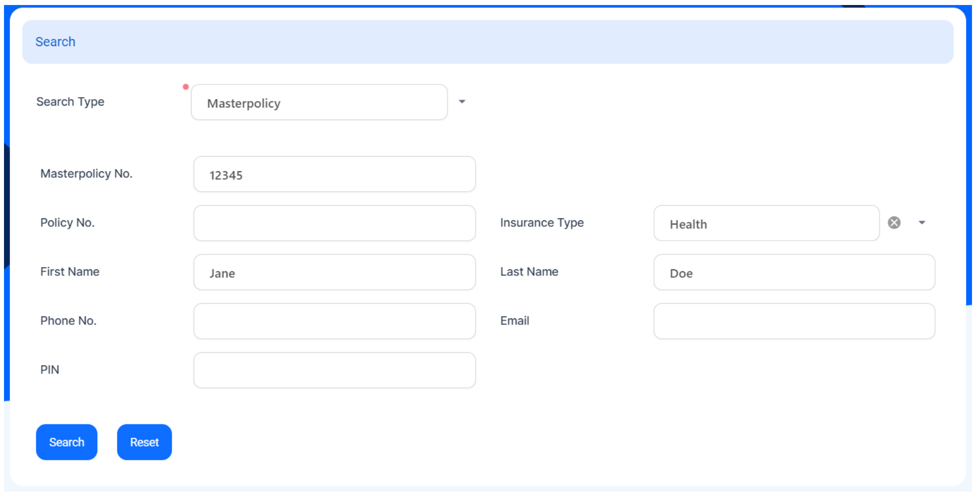
Cancel a Master Policy

The Masterpolicy cancellation works in a similar way as the masterpolicy’s MTA. This means that cancelling a Masterpolicy inherently cancels all the policies with status **In Force** MTA that are linked to it.

Register a Master Policy Cancellation

In order to open a cancellation on a Masterpolicy, you must first search for a specific one. After finding the right masterpolicy, you can request a cancellation.

1. In the main menu, navigate to **Policy Admin > Cancellations**.
2. Click the **Insert** button. The **Masterpolicy Search** form is displayed. Select the Masterpolicy option from the drop down. You can further refine the search by filling at least one of the fields in the grid.


Search

Search Type	Masterpolicy		
Masterpolicy No.	12345		
Policy No.		Insurance Type	Health
First Name	Jane	Last Name	Doe
Phone No.		Email	
PIN			

3. Click the **Search** button. A grid is displayed, showing the Masterpolicies with the matching details inserted in the previous form.

   In this grid, only Masterpolicies having the specific statuses configured in the correlated processor appear. The statuses available for the Masterpolicy cancellation are either in the **Inforce** or **Issued** status.

4. When you've found the right policy, click the **Choose Option** button.

5. The main **Cancellation** form is displayed in order to register a request.

   If you choose a policy which already has a Cancellation request in the **InProgress** status, the process is not available, and a warning message is displayed.

6. When starting the registration, two tabs are displayed in the interface: **Change Request** and **Masterpolicy Details**.

   - The **Change Request** tab presents the following 3 sections:
     - The **Change Request Summary** section, with the following fields to be filled in: Cancellation Notification Date, Reason Type, Requested End Date, Final End Date.

       The algorithm for the **Final End Date** and **Requested End Date** fields according to the selected **Reason Type** is the following, taking into consideration the validations stated above:

Reason Type	Requested End Date	Final End Date
Withdrawal	Automatically completed with the **Masterpolicy Start Date**.	Automatically completed with the **Masterpolicy Start Date**.
Property sold	The **Requested End Date** can be set to any day before or including the day of the request for cancellation.	The **Final End Date** is filled with the value from the **Requested End Date** but with the possibility to edit.

Reason Type	Requested End Date	Final End Date
Cancelled by Client	You can manually fill the field with any date.	If the **Notification Date** - **Policy (Issued) Date** <= X days (free withdrawal limit date), then this field is automatically completed with the **Policy Begin Date** with the possibility to be adjusted. Otherwise, this is automatically completed with the **Notification Date** + 21days to go. **NOTE** If the **Final End Date** is in the X days (from the free qithdrawal limit date), then the policy is transitioned to

Reason Type	Requested End Date	Final End Date
		Withdrawal on client request, and if it is after those X days, then the policy is transitioned to the **Cancelled** status.
Decline by screening	You can manually fill the field with any date.	It can be filled with the value from the **Requested End Date**, but with the possibility to edit.
Cancelled	You can manually fill the field with any date.	It is automatically completed with the **Notification Date** + 21 days to go.

Reason Type	Requested End Date	Final End Date
Closed by Claim	It is pre-filled with the **Notification Date** + 21 days from the present on.	It is filled with the value from the **Requested End Date**, but with the possibility to edit.
Insured Death	You can manually fill the field with any date.	It is filled with the value from the **Requested End Date**, but with the possibility to edit.

- The **Masterpolicy** section, with containing read-only fields in regards to the Masterpolicy details. A specific grid with all the policies included in that Masterpolicy is displayed below the mentioned fields. Double-click a record to be redirected to that policy form.
- The **Payment Beneficiary** section, where you need to fill in the fields in regards to the payment beneficiary.

- The **Masterpolicy Details** tab displays a grid with all the Masterpolicy **Installments** and a grid with all the Masterpolicy **Claims**. The records cannot be adjusted.

7. After completing all the information, click the **Register** request button, to transition the status for the request to the **In Progress** status.
8. The **Premium Returned** tab is displayed, containing more information about the reimbursement.
9. Alternatively, click the **Cancel** button to close the current request. You have to complete the **Resolution Reason** attribute in order to explain the reason of cancelling the current request.
10. The request status is transitioned to the **Cancelled** status.

Propose a Masterpolicy Cancellation Request

If in the Masterpolicy searching grid you choose a policy which already has a Cancellation request in **InApproval** status, the process is not available and the following warning message is displayed: ”This Masterpolicy already has a Cancellation request opened on it!”.

After registering a Masterpolicy cancellation, the **Premium Returned** tab is displayed and you either have the possibility to **Propose** the request further to be approved, or to **Cancel** it.

The **Premium Returned** tab contains some static information regarding the premium to be returned. The **Returned Premium Amount** section contains the premium returning details for the Masterpolicy and a specific policy grid with all these details but for each policy.

The following fields are displayed in read-only mode for this section: Masterpolicy Start Date, Masterpolicy End Date, Premium Amount, Premium Currency, Paid Amount, Paid Currency, Returned Premium Amount, Returned Premium Currency.

After checking the information, you are able to perform the following actions:

- Further propose the request by clicking the **Propose Change Request** button which changes the status to **InApproval** and triggers the displaying of the **Request Approval** tab.
- Cancel the request, and you have to fill the **Resolution Reason** attribute in order to explain the reason of cancelling the current request. Also, the request status is transitioned to **Cancelled**.

NOTE
By clicking one of the existing buttons, the first tab becomes read-only with no other possibilities for adjustments. The buttons disappear after clicking them.

Policy Configurations

The **Core Policy Admin** module keeps a traceability during the life period of an insurance contract and its adjustments through time. Check the following pages to find out more about how this solution works:

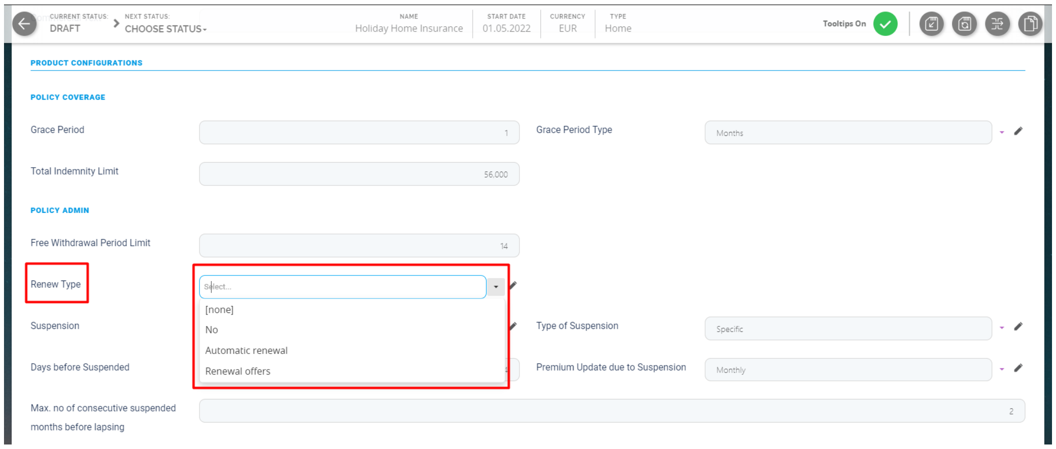
- Configuring Policy Renewals - for details about how to configure the policy renewal offers.
- Configuring Cancellation Reason Types - for details about how to configure the cancellation reason types.
- Working With APIs - for details about the following APIs:
    - Policy Generation API - for details about the Policy Generation API.
    - Generate Master Policy API - for details about how to generate a Master Policy.
    - Get Policy Data API - for details about the Get Policy Data API.
    - Policy Status Change API - for details about the Policy Status Change API.
    - Policy Claim Data API - for details about logging the policy claim data into the system.
- Working With Journeys - for details about the following journeys:
    - Mid Term Adjustment Journeys - for details about the mid term adjustment journeys.
    - Policy Cancellation Journeys - for details about the policy cancellation journeys.
    - Other Automated Journeys - for details about other automated journeys.
- Flow Parameters and Scheduled Jobs - for details about the flow parameters and scheduled jobs.
- Core Policy Admin Formulas - for details about the Core Policy Admin Formulas.

Configuring Policy Renewals

Perform the following configurations for policy renewals:

Configuring Policy Automatic Renewal

In order to configure whether the contracts under a specific Insurance Product should be automatically renewed or not, you must set the **Renew Type** on the Insurance Product level.



CURRENT STATUS:	NEXT STATUS:	NAME	START DATE	CURRENCY	TYPE
DRAFT	CHOOSE STATUS	Holiday Home Insurance	01.05.2022	EUR	Home

PRODUCT CONFIGURATIONS

POLICY COVERAGE

Grace Period	1	Grace Period Type	Months
Total Indemnity Limit	56,000		

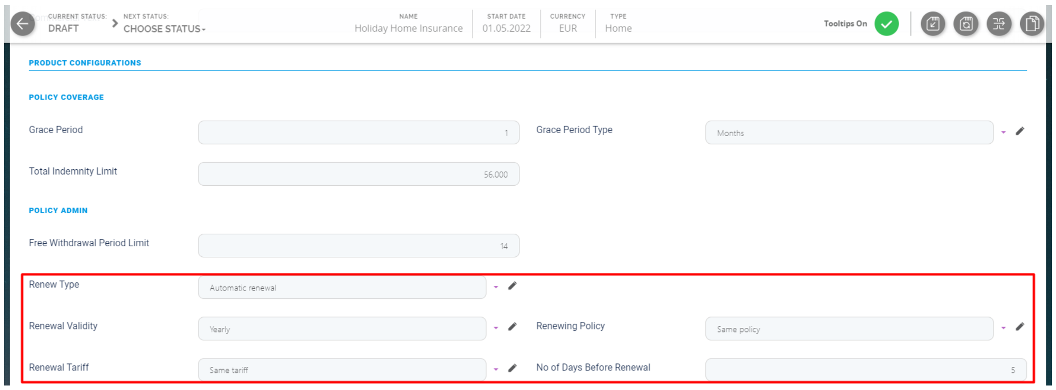
POLICY ADMIN

Free Withdrawal Period Limit	14		
Renew Type	Select...		
Suspension		Type of Suspension	Specific
Days before Suspended		Premium Update due to Suspension	Monthly
Max. no of consecutive suspended months before lapsing	2		

You may choose between:

- No - no renewal applied for the current product and subjected insurance policies;
- Automatic renewal - the process of automatic renewal is applied according to some extra configurations triggered by this option (see below);
- Renewal offers - a renewal offer is generated as a JSON object for the policies which have to be renewed under the current product.

If you configure your product to Automatic renewal, then you have to fill out the following fields:



CURRENT STATUS: DRAFT | NEXT STATUS: CHOOSE STATUS | NAME: Holiday Home Insurance | START DATE: 01.05.2022 | CURRENCY: EUR | TYPE: Home | Tooltips On

PRODUCT CONFIGURATIONS

POLICY COVERAGE

Grace Period	1	Grace Period Type	Months
Total Indemnity Limit	56,000		

POLICY ADMIN

Free Withdrawal Period Limit	14		
Renew Type	Automatic renewal		
Renewal Validity	Yearly	Renewing Policy	Same policy
Renewal Tariff	Same tariff	No of Days Before Renewal	5

Renewal Validity - the new validity which is applied for the correlated contracts of current product:

- Yearly - the new validity is set for 12 months (a year) in order to be automatically renewed next year, so that the new End Date of the renewed policy is the Start Date + Policy validity where the value is retrieved according to Renewal validity;
- Monthly - the new validity is set for x month in order to be renewed, so that the new End Date of the renewed policy is the Start Date + Policy validity where the value is retrieved according to Renewal validity;
- Same Validity - the new validity is calculated as for the previous policy to be renewed, so that the new End Date of the renewed policy is the Start Date + Policy validity where the value is retrieved according to Renewal validity; also, for this calculation, the values retrieved from the old policy are kept in the Policy Validity and Validity type attributes.

Example 1

For the previous policy: Policy Validity = 365 and Validity type = days

For the new policy: Start Date = Renewed policy End Date + 1 and End Date = Start date + 365 days - 1 day

Example 2

For the previous policy: Policy Validity = 13 and Validity type = months

For the new policy: Start Date = Renewed policy End Date + 1 and
End Date = Start date + 12 months - 1 day

Renewing Policy - troughout the automatic renewal process, policies can be renewed as new policy records in the system, keeping the old policy records with a final status, or they can be renewed by creating a new version of the same old policy records:

- Same policy - policy new version, keeping the same record in the database;
- New policy - brand new policy as new record in the database.

Renewal Tariff - the calculation of renewed policies can be either kept as for the previous policy that have been renewed or it can be calculated according to the current insurance product tariff version

- Same tariff - keeping the same tariff as for the old policy which have been renewed without taking into consideration the current insurance product tariff version;
- Actual tariff - taking into account the current insurance product tariff version and calculate the premium amount with the new values configured on product level.

IMPORTANT!
In order to take into account the new insurance product tariff configuration, the insurance product must be an approved version!

No. of Days Before Renewal - this stores the number of days before renewal, more precisely, with how many days before the policy End Date, a policy should be renewed; the automatic renewal scheduled job looks after this value in order to renew the policies which meet this renewal condition x days before End Date.

Example

If No. of days before renewal = 2, the policies after which the automatic renewal scheduled job looks after are the policies with

Policy End Date (old policy Initial End Date) - No. of days before renewal = Current Date and applies the renewal process to them

15.01.2022 (old policy Initial End Date) - 2 days (No. of days before renewal) = 13.01.2022 (Current Date) → the policies with this End Date is renewed.

NOTE
Policy Initial End Date <= Current Date + x days before renewal and Policy Initial End Date > Current Date

Configuring Renewal Offers

If at product level, the **Renewal Type** is set to Renewal offer, then the policies found to be renewed generate a JSON object as an insurance renewal offer containing information about the new policy renewed.

```
{
"startDate": "2022-12-19",
"validity": 12.0,
"validityType": "Months",
"issuedDate": "2021-12-17",
"totalIndemnityLimit": null,
"isRenewal": true,
"renewedPolicyId": "3bdfb0b6-5a91-48ac-854f-
59e5e1650622",
"mentions": "Insert comment here",
"quoteNo": "User0063",
"noOfRenewals": 1.0,
"insuranceTypeName": "Personal Accidents",
"productCode": "PA",
"agent": {
"agentId": null,
"agentType": "Individual person"
},
"broker": {
```

```
"brokerId": null,
"distributionChannel": null
},
"contractor": {
"uniqueIdentifier": "1911110223344",
"firstName": "Praslea",
"lastName": "NoMail",
"type": "Other"
},
"insured": {
"uniqueIdentifier": "1911110223344",
"firstName": "Praslea",
"lastName": "NoMail",
"type": "Other"
},
"beneficiary": {
"uniqueIdentifier": "1911110223344",
"firstName": "Praslea",
"lastName": "NoMail",
"type": "Other"
},
"currency": "RON",
"paymentType": "brokerCollection",
"paymentFrequency": "monthly",
"renewedPolicyNo": "80001342",
"insuranceProductItemList": [
{
"code": "MEACC",
"insuredAmount": 20000.0,
"finalPremiumAmount": 691.2
},
{
"code": "ICPA",
"insuredAmount": 25000.0,
"finalPremiumAmount": 792.0
},
{
"code": "PDA",
"insuredAmount": 35000.0,
"finalPremiumAmount": 1209.6
},
{
"code": "DPA",
"insuredAmount": 50000.0,
"finalPremiumAmount": 144.0
```

```
    }
  ]
}
```

Configuring Cancellation Reason Types

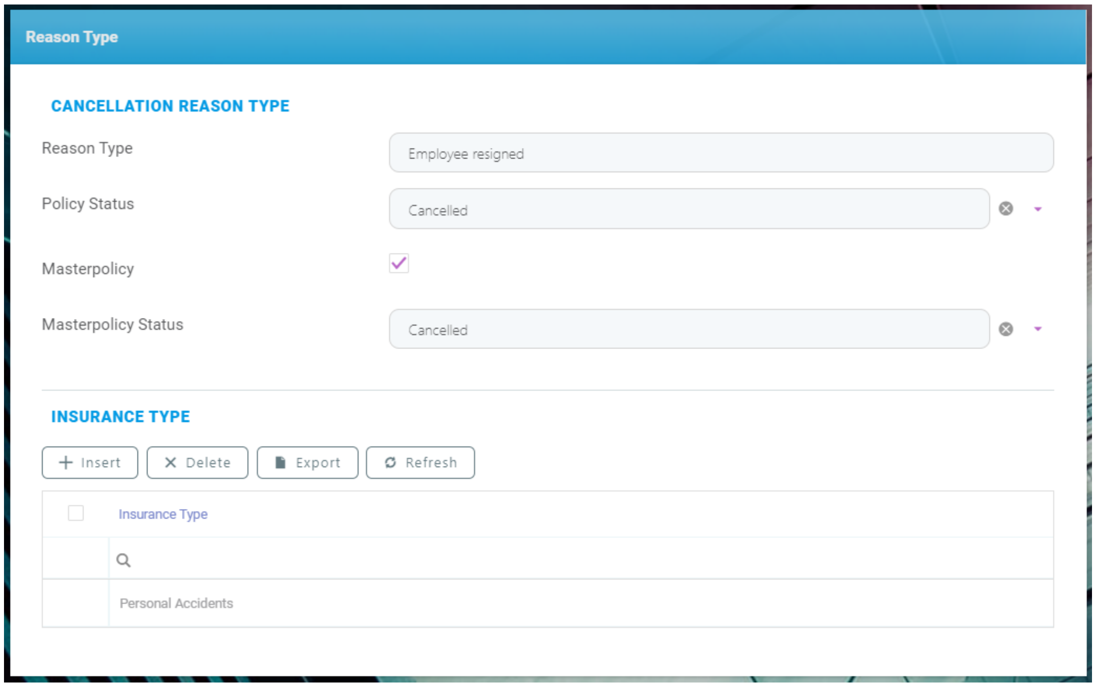
Core Policy Admin offers you the ability to configure if a specific cancellation reason type should be included as a reason type for a Masterpolicy beside the policy.

1. In **FintechOS Portal**, navigate to **Settings > Cancellation Reason Types**.
2. The **Cancellation Reason Types** grid is displayed. Insert a new reason type by clicking the **Insert** button.
3. Fill in the **Reason Type** and **Policy Status** fields.
4. Check the **Masterpolicy** option and choose the **Masterpolicy Status** from the drop down field.

   The **Masterpolicy** boolean check is present on this form in order to be checked or not:

   - If it is checked, then the current cancellation reason type also appears for a Masterpolicy cancellation;
   - If it is not checked, the current cancellation reason type is not displayed as a reason type for a Masterpolicy cancellation.
5. Click **Save and Reload**.
6. The **Insurance Type** grid is unfolded, where you can configure the insurance type for which the reason type applies.

7. Click **Save and Close**.


Reason Type

CANCELLATION REASON TYPE

Reason Type	Employee resigned
Policy Status	Cancelled
Masterpolicy	✓
Masterpolicy Status	Cancelled

INSURANCE TYPE

Insurance Type
Personal Accidents

You are also able to configure a business status mapping between the Cancellation reason type and the policy and Masterpolicy statuses. For example, the Decline by Screening reason type triggers the transition of a policy and a Masterpolicy in the **Decline by Screening** business status after reaching the **Cancellation Final End Date**.

Presented below are the Reason Types for the cancellation, and the business statuses for the Policies and Masterpolicies.

Reason Type	Policy Status	Masterpolicy Status
Decline by Screening	Decline by Screening	Cancelled
Property Sold	Cancelled or Surrender	Cancelled
Insured Death	Cancelled or Surrender	Cancelled
Withdrawal	Withdraw on Client’s Request	Withdraw on Client’s Request
Cancelled by Client	Cancelled or Surrender	Cancelled
Closed by Claim	Closed by Claim	Cancelled
Cancelled	Cancelled or Surrender	Cancelled

Working With APIs

Learn how to use the APIs implemented with Core Policy Admin:

Policy Generation API

The **Policy Generation API** is responsible for generating generic policies into the core system. Requests for policy generation can be received from various external systems. Once the integration with an external system is achieved, some information should be received from the external system in order for the action to be successful - namely, policies are generated into the core system according to the information received from the external system.

Below you can find examples about generating a new policy by an insurance broker and by an insurance agent.

Example Personal Accidents - Broker

A Broker registers a new generic Policy into the system.

```
let pa = {
insuredObject: {
addressId: {
apartmentNo: null,
buildingNo: null,
city: "City",
subcity: "Subcity",
districtCode: "AG",
entrance: 1,
floorNo: 1,
postalCode: "1234",
street: "Street Name",
streetNo: 123,
streetType: null
}},
policyList: [
{
insuranceTypeName: "Personal Accidents",
productCode: "PA",
insuranceProductItemList: [{
code: "DPA",
insuredAmount: 50000.0,
finalPremiumAmount: 144.0
}, {
code: "ICPA",
insuredAmount: 25000.0,
finalPremiumAmount: 792.0
},
{
code: "PDA",
insuredAmount: 35000.0,
finalPremiumAmount: 1209.6
}, {
code: "MEACC",
insuredAmount: 20000.0,
finalPremiumAmount: 691.2
}],
issuedDate: "2022-06-22",
startDate: "2022-06-23",
renewedPolicyNo: null,
quoteNo: "AnaTest30",
totalIndemnityLimit: 1300000,
validityType: "Months",
validity: 12,
agent: {
agentId: null,
type: null
},
broker: {
brokerId: 'Broker1'
},
contractorCode: '816',
insuredCode: '815',
beneficiaryCode: '816',
currency: "RON",
paymentType: "BrokerCollection",
paymentFrequency: "monthly",
mentions: "Insert comment here",
renewType: "Manual",
cardId: "",
cardName: "PA-Card A",
dntResponses: [
{
questionCode: "EXP", // direct
answerValue: false
},
{
questionCode: "CMP", // online
answerValue: true
}
],
quoteConfigCode: 'PAQC'
},
],
masterPolicyNo: null
};
ebs.callActionByNameAsync("PolicyGenerationAPI", pa)
.then(
function(e){
console.log(e.UIResult.Data)
}
);
```

Example Personal Accidents - Agent

An Agent registers a new generic Policy into the system.

```
let pa = {
insuredObject: {
addressId: {
apartmentNo: null,
buildingNo: null,
city: "City",
subcity: "Subcity",
districtCode: "AG",
entrance: 1,
floorNo: 1,
postalCode: "1234",
street: "Street Name",
streetNo: 123,
streetType: null
}},
policyList: [
{
insuranceTypeName: "Personal Accidents",
productCode: "PA",
insuranceProductItemList: [{
code: "DPA",
insuredAmount: 50000.0,
finalPremiumAmount: 144.0
}, {
code: "ICPA",
insuredAmount: 25000.0,
finalPremiumAmount: 792.0
},
{
code: "PDA",
insuredAmount: 35000.0,
finalPremiumAmount: 1209.6
}, {
code: "MEACC",
insuredAmount: 20000.0,
finalPremiumAmount: 691.2
}],
issuedDate: "2022-06-22",
startDate: "2022-06-23",
renewedPolicyNo: null,
quoteNo: "AnaTest30",
totalIndemnityLimit: 1300000,
validityType: "Months",
validity: 12,
agent: {
agentId: "23435788",
type: "Individual person"
},
broker: {
brokerId: null
},
contractorCode: '816',
insuredCode: '815',
beneficiaryCode: '816',
currency: "RON",
paymentType: "BrokerCollection",
paymentFrequency: "monthly",
mentions: "Insert comment here",
renewType: "Manual",
cardId: "",
cardName: "PA-Card A",
dntResponses: [
{
questionCode: "EXP", // direct
answerValue: false
},
{
questionCode: "CMP", // online
answerValue: true
}
],
quoteConfigCode: 'PAQC'
},
],
masterPolicyNo: null
};
ebs.callActionByNameAsync("PolicyGenerationAPI", pa)
.then(
function(e){
console.log(e.UIResult.Data)
}
);
```

Request Data Parameters

Here is the list of data parameters included in the request:

Parameter	Description
insuredObject	Object containing keys to describe the insured object.*
addressId*	Object describing address of the insured object.
apartmentNo	Apartment number.
buildingNo	Building number.
city	City name of a record from City entity or null. City name value must exist in the city nomenclature.
subcity	Subcity name.

Parameter	Description
districtCode	District code of a record from District entity or null District code must exist in the district nomenclature.
entrance	Entrance.
floorNo	Floor.
postalCode	Postal code.
street	Street name.
streetNo	Street number.
streetType	Street type – one of the values existing in the StreetType entity.
policyList	List of policies to generate; Multiple policies can be generated at once.
insuranceTypeName	The name of an insurance type configured in the system, in the Insurance Type entity.
productCode	Insurance product code.
insuranceProductItemList	List of items to be included on the policy.
code	Item code.
insuredAmount	Item insured amount.
finalPremiumAmount	Item premium amount.
excessType	deductibles for coverages.
excessValue	value of deductibles.
issuedDate	Date of issuance, basic format ISO 8601 YYYY-MM-DD.
startDate	Policy begin date, basic format ISO 8601 YYYY-MM-DD.
renewedPolicyNo	Old policy number, in case of renewal.
quoteNo	Quote number.
totalIndemnityLimit	Total indemnity limit on the policy.
validityType	Type of validity Months value supported in the first version of the API .
cardId	Lookup to FTOS_IP_Card, not mandatory.
cardCode	Code of Card (from Proposal Configurator FTOS_ IP_Card), not mandatory, used only if cardId is null or undefined.
validity	How many years/months/days we want this policy to be valid for. 12 value supported in the first version of the API.
agent	Object containing the agent details.
agentId	Agent Id.

Parameter	Description
type	Type of the agent issuing the policy (“Individual person” or “Legal person”).
broker	Object containing broker details.
brokerId	Broker ID.
contractorCode	CustomerInternalId from the Account entity.
insuredCode	CustomerInternalId from the Account entity.
beneficiaryCode	CustomerInternalId from the Account entity.
currency	Currency.
paymentType	Payment type Values: OP for bank transfers PayU for PayU PayU-on time for PayOnTime brokerCollection for Broker Collection.
paymentFrequency	Payment Frequency Values: full for Full (entire payment at once) annually for Annually semiAnnually for Semi-Annually quarterly for Quarterlymonthly for Monthly.
mentions	Special mentions at Policy level.
renewType:	The tipe of the renewal. It can be Manual or Automatic etc.
dntResponses	Array with objects that contains question code and answer for DNT.
masterPolicyNo	The number of Master Policy that you want to link on.

*The keys present in the insuredObject object will change for each type of policy accordingly to the insured object type configurated at insurance product level(for each dimension present on the insured object type configuration there will be a corresponding key into the body, under the insuredObject key).

*The address is mandatory or not, depending on the insured object type’s setting for address. The addressId object structure is defined in the FTOS_ INSQB_Address processor.

Other Insured Object Types Example

For other insured object types the insuredObject key follows the following structure:

```
insuredObject: {
addressId: {
apartmentNo: null,
buildingNo: null,
city: "ALBOTA",
subcity: "GURA VAII",
districtCode: "AG",
entrance: 1,
floorNo: 1,
postalCode: "1234",
street: "Splai Independentei",
streetNo: 123,
streetType: null
},
dimension1: value1
dimension2: value2,
...
dimensionN: valueN
}
```

Response

This is an example of a response:

```
{
    "errorMessage": null,
    "errorCode": null,
    "isSuccess": true,
    "result": {
        "policyData": [{
            "policyBeginDate": "2021-07-17",
            "policyEndDate": "2022-07-16",
            "policyId": "8707f5a4-4fae-40b2-b899-
e27735c5b98e",
            "policyNumber": "80000480",
        }]
    }
}
```

Response description:

Key	Description
Error code	Error code.
Error message	Error message.
isSuccess	Marks if the request was successful or not.
result	Array of objects containing details about the identified policies.
policyData	Array of details for each generated policy.
policyBeginDate	Policy Start Date.
policyEndDate	Policy End Date.
policyId	GUID identifying the policy inside the Core Insurance Master system.
policyNumber	Policy Number.

Error Messages

The following are the error messages that can be encountered during the policy generation process:

Code	Text	Description
ERR.PA.50101	Broker data cannot be identified!	Broker does not exist
ERR.PA.50102	Invalid issued date!	Issue date value provided on issuedDate key is less than current date
ERR.PA.50103	Invalid start date!	Start date value provided on startDate key is less than or equal with issuedDate value
ERR.PA.50104	Invalid paymentType!	Value provided on paymentType key is not valid (not part of the accepted values)
ERR.PA.50105	Invalid currency!	Currency code provided on currency key is not identifed
ERR.PA.50106	Invalid Renew Type!	Renew type invalid
ERR.PA.50107	Quote Number missing!	Quote number is mandatory when creating new policies (but not for renewal)

Code	Text	Description
ERR.PA.50108	Existing policy for the same quote and insurance type!	Another policy of the same insurance type is already registered
ERR.PA.50109	Invalid Excess Type!	Excess type invalid
ERR.PA.50110	Excess (deductibles) cannot have a negative value!	Negative Value for Excess
ERR.PA.50111	Card does not exist	No card with specified card id (available only for Policy Admin 3.4.0)
ERR.PA.50112	Card Name does not exist	No card with specified card name(available only for Policy Admin 3.4.0)
ERR.PA.50113	Card Id is invalid	Card Id is invalid (available only for Policy Admin 3.4.0)
ERR.PA.50144	Invalid ContractorCode!	ContractorCode invalid.
ERR.PA.50145	Invalid InsuredCode!	InsureCode invalid.
ERR.PA.50146	Invalid BeneficiaryCode!	BeneficiaryCode invalid.
ERR.PA.50148	Mandatory attribute <<attributeName>> is missing!	The mandatory attribute <<attributeName>> is missing.
ERR.PA.50149	Value passed by attribute <<attributeName>> does not exist!	The value passed by attribute <<attributeName>> does not exist.
ERR.PA.50150	Policy cannot be issued without an address!	The policy cannot be issued without an address.
ERR.PA.50151	Parent value for attribute <attribute name> is missing!	The parent value for attribute <attribute name> is missing.
ERR.PA.50152	The Insured Object Type configurations are missing on Insurance Product level!	The Insured Object Type configurations are missing on Insurance Product level.

Endpoints

PolicyGenerationAPI

The endpoint is responsible for generating generic policies in the Core system. When generating a policy, some information should be sent in order for the action to be successful. This endpoint also aligns the policies payment schedule with the one from the associated Master Policy.

In the moment a new policy is added on the Master Policy (this happens when the policy is generated, meaning the policy is in the **Proposal** status), the payment schedule for that policy is set up taking into consideration the unpaid installments of the associated Master Policy. More precisely, the system verifies the Master Policy payment schedule and retrieves the number of unpaid installments of that Master Policy. The resulting numberis the number of the installments for that newly added policy.

Example:

Say the **Master Policy Payment Frequency** is set as Quarterly.

In the fourth month from the **Begin Date** of the Master Policy, a new policy is added, where the **End Date** of the policy is the same as the **End Date** of the Master Policy. In this moment, on the Master Policy, the system checks the number of unpaid installments for that Master Policy and finds that there are 2 unpaid installments.

1. The new policy is generated with 2 installments of the newly issued policy. These have the same due date as the 2 unpaid installments of the Master Policy.
2. The total premium amount of the policy is divided into 2.
3. The system updated the following elements of the Master Policy:
    - The **Payment Schedule** is updated by adding the 2 installment sums of the policy to those 2 unpaid installment sums of the Master Policy;
    - The **Payment Amount** is updated by adding to the old premium amount of the Master Policy the newly added policy premium amount;
    - The **IPT Amount** is updated by adding to the old IPT amount of the Master Policy the newly added policy IPT amount.

FTOS_INSPA_getModuleData

The endpoint is used for selecting all the data for a module that has the Id matching to the one obtained from the context JSON object.

Function: `getModuleData()` - The function contains a fluent query that returns the required attributes from the **FTOS_INSPA_PolicyInsuranceItem** entity for a module with a specific Id. After retrieving the results, the function calls the `setData` method for passing the object to the result parameter of the client-side callback function.

Input parameters: N/A.

Output parameters: `moduleList` - JSON object containing the module data.

FTOS_INSPA_getCoveredRisk

The endpoint is used for selecting all the data for a policy covered risk that has the Id matching to the one obtained from the context JSON object. When calling the endpoint with the `ebs.callActionByNameAsync` method, an object containing the `formData.id` gets passed as a parameter.

Function: `getCoveredRisk()` - The function contains a fluent query that returns the required attributes from the FTOS_INSPA_ PolicyInsItemXCoveredRisk entity for a covered risk with a specific id. After retrieving the results the function calls the setData method for passing the object to the result parameter of the client-side callback function.

Input parameters: N/A.

Output parameters: `coveredRiskList` - JSON object containing the covered risk data

LH_PolicyGenerationAPI

This endpoint is used to generate a term life guaranteed/jet issue policy.

Input parameters:

- insuranceTypeName: ''Term Life''
- productCode: "TLF"

- termLifeType: option set
- insuranceProductItemList
    - code: "TLF0"
        - sumInsured: numeric
        - regularPremiumBase: numeric
        - annualizedPremiumBase: numeric (attribute to be added, numeric)
    - code: "CI0"
        - isCriticalIllnessIncluded: (attribute to be added, Boolean type)
        - regularPremiumCriticalIllness: numeric
        - annualizedPremiumCriticalIllness: numeric (attribute to be added)
    - regularPremiumTotal: numeric
    - annualizedPremiumTotal: numeric (attribute to be added)
    - policyFee: numeric
- issuedDate (invariant Date type)
- startDate (invariant Date type)
- renewedPolicyNo: null (text type)
- renewTypeId: null (option set)
- quoteNo: text
- validityType: option set (days, months, years)
- validity: numeric
- policyTerm: numeric

- frequency: option set
- currency: currency code
- paymentType: option set (FTOS_INSPA_PolicyPaymentType, with "directDebit" our only option here)
- agent:
    - FTOS_INS_Agentid (lookup)
- broker:
    - brokerId: null (attribute to be added, lookup)
- contractor (policyholder):
    - Name
    - PIN
    - dateOfBirth
    - email
    - phone
- insured:
    - Name
    - PIN
    - dateOfBirth
    - email
    - phone

Response:

- policyBeginDate;
- policyEndDate;

- policyId;
- policyNumber.

Validations:

- If the issue date value provided on `issuedDate` is less than current date, the following error message is displayed: `"Invalid issued date!"`;
- If the start date value provided on `startDate` is less than or equal with `issuedDate` value, the following error message is displayed: `"Invalid start date!"`;
- If the value provided on `paymentType` is not valid, not part of the accepted values, the following error message is displayed: `"Invalid payment Type!"`;
- If the currency code provided on `currency` is not identified, the following error message is displayed: `"Invalid currency!"`;
- If another policy of the same insurance type is already registered, the following error message is displayed: `"Existing policy for the same quote and insurance type!"`.

Generate Master Policy API

The master policy is a type of "Mother Policy" that includes a collection of "Child Policies" (general policies) that have the same Insurance type. For example, a firm makes a master policy to assure all the cars that the firm hold. In the same time, every car has his own insurance policy.

A Master Policy can be manually generated by the user, but also using the `generateMasterPolicy` API.

Example Masterpolicy

Example - Masterpolicy - generate with **agent**:

```
let p = {
    contractorCode: '816',
    quoteConfigId: '37584433-6116-4ad9-8b29-
578a49ea40f6', //Quote config id
    quoteConfigCode: 'HHI',
    renewalType: 'No', //Renewal type(name)
    agent: {
        agentId: "Agent Test"
    },
    broker: {
        brokerId: null
    },
    startDate: '2022-02-08', // YYYY-MM-DD
    issuedDate: '2022-02-07',
    validity: 12,
    validityType: "Months",
    currency: "EUR",
    paymentFrequency: "monthly",
    paymentType: "BrokerCollection",
    quoteNo: "alina0113",
    //'renewedMasterPolicyId': '', // FTOS_INS_
MasterPolicyid
    mentions: 'Mentions',
};
ebs.callActionByNameAsync("generateMasterPolicy", p)
    .then(
        function(e){
            console.log(e.UIResult)
        }
);
```

Example - Master Policy - generate with **broker**:

```
let p = {
    contractorCode: '816',
    quoteConfigId: '37584433-6116-4ad9-8b29-
578a49ea40f6', //Quote config id
    quoteConfigCode: 'HHI',
    renewalType: 'No', //Renewal type(name)
    agent: {
        agentId: null
    },
    broker: {
        brokerId: 'Broker1'
    },
    startDate: '2022-02-08', // YYYY-MM-DD
    issuedDate: '2022-02-07',
    validity: 12,
    validityType: "Months",
    currency: "EUR",
    paymentFrequency: "monthly",
    paymentType: "BrokerCollection",
    quoteNo: "alina0113",
    //'renewedMasterPolicyId': '', // FTOS_INS_
MasterPolicyid
    mentions: 'Mentions',
};
ebs.callActionByNameAsync("generateMasterPolicy", p)
    .then(
        function(e){
            console.log(e.UIResult)
        }
);
```

Requested Data Parameters:

Here is the list of data parameters included in the request:

Parameter	Description
contractorCode	CustomerInternalId from Account entity.
quoteConfigId	The quote config ID.
quoteConfigCode	The quote config code.
renewalType	The renewal type (name, not display name).
agent	Object containing the agent details.
agentId	Agent ID The value from the agentId needs to be previously added in the Agent entity, if not, an error is received.
type	Type of the agent issuing the policy ("Individual person" or "Legal person").
broker	Object containing broker details.
brokerId	Broker ID.
issuedDate	Date of issuance, basic format ISO 8601 YYYY-MM-DD.

Parameter	Description
startDate	Policy begin date, basic format ISO 8601 YYYY-MM-DD.
validity	How many years/months/days we want this policy to be valid for. 12 value supported in the first version of the API.
validityType	Type of validity Months value supported in the first version of the API
currency	Currency
paymentFrequency	Payment Frequency Values: full for Full(entire payment at once) annually for Annually semiAnnually for Semi-Annually quarterly for Quarterly monthly for Monthly.
paymentType	Payment type Values: OP for bank transfers PayU for PayU PayU-on time for PayOnTime brokerCollection for Broker Collection.
quoteNo	Quote number.
renewedMasterPolicyId	Old Master policy number, in case of renewal.
mentions	Special mentions at Policy level.

Response

This is an example of a response:

```
{
    "isSuccess": true,
    "errorMessage": null,
    "errorCode": null,
    "result": {
        "masterPolicyData": {
            "masterPolicyId": "f57ea260-b442-4604-8cd8-
84a6fa785363",
            "masterPolicyNo": "MP000664",
            "quoteNo": "Armand1001",
            "masterPolicyStartDate": "2022-03-26",
            "masterPolicyEndDate": "2023-03-25"
        }
    }
}
```

Response description:

Key	Description
Error code	Error code
Error message	Error message
isSuccess	Marks if the request was successful or not
result	Array of objects containing details about the identified policies
masterPolicyData	Array of details for each generated policy
masterPolicyId	GUID identifying the policy in CORE
masterPolicyNo	The master policy number
quoteNo	The quote number
masterPolicyStartDate	Master Policy Start Date
masterPolicyEndDate	Master Policy End Date

Error Messages

The following are the error messages that can be encountered during the Masterpolicy generation process:

Code	Text	Description
ERR.PA.50101	Broker data cannot be identified!	Broker does not exist.
ERR.PA.50102	Invalid issued date!	Issue date value provided on issuedDate key is less than current date.
ERR.PA.50103	Invalid start date!	Start date value provided on startDate key is less than or equal with issuedDate value.
ERR.PA.50104	Invalid paymentType!	Value provided on paymentType key is not valid (not part of the accepted values).
ERR.PA.50105	Invalid currency!	Currency code provided on currency key is not identified.

Code	Text	Description
ERR.PA.50108	Existing policy for the same quote and insurance type!	Another policy of the same insurance type is already registered.
ERR.PA.50111	Field is mandatory. Nothing was saved!'	Câmpul este obligatoriu. Datele nu au fost salvate!
ERR.PA.50112	Agent data cannot be identified!	Invalid Agent!
ERR.PA.50137	The quote config code and quote config id are invalid!	Quote config id-ul si quote config code sunt invalide!
ERR.PA.50138	Quote Config ID ore quote config code is mandatory for a Masterpolicy!	Quote Config ID sau quote config code este obigatoriu pentru Masterpolicy!
ERR.PA.50139	Invalid ContractorCode!	ContractorCode invalid!

Endpoints

generateMasterPolicy

The endpoint is responsible for generating Masterpolicies in the Core system. When generating a Masterpolicy, some information should be sent in order for the action to be successful. You can generate a new Masterpolicy using the object above.

Policy Status Change API

This API is used in order to automatize the transition of a policy in the **Issued** and **In Force** statuses.

The `PolicyStatusChangeAPI` endpoint uses a library hat contains functions used in the process.

PolicyStatusChange Library

This library contains 2 main functions (General and Notifications), and each of these 2 functions contain different functions that helps on policy status change process.

The first main function (General) contains the functions presented below and returns them as methods from an object. The object construction is given below:

```
{
    findPolicy: findPolicy,
    validateInput: validateInput,
    updatePolicy: updatePolicy
}
```

(General) functions:

`findPolicy(inputValues)`: function that executes a query to find a specific policy.

Input parameters:

- inputValues - parameters

Output parameters:

- query - found policy

`validateInput(inputValues)`: function that validates the input.

Input parameters:

- inputValues - parameters

Output parameters:

- rez

`updatePolicy(inputValues)`: function that make the updates for the status and other data .

Input parameters:

- inputValues - parameters

Output parameters:

- response

The second main function (Notifications) is used to send notifications on the policy status change (transitions) and contains the functions presented below and returns them as methods from an object. The object construction is given below:

```
{
    getPolicyStatusName: getPolicyStatusName,
    getAccountDataForNotification:
getAccountDataForNotification,
    checkEmailTemplateExists: checkEmailTemplateExists,
    generateNotifcation: generateNotifcation,
    sendNotification: sendNotification
}
```

(Notifications) functions:

`getPolicyStatusName(statusName);`: function that returns the business status display name of a entity.

Input parameters:

- statusName - (string) - The entity status name

Output parameters:

- query - Array with the results

`getAccountDataForNotification(accountdId);`: function that returns different attributes from the “Account” entity, based on the account ID.

Input parameters:

- accountdId - (string) - The account id;

Output parameters:

- query- Array that contains an object with the following results:
    - FirstName
    - LastName
    - Email
    - Phone

`checkEmailTemplateExists(templateName);`: function that checks if an email template exists in the `emailTemplate` entity.

Input parameters:

- templateName - (string) - The template name;

Output parameters:

- tokens - An object with the following results:
    - TemplateName

`generateNotifcation(values);`: function that creates the object that will be used to send a notification.

Input parameters:

- values - (object) - The context object

Output parameters:

- tokens - An object with the following results
    - FirstName
    - LastName
    - Name
    - PolicyNumber

- IssuedDate
- Email
- BusinessStatus
- PolicyEndDate

`sendNotification(policyData, templateName);`: function that sends a notification with the help of the functions presented above and only if the `PolicyAdminUsed` parameter is set to 1. This function will be triggered only on the following FTOS_INSPA_Policy transitions status change:

- Decline by screening
- Closed by Claim
- Withdraw on client's request
- Cancelled
- Lapsed

Input parameters:

- policyData- (object) - The context object
- templateName- (string) - The template name

Output parameters:

- N/A

Call:

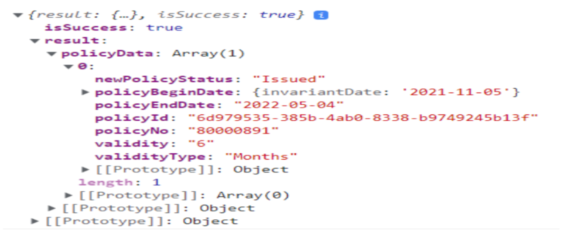
```
var p = {
    "policyIdentifiers": {
        "policyId": null,
        "policyNo": '8000043'
    },
    "updateType": "Issued", // "In Force"
    "policyUpdates": {
        "policyNo": null,
        "policyBeginDate": null,
        "policyValidityType": null,
        "policyValidity": '6'
    }
}

ebs.callActionByNameAsync("PolicyStatusChangeAPI", p)
    .then(
        function (e) {
            console.log(e.UIResult.Data)
        }
    );
```

Error Codes:

- `"ERR06.01 - PolicyId or PolicyNo could not be null!"` → could not find a policy if at least one of the params is not specified.
- `"ERR06.03 - The Policy is not in Proposal status, therefore the transition to Issued is not possible!"` → when transitioning the policy to Issued status the policy need to be in Proposal Status;
- `"ERR06.04 - The Policy is not in Issued status, therefore the transition to In Force is not possible!";` → when transitioning the policy to In Force status the policy need to be in Issued Status;
- `ERR06.05 - Policy does not exists!;` → when no policy with specified name/id is found

Response:



```
▼{result: {…}, isSuccess: true}
   isSuccess: true
  ▼result:
    ▼policyData: Array(1)
      ▼0:
          newPolicyStatus: "Issued"
        ▶policyBeginDate: {invariantDate: '2021-11-05'}
          policyEndDate: "2022-05-04"
          policyId: "6d979535-385b-4ab0-8338-b9749245b13f"
          policyNo: "80000891"
          validity: "6"
          validityType: "Months"
        ▶[[Prototype]]: Object
        length: 1
      ▶[[Prototype]]: Array(0)
    ▶[[Prototype]]: Object
  ▶[[Prototype]]: Object
```

Get Policy Data API

The `FTOS_GetPolicyDataAPI` script is called with an object as data and calls the `getPolicyData` function from the "policyDataAPIs" server automation script library.

An example of calling this function is given below:

```
var p = {
  policyNo: "80000753",
  validityDate: '2021-10-24'
}

ebs.callActionByNameAsync('FTOS_GetPolicyData_API', p)
.then(function(e) {
  console.log(e.UIResult);
});
```

Endpoints

`FTOS_GetPolicyData_API` - This endpoint is used to run the `FTOS_GetPolicyData_API` server automation script.

Request data parameters:

- policyNo: policy number (mandatory);
- validityDate: The request date (optional).

Response:

```
{
  "isSuccess": true,
  "errorMessage": null,
  "errorCode": null,
  "result": {
    "PolicyIssuedDate": "2021-10-05",
    "PolicyBeginDate": "2021-10-06",
    "PolicyEndDate": "2022-10-09",
    "Coverages": [
      {
        "Code": "DPA",
        "BeginDate": "2021-10-10",
```

```json
          "EndDate": "2022-10-09",
          "AmountInsured": 50000.0,
          "PremiumAmount": 96.0,
          "Currency": "RON",
          "IndemnityLimit": 50000.0,
          "WaitingPeriod": 0.0,
          "WaitingPeriodType": null,
          "Modules": [
            {
              "Code": "DPA1",
              "AmountInsured": 50000.0,
              "Currency": "RON",
              "Risks": [
                {
                  "Name": "Personal Accident",
                  "ImplicitReserve": 2000.00,
                  "ValueLimit": 50000.00,
                  "Currency": "RON"
                }
              ]
            }
          ]
        },
        {
          "Code": "ICPA",
          "BeginDate": "2021-10-10",
          "EndDate": "2022-10-09",
          "AmountInsured": 25000.0,
          "PremiumAmount": 528.0,
          "Currency": "RON",
          "IndemnityLimit": 25000.0,
          "WaitingPeriod": 0.0,
          "WaitingPeriodType": null,
          "Modules": [
            {
              "Code": "ICPA1",
              "AmountInsured": 25000.0,
              "Currency": "RON",
              "Risks": [
                {
                  "Name": "Personal Accident",
                  "ImplicitReserve": 2000.00,
                  "ValueLimit": 25000.00,
                  "Currency": "RON"
                }
              ]
            }
          ]
        },
        {
          "Code": "PDA",
          "BeginDate": "2021-10-10",
          "EndDate": "2022-10-09",
          "AmountInsured": 35000.0,
          "PremiumAmount": 806.4,
          "Currency": "RON",
          "IndemnityLimit": 35000.0,
          "WaitingPeriod": null,
          "WaitingPeriodType": null,
          "Modules": [
            {
              "Code": "PDA1",
              "AmountInsured": 35000.0,
              "Currency": "RON",
              "Risks": [
                {
                  "Name": "Personal Accident",
                  "ImplicitReserve": 2000.00,
                  "ValueLimit": 35000.00,
                  "Currency": "RON"
                }
              ]
            }
          ]
        },
        {
          "Code": "MEACC",
          "BeginDate": "2021-10-10",
          "EndDate": "2022-10-09",
          "AmountInsured": 20000.0,
          "PremiumAmount": 460.8,
          "Currency": "RON",
          "IndemnityLimit": 20000.0,
          "WaitingPeriod": 0.0,
          "WaitingPeriodType": null,
          "Modules": [
            {
              "Code": "MEACC1",
              "AmountInsured": 20000.0,
              "Currency": "RON",
              "Risks": [
                {
                  "Name": "Personal Accident",
                  "ImplicitReserve": 2000.00,
                  "ValueLimit": 20000.00,
                  "Currency": "RON"
                }
              ]
            }
          ]
        }
      ],
      "BusinessStatus": "Version Closed"
    }
  }
```

Response description:

- isSuccess: Marks if the request was successful or not;
- Error code: Error code;
- Error message: Error message;
- Result: Null or an object returned as response for the API call.

Error messages:

Code	Text	Description
ERR01.01	ERR01.01 - Invalid/missing policy number!	There is no policy in the system with the requested number!
ERR01.02	ERR01.02 - Invalid date format!	The date should have the following format: 'xxxx-xx-xx' or 'xxxx/xx/xx'!
ERR01.03	ERR01.03. - Invalid date!	The policy was not valid at the requested date!
ERR02.01	ERR02.01 - No coverage!	The policy had no coverages at the requested date!

Code	Text	Description
ERR03.01	ERR03.01 - No modules for coverage!	This coverage had no modules associated at the requested date!
ERR04.01	ERR04.01 - No covered risks!	This module had no covered risks at the requested date!

Policy Claim Data API

The Policy Claim Data API is used to log the policy claim data into the system.

Server Automation Script

`FTOS_INSPA_AddClaimFilesLog`: This script is called with an object as data and calls the `checkObjectFieldsAndInsert` function from the FTOS_INSPA_ AddClaimFilesLogLibrary server automation script library.

An example of calling the function is given below:

```
let p = {
    callDate: '2022-01-14',
    callSource: 'https://www.testexample.com/',
    callUser: 'TestUser',
    policyNo: '80001379',
    claimFileNo: '00001',
    lossDate: '2022-01-10',
    occuredRisk: 'Personal Accident',
    affectedCoverage: 'Death by accidents',
    currentReserve: '5000',
    currentIndemintyPaid: '0',
    currency: 'EUR',
    claimNotificationDate: '2022-01-12',
    claimFileStatus: 'Test lorem ipsum'
};
ebs.callActionByNameAsync("FTOS_INSPA_AddClaimFilesLog", p)
.then(
    function(e){
        console.log(e.UIResult)
    }
);
```

Endpoint

`FTOS_INSPA_AddClaimFilesLog`: This endpoint is used to run the `FTOS_INSPA_AddClaimFilesLog` server automation script.

Request Data Parameters

The request data parameters for this endpoint are given below:

Parameter	Description
callDate	The date when the call was made.
callSource	The source from where the endpoint was called.
callUser	The user name of the person who calls the endpoint.
policyNo	The policy number.
claimFileNo	The claim file number.
lossDate	The loss date.
occuredRisk	The occurred risk.
affectedCoverage	The affected coverage.
currentReserve	The current reserve.
currentIndemintyPaid	The current indemnity paid.
currency	The currency.
claimNotificationDate	The claim notification date.
claimFileStatus	Message (optional).

Response

The response is given below:

```
{
    "isSuccess": true,
    "errorMessage": null,
    "errorCode": null,
    "result": "Call sent sucessfully!"
}
```

Response description:

Key	Description
isSuccess	Marks if the request was successful or not
Error code	Error code
Error message	Error message
Result	Null or a confirmation message returned as response for the API call

Error Messages

The following are the error messages that could be encountered in the policy claim data.

Code	Text	Description
IP-30107	IP-30107 - Empty field!	The field ' (dynamic variable) ' cannot be empty!(IP-30107)
IP-30108	IP-30108 - Invalid policy number!	There is no policy with the " (dynamic variable for policy number)" number!(IP-30108)!
IP-30109	IP-30109 - Invalid currency!	The "currency" field is invalid!(IP-30109)
IP-30110	IP-30110 - Invalid "affectedCoverage"!	'The "affectedCoverage" field is invalid!(IP-30110)
IP-30111	IP-30111 - Invalid "occuredRisk"!	The "occuredRisk" field is invalid!(IP-30111)

Flow Parameters and Scheduled Jobs

The following flow parameters and scheduled jobs are used with the **Core Policy Admin** solution.

1 Flow Parameters

Parameter Name	Policy Automatically Withdraw
Details	Type: Integer; Code: **PWDAY**
Component	Policy Admin
Correlated with	**FTOS_INSPA_Policy_AutomaticallyWithdraw** scheduled job
Description	This parameter sets the **Automatically Withdrawal Day** of a policy after a number of days since the **First Installment** on that policy moved to **Unpaid** status. When the parameter is fulfilled, the policy is transitioned to **Withdraw** status.

Parameter Name	maturityNotification
Details	Type: Integer; Code: **MTN**
Component	Policy Admin, Notifications
Correlated with	N/A
Description	This parameter sets the number of days for notifying the policy **Maturity** status to the policy holder, after the policy reaches that specific status. For example: 5 days after the policy reaches **Maturity** status, a maturity notification is sent to the contract holder informing that the policy reached its end date.

Parameter Name	Days After Unpaid Due Date
Details	Type: Integer; Code: **NDBR**
Component	Policy Admin, Billing & Collection
Correlate with	**FTOS_PA_PolicyLapsed** scheduled job

Parameter Name	Days After Unpaid Due Date
Description	This parameter sets the number of days after an unpaid installment's due date, registered on a policy. When the **DAUDD** parameter is fulfilled, the policy is transitioned to **Lapsed** status. For example: the **Lapsing Day** of a policy is triggered after the passing of the chosen number of days since the **Last Installment** on that policy moved to **Unpaid** status.

Parameter Name	No of Days Before Renewal
Details	Type: Integer; Code: **NDBR**
Component	Policy Admin
Correlated with	**FTOS_PA_PolicyRenewal** scheduled job
Description	This parameter correlates the starting date of the **Renewal** process with a given number of days before the **End Date** of a policy. For example: it is required to start the **Renewal** process within 5 days before policy reaching its **End Date**. As a result the parameter is set to 5 days.

2 Scheduled Jobs

Scheduled jobs are automated procedures that perform certain tasks, running at a specific time or on a recurring schedule. Read detailed information about scheduling jobs in the Innovation Studio User Guide's dedicated page.

The following scheduled jobs are available with **Core Policy Admin**:

FTOS_PA_PolicyLapsed

Scheduled for 04:00 AM, daily run.

Used for policies with last installments in **Unpaid** status, this job automatically moves policies from **Enforced** to **Lapsed** status, when the **DAUDD** parameter is fulfilled.

When in **Lapsed**, the policy is modified as follows:

- The **End Date** becomes the **Due Date** of the unpaid installment.
- If there are no claims registered, the **Premium Amount** becomes equal to the **Paid Premium** (sum of paid installments). If there are claims registered on the policy, the **Premium Amount** doesn't change.
- If the **Premium Amount** is changed, the installment schedule is updated including just the paid installments.

The system also checks the Master Policies payment and, if there will be any installments still unpaid, being in the **Unpaid** status) after X days for a Master Policy (X days since the due date of the last unpaid installment of the Master Policy), then that Master Policy changes its status from **Inforce** to **Lapsed**.

After a Master Policy is moved in the **Lapsed** status, that Master Policy is modified as follows:

- The end date becomes the due date of the unpaid installment.
- The new premium amount is equal to the paid premiums (the sum of paid installments).
- Because the premium amount is changed, the payment schedule of the Master Policy is updated and only includes the paid installments.

The number of days is set in the "Days after unpaid due date" existing and generic parameter, set at portfolio/contract level.

The following service runs as part of this job:

1. **FTOS_PA_PolicyLapsed**

   This service automatically moves policies from **Enforced** to **Lapsed** status, when the **DAUDD** parameter is fulfilled.

FTOS_INSPA_Policy_AutomaticallyWithdraw

Scheduled for 02:00 AM, daily run.

This job moves policies from **Proposal** to **Withdraw** status when they satisfy the following conditions: their first premium is still in **Unpaid** status after a given number of days - **PWDAY** parameter, after their first due date.

FTOS_INSPA_Policy_DeleteAfter10m

Scheduled for 01:00 AM, daily run.

This job deletes from the system the policies with no business transition in the last x minutes - number of minutes set in the **NMACLBWT** parameter.

FTOS_PA_PolicyRenewal

Scheduled for At 03:00 AM, daily run.

This job searches for the **End Date** and the **NDBR** parameter in order to identify the policies in **InForce** status that are suitable for automatic or manual renewal, for renewal offers.
The job returns the policies with `Policy End Date (Initial End Date) - No of days before renewal (parameter set) = Current Date`.

The following service runs as part of this job:

1. **FTOS_PA_PolicyRenewal**

   This service searches for the **End Date** and the **NDBR** parameter in order to identify the policies in **InForce** status that are suitable for automatic or manual renewal, for renewal offers.

FTOS_INSPA_Policy_IssuedToEnforced

Scheduled for 01:00 AM, daily run.

For the policies that are in the **Issued** business status, have the same **Begin Date** as the **Current Date** and have the **Automatic InForce** checkbox selected, at Product level.
This job verifies whether the above conditions are fulfilled and moves policies from **Issued** to **InForce** status.

The following service runs as part of this job:

1. **FTOS_INSPA_Policy_IssuedToEnforced**

   This service verifies if policies that are in the **Issued** business status, have the same **Begin Date** as the **Current Date** and have the **Automatic InForce** checkbox selected, and moves them from **Issued** to **InForce** status.

FTOS_PA_PolicyTerminationProcess

Scheduled for every 50 minutes, starting at 2 minutes past the hour.

This job verifies whether the **Policy End Date** is equal to the **Current date** and moves **InForce** policies to **Maturity** status.

FTOS_PA_PolicyMaturity

Scheduled to run daily at 23:59.

This job is set to transition policies' status from **In Force** or **Suspended** to the **Maturity** status when policies Current Date is greater than the End Date. This job also automatically transitions Master Policies' status from **In Force** to **Maturity** when the Master Policies Current Date is greater than the End date.

The following service runs as part of this job:

1. **FTOS_PA_PolicyMaturity**

   This service transitions the policies and master policies to the **Maturity** status, when their Current Date is greater than their End Date.

Core Policy Admin Formulas

The formulas are defined using Business Formulas in Innovation Studio. The current page reveals the logic behind formulas used in the Life and Health module. For more information on how the Business Formulas works, check out the Business Formulas chapter in the Innovation Studio product documentation.

The following formulas and calculations are used in the Core Policy Admin module.

Register/Cancel an MTA Request

- ***The additional coverage premium amount*** *= (new annual premium - initial annual premium)/12* No. of uninsured months*
- ***Updated coverage premium amount*** *= Initial premium amount * (12-No of uninsured months)/12 + The additional coverage premium amount*
- ***Additional policy premium amount*** *= Sum (The additional coverage premium amounts)*
- ***Updated policy premium amount*** *= Sum (Updated coverage premium amount)*

MTA Refactoring According to Pro Rata Type

Daily prorata = 1/365 from the premium amount of a contract.

Monthly prorata = 1/12 from the premium amount of a contract.

If Prorata type = daily

- ***AdditionalCoveragePremiumAmount*** *= (coverageNewPremium - initialPremiumAmount) / 365 * uninsuredPeriod*
- ***UpdatedCoveragePremiumAmount*** *= initalPremiumAmount * (365 - uninsuredPeriod) / 365*
- ***FreqAdditionalCoveragePremiumAmount*** *= (newPremiumAmount - initialPremiumAmount) / 365 * unisuredPeriod*
- ***FreqUpdatedCoveragePremiumAmount*** *= (initialPremiumAmount * (365- uninsuredPeriod) / 365) + (newPremiumAmount / 365 * uninsuredPeriod)*

If Prorata type = monthly

- ***AdditionalCoveragePremiumAmount*** *= (coverageNewPremium - initialPremiumAmount) / 12 * uninsuredPeriod*

- ***UpdatedCoveragePremiumAmount*** = *initalPremiumAmount * (12 - uninsuredPeriod) / 12*
- ***FreqAdditionalCoveragePremiumAmount*** = *(newPremiumAmount - initialPremiumAmount) / 12 * unisuredPeriod*
- ***FreqUpdatedCoveragePremiumAmount*** = *(initialPremiumAmount * (12 - uninsuredPeriod) / 12) + (newPremiumAmount / 12 * uninsuredPeriod)*

Calculate Premium Amounts after MTA Cancellation

The following calculations are used for the premium amounts after a policy MTA is cancelled.

- **The additional coverage premium amount** = (new annual premium - initial annual premium)/12* No. of uninsured months
- ***Updated coverage premium amount*** = *Initial premium amount * (12- No of uninsured months)/12 + The additional coverage premium amount*
- ***Additional policy premium amount*** = *Sum (The additional coverage premium amounts)*
- ***Updated policy premium amount*** = *Sum (Updated coverage premium amount)*

Calculate Premium Amounts According to the Prorata Type

The parameter contains 2 types of prorata:

- Daily prorata as 1/365 from the premium amount of a contract;
- Monthly prorata as 1/12 from the premium amount of a contract.

Parameter location: Settings - Insurance parameters

Parameter name: Prorata type

Parameter code: PRT

Parameter type and value: Option set: either Daily or Monthly

For the premium adjustments, either the daily pro rata or the monthly pro rata can be chosen.

If Prorata type = daily

- ***AdditionalCoveragePremiumAmount*** *= (coverageNewPremium - initialPremiumAmount) / 365 * uninsuredPeriod*
- ***UpdatedCoveragePremiumAmount*** *= initalPremiumAmount * (365 - uninsuredPeriod) / 365*
- ***FreqAdditionalCoveragePremiumAmount*** *= (newPremiumAmount - initialPremiumAmount) / 365 * unisuredPeriod*
- ***FreqUpdatedCoveragePremiumAmount*** *= (initialPremiumAmount * (365- uninsuredPeriod) / 365) + (newPremiumAmount / 365 * uninsuredPeriod)*

If Prorata type = monthly

- ***AdditionalCoveragePremiumAmount*** *= (coverageNewPremium - initialPremiumAmount) / 12 * uninsuredPeriod*
- ***UpdatedCoveragePremiumAmount*** *= initalPremiumAmount * (12 - uninsuredPeriod) / 12*
- ***FreqAdditionalCoveragePremiumAmount*** *= (newPremiumAmount - initialPremiumAmount) / 12 * unisuredPeriod*
- ***FreqUpdatedCoveragePremiumAmount*** *= (initialPremiumAmount * (12 - uninsuredPeriod) / 12) + (newPremiumAmount / 12 * uninsuredPeriod)*

Calculate the Duration and Installments of Master Policies

In the Master Policy entity, the `noOfValidityMonths` attribute represents the number of calendar months between the begin date and the end date of the Master Policy. It is calculated as per below:

No of validity months *= Master Policy Begin Date - Master Policy End Date*

The number of installments of the Master Policy (`noOfInstallments` attribute, numeric type) is calculated based on the attributes: number of validity months (calculated above) and payment frequency of the Master Policy.

Payment frequency	**No of installments when v>-12 months**	**6<=v<12 months**	**v<6 months**
annually	roundup(v/12;0)	N/A	N/A
semi-annually	roundup(v/6;0)	roundup (v/6;0)	N/A
quarterly	roundup(v/3;0)	roundup (v/3;0)	roundup (v/3;0)
monthly	v	v	v

Where:

- v = The number of validity months of the Master Policy (noOfValidityMonths);
- N/A = The system does not allow this combination.