

# FintechOS HPFI 22.1

## Administration Guide

# TOC

|  |    |
|--|----|
| FintechOS HPFI Administration Guide .....            | 12 |
| Installation .....                                   | 13 |
| System Requirements .....                            | 13 |
| HPFI Platform System Requirements .....              | 13 |
| Required Server Roles .....                          | 13 |
| Required Features .....                              | 14 |
| Required Web Server Role (IIS) / Role Services ..... | 14 |
| FintechOS Portal System Requirements .....           | 15 |
| System Requirements for WebRTC Components .....      | 16 |
| Cookie Specifications .....                          | 17 |
| Server-Side Encrypted Cookies .....                  | 17 |
| Client-Side Unencrypted Cookies .....                | 17 |
| Self-Hosted Installation .....                       | 18 |
| System Requirements .....                            | 18 |
| Reference Architecture .....                         | 19 |
| Pre-Installation Checklist .....                     | 21 |
| Installation Steps .....                             | 22 |
| System Digital Solution Packages Installation .....  | 59 |
| Prerequisites .....                                  | 61 |
| Pre-Installation Checklist .....                     | 61 |
| Automatic Installation Steps .....                   | 62 |
| Post-Installation Setup .....                        | 64 |
| Manual Import Installation Steps .....               | 65 |
| Self-Hosted Migration from v21.x to v22.x .....      | 66 |
| 1 Environment Deployment .....                       | 66 |

|   |    |
|---|----|
| 2 Install Additional FintechOS Portal Instances .....                       | 68 |
| 3 Final Checks .....  | 69 |
| Upgrade Process .....   | 70 |
| Prerequisites .....   | 70 |
| Upgrade Steps .....   | 70 |
| 1 Preparing The Data Deployment Package .....                               | 70 |
| 2 Upgrading Portal and Studio .....   | 70 |
| 3 Upgrading The Services .....  | 71 |
| MainDbServer .....  | 71 |
| PortalWebApp .....  | 71 |
| DesignerWebApp .....  | 72 |
| JobServer .....   | 72 |
| MessageComposer .....   | 73 |
| 4 Importing Deployment Packages .....                                       | 74 |
| Generate an SSL Certificate with win-acme .....                             | 74 |
| Migrate User Accounts and Roles to the FintechOS<br>Identity Provider ..... | 77 |
| Check the Migration Output .....  | 83 |
| Troubleshooting .....   | 84 |
| Configuration Manager .....   | 86 |
| Manage Vault Secrets .....  | 86 |
| Directory Structure .....   | 86 |
| Secrets .....   | 87 |
| Vault Connection .....  | 88 |
| Enable web.config Override .....  | 88 |
| DevOps .....  | 90 |
| Configure the File Upload Folder .....                                      | 90 |
| Where Should I Store Files? .....   | 90 |
| Local File System Storage .....   | 91 |

|  |            |
|--|------------|
| Automatically Create File Upload Subfolders .....  | 92         |
| Azure Blob Storage .....   | 93         |
| Azure Resource Manager templates support .....   | 94         |
| Amazon S3 Buckets Storage .....  | 94         |
| Importing and Exporting Deployment Packages .....  | 96         |
| File-Type Upload Control .....   | 97         |
| Enable the file-type upload control .....  | 97         |
| File-Type Upload Processing .....  | 98         |
| HPFI API a Standalone Web App .....  | 99         |
| Activating Localization Debug Mode .....   | 100        |
| Configure Notifications for Operations .....   | 101        |
| Observability .....  | 103        |
| Send log messages to the system console .....  | 103        |
| Send log messages to local file storage .....  | 104        |
| Send log messages to a Seq structured log server .....   | 105        |
| Send log messages to an Azure Application Insights service .....                                   | 106        |
| Configure Azure Application Insights telemetry .....   | 106        |
| Logging context .....  | 107        |
| Prevent Sequencer Infinite Loops .....   | 107        |
| <b>Integrations .....</b>  | <b>109</b> |
| FintechOS Service Pipes .....  | 109        |
| App Service Configuration .....  | 109        |
| Configuration Manager Settings .....   | 111        |
| Configuration for Environments Using the FintechOS Identity Provider .....                         | 112        |
| Configuration for Environments Using Legacy Authentication (non-FintechOS Identity Provider) ..... | 124        |
| User Roles .....   | 127        |
| Connect to Azure Notification Hubs .....   | 128        |

|  |     |
|--|-----|
| Push Notifications Log .....   | 131 |
| FAQs .....   | 132 |
| Collect Logging Data in Azure Application Insights .....                                 | 133 |
| Prerequisites .....  | 133 |
| Configuration .....  | 134 |
| Azure Application Insights Logging Vault Configuration .....                             | 134 |
| Azure Application Insights Telemetry Configuration .....                                 | 135 |
| Collected Data .....   | 136 |
| Severity Levels .....  | 136 |
| Information Severity Level Examples .....  | 136 |
| Warning Severity Level Example .....   | 136 |
| Error Severity Level Examples .....  | 137 |
| CertSign Integration for electronic signature .....                                      | 137 |
| Set up for the automatic signature with qualified<br>electronic sign .....               | 139 |
| Calling the automatic signature with qualified electronic<br>sign .....                  | 140 |
| FTOS ESign Services API .....  | 143 |
| RequestSign .....  | 143 |
| Configure the CData Sync Service .....   | 148 |
| System Requirements .....  | 149 |
| Installation .....   | 149 |
| Upgrade .....  | 150 |
| Uninstall .....  | 150 |
| Configure the Payment Processor Service Provider .....                                   | 150 |
| 1 Define a new type of section in the web.config file for<br>the payment processor ..... | 150 |
| 2 Add the connection settings for your payment<br>processor .....                        | 151 |
| Configure the FTOSApiSMS Service .....   | 152 |

|   |            |
|---|------------|
| 1 Add a new section in the web.config file for the FTOSApiSMS service .....             | 152        |
| 2 Add the configuration settings for the FTOSApiSMS service .....                       | 152        |
| Customize the SMS messages sent for Multi-Factor Authentication .....                   | 153        |
| Configure the OneyTrust Digital Review service .....                                    | 154        |
| <b>Security .....</b>   | <b>155</b> |
| Data Encryption and Security .....  | 155        |
| XSS Prevention .....  | 156        |
| Authentication .....  | 157        |
| FintechOS Identity Provider .....   | 157        |
| Identity Brokering .....  | 158        |
| FintechOS Identity Provider Settings .....  | 158        |
| Set up Service Account Roles for the Innovation Studio Client .....                     | 165        |
| How users log in the HPFI Portal or Innovation Studio .....                             | 166        |
| HPFI user account automatic synchronization .....                                       | 166        |
| Using Azure AD as External Identity Provider .....                                      | 167        |
| 1 Register the FintechOS Identity Provider as an Azure App .....                        | 167        |
| (Optional) Configure Access for Azure AD Users .....                                    | 168        |
| Grant Consent to Access APIs .....  | 168        |
| 2 Set up the Azure AD App as Identity Provider in the FintechOS Identity Provider ..... | 168        |
| 3 Map Azure AD Security Groups to FintechOS Security Roles .....                        | 169        |
| Set Up the ID Tokens Sent by Azure AD to Include Security Groups Information .....      | 170        |
| Define Mappings between Azure AD Security Groups and FintechOS Security Roles .....     | 170        |
| 4 Disable User Account Editing in Innovation Studio .....                               | 171        |
| Using Okta as External Identity Provider .....  | 172        |
| 1 Create an Okta App Integration for the FintechOS Identity Provider .....              | 172        |

|   |     |
|---|-----|
| 2 Set up the Okta server as Identity Provider in the FintechOS Identity Provider .....        | 173 |
| 3 Map Okta User Groups to FintechOS Security Roles .....                                      | 175 |
| Set Up the ID Tokens Sent by Okta to Include Security Groups Information .....                | 175 |
| Define Mappings between Okta Groups and FintechOS Security Roles .....                        | 176 |
| 4 Disable User Account Editing in Innovation Studio .....                                     | 177 |
| Using AWS Cognito as External Identity Provider .....   | 178 |
| 1 Create an AWS Cognito App for the FintechOS Identity Provider .....                         | 178 |
| 2 Set up the AWS Cognito server as Identity Provider in the FintechOS Identity Provider ..... | 179 |
| 3 Map AWS Cognito User Groups to FintechOS Security Roles .....                               | 181 |
| 4 Disable User Account Editing in Innovation Studio .....                                     | 182 |
| Deprecated Identity Providers .....   | 183 |
| Microsoft Active Directory Authentication .....   | 183 |
| AD Standard Login Configuration .....   | 184 |
| Automatically Adding Users from AD .....  | 185 |
| Preserving System Users .....   | 186 |
| Limiting Query Scope on AD .....  | 187 |
| Customizing Group Membership Checks .....   | 188 |
| Azure Active Directory Authentication .....   | 189 |
| Configure OpenID Settings .....   | 190 |
| Configuration Keys .....  | 191 |
| Parameters .....  | 191 |
| Set up Login/Logout Redirect URIs .....   | 193 |
| Groups Mapping .....  | 194 |
| Authentication with Okta .....  | 195 |
| How to Set up the Okta Authentication .....   | 195 |
| Step 1. Create and configure the Okta app .....   | 196 |
| Step 2. Configure the Experience Portal .....   | 197 |

|  |     |
|--|-----|
| How it Works .....   | 200 |
| Group mapping in FintechOS .....   | 201 |
| How users log in the Portal .....  | 201 |
| Troubleshooting Okta Redirect Error .....                                | 202 |
| Authentication with Active Directory Federation Services .....           | 204 |
| Add keys to Vault secrets .....  | 204 |
| Configuration Keys: .....  | 205 |
| Parameters: .....  | 205 |
| ADFS configuration .....   | 206 |
| Group mapping in FintechOS .....   | 219 |
| Authentication with AWS Cognito .....                                    | 220 |
| Add keys to Vault secrets .....  | 220 |
| Configuration Keys: .....  | 221 |
| Parameters: .....  | 222 |
| Group mapping for users .....  | 223 |
| Browser Based Multi-Factor Authentication .....                          | 224 |
| 1 Create a Browser Authentication Flow .....                             | 224 |
| (Optional) Enable Conditional OTP only for specific roles .....          | 226 |
| 2 Associate the Authentication Flow to a Client .....                    | 227 |
| Authenticator Reset .....  | 227 |
| Email/SMS/IVR Multi-Factor Authentication .....                          | 228 |
| 1 Set Up Your Email/SMS/IVR Service Providers .....                      | 229 |
| 2 Configure the Service Pipes to Use the Defined Service Providers ..... | 231 |
| 3 Create an Authentication Flow .....                                    | 231 |
| 4 Configure the Flow's MFA Execution Step .....                          | 232 |
| 5 Activate the Authentication Flow .....                                 | 234 |
| Deprecated Multi-Factor Authentication .....                             | 235 |
| SMS-based Two-Factor Authentication .....                                | 236 |
| How it works? .....  | 236 |
| How to set up the SMS-based MFA? .....                                   | 236 |

|  |     |
|--|-----|
| 1 Enable Multi-Factor Authentication .....                                 | 236 |
| 2 Configure the Job Server for MFA .....                                   | 239 |
| Configure Multi Factor Authentication to use an SMS Service provider ..... | 241 |
| Password reset SMS for the log-in credentials .....                        | 242 |
| Email-based Two-Factor Authentication .....                                | 243 |
| How it works? .....  | 243 |
| How to set up the Email-based MFA? .....                                   | 243 |
| Step 1 Enable Multi-Factor Authentication .....                            | 243 |
| Step 2. Configure the Job Server for MFA .....                             | 246 |
| Register TLS Client Certificates .....                                     | 248 |
| Usage in server-side scripts .....   | 251 |
| Configure JSON Web Token (JWT) Providers .....                             | 251 |
| Usage in server-side scripts .....   | 254 |
| Authorization .....  | 254 |
| Security Roles .....   | 254 |
| Data Ownership .....   | 256 |
| Password Security .....  | 256 |
| Locked account .....   | 257 |
| Password expired .....   | 258 |
| Activate Forgot Password Feature .....                                     | 259 |
| Configure Password Change .....  | 260 |
| Setting password minimum age .....   | 261 |
| Setting password expiry .....  | 261 |
| Configuring password change based on password history .....                | 262 |
| Setting password about to expire notifications .....                       | 263 |
| Skipping the password expiry rule for specific security roles .....        | 264 |
| Reset Password Global Email Template .....                                 | 265 |
| Customize Reset Password Email Template .....                              | 265 |

|  |     |
|--|-----|
| Step 1. Add a specific secret in Vault .....                             | 266 |
| Step 2. Create FTOS_ResetPasswordEmail on-demand automation script ..... | 266 |
| Global Password Complexity Settings .....                                | 267 |
| Customize Password Complexity Rules .....                                | 268 |
| Step 1. Add a specific secret in Vault: .....                            | 268 |
| Step 2. Create FTOS_ResetPasswordRules on-demand automation script ..... | 269 |
| Temporary Blocked User .....   | 270 |
| How to setup the number of retries Portal .....                          | 271 |
| When using the EbsAuth provider .....                                    | 272 |
| Send Notifications for Locked Accounts or Password Resets .....          | 273 |
| communicationChannels .....  | 274 |
| Custom email providers .....   | 275 |
| notificationTypes .....  | 275 |
| Random Character Password Authentication .....                           | 276 |
| Architecture .....   | 277 |
| 1 Capture the Username .....   | 277 |
| 2 Generate the random characters .....                                   | 277 |
| Unauthorize Inactive Users .....   | 278 |
| Session Expiration Time .....  | 278 |
| OTP Login Session .....  | 279 |
| File Upload Malware Scanning .....                                       | 280 |
| Log Management .....   | 281 |
| Security Logs .....  | 282 |
| Operating system logs and application logs .....                         | 284 |
| Data Audit .....   | 287 |
| Entity Audit .....   | 287 |
| HPFI Logging .....   | 289 |

How to Configure the Logging of CRUD Operations .....289

    HPFI API Logging .....289

EbsLogs.ApiLog Schema .....290

How to Configure the HPFI API Logging .....290

    Restrict Access to Innovation Studio Based on the  
    StudioUser Role .....292

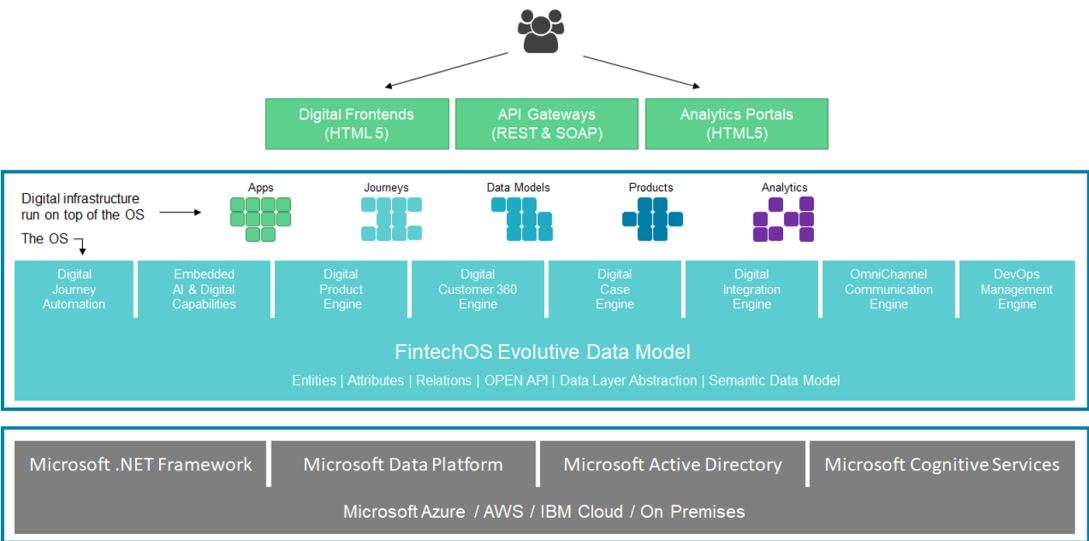
# FintechOS HPFI

## Administration Guide

The FintechOS High Productivity Fintech Infrastructure (HPFI) is an innovation acceleration software platform that enables fast, plug & play, comprehensive digital transformation of companies that offer financial services.

HPFI is a highly scalable technology that can be run both on premises and from the cloud.

While deployment on premises are still permitted with the current version of HPFI, we recommend using one of the enterprise cloud providers that HPFI is compatible with. Below you can see a more detailed technological architecture on how HPFI runs on Microsoft Azure, AWS, IBM Cloud, or on-premises deployments.



# Installation

This section provides system specifications and describes how to install and set up the FintechOS Platform.

---

## System Requirements

- ["HPFI Platform System Requirements" below](#)
- ["FintechOS Portal System Requirements" on page 15](#)
- ["System Requirements for WebRTC Components" on page 16](#)

## HPFI Platform System Requirements

| Software minimum required version | 18.2.x                 | 20.1.x (Genie)         | 20.2.x (Pulsar)        | 22.1                |
|-----------------------------------|------------------------|------------------------|------------------------|---------------------|
| .NET Framework                    | 4.6.2                  | 4.6.2                  | 4.7.2                  | 4.7.2               |
| SQL Server                        | SQL Server 2012 (11.x) | SQL Server 2012 (11.x) | SQL Server 2012 (11.x) | SQL Server 2019     |
| Windows Server                    | Windows Server 2012 R2 | Windows Server 2012 R2 | Windows Server 2012 R2 | Windows Server 2019 |

Below are details about which Windows Server roles and features are required. They were determined on a Windows Server 2012 R2. You must determine the equivalents for your particular Windows Server version.

### Required Server Roles

Web Server (IIS)

## Required Features

- NET Framework 3.5 Features \ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- NET Framework 4.5 Features \ .NET Framework 4.5
- NET Framework 4.5 Features \ ASP.NET 4.5
- NET Framework 4.5 Features \ WCF Services \ HTTP Activation
- NET Framework 4.5 Features \ WCF Services \ TCP Port Sharing
- Windows PowerShell \ Windows PowerShell 4.0
- Windows Process Activation Service \ Process Model 17
- Windows Process Activation Service \ Configuration APIs

## Required Web Server Role (IIS) / Role Services

- Web Server \ Common HTTP Features \ Default Document
- Web Server \ Common HTTP Features \ Directory Browsing
- Web Server \ Common HTTP Features \ HTTP Errors
- Web Server \ Common HTTP Features \ Static Content
- Web Server \ Common HTTP Features \ HTTP Redirection
- Web Server \ Health and Diagnostics \ HTTP Logging
- Web Server \ Performance \ Static Content Compression
- Web Server \ Performance \ Dynamic Content Compression
- Web Server \ Security \ Request Filtering
- Web Server \ Security \ Basic Authentication
- Web Server \ Security \ URL Authorization
- Web Server \ Security \ Windows Authentication

- Web Server \ Application Development \ .NET Extensibility 4.5
- Web Server \ Application Development \ Application Initialization
- Web Server \ Application Development \ ASP.NET 4.5
- Web Server \ Application Development \ ISAPI Extensions
- Web Server \ Application Development \ ISAPI Filters
- Web Server \ Application Development \ Server Side Includes
- Web Server \ Application Development \ WebSocket Protocol
- Web Server \ Management Tools \ IIS Management Scripts and Tools

## FintechOS Portal System Requirements

FintechOS Portal can run on the following browsers, on both desktop and mobile devices:

| Browser                | Operating System         |
|------------------------|--------------------------|
| Google Chrome          | Windows 10               |
| Mozilla ESR            | Windows 10               |
| Mozilla Firefox        | Windows 10               |
| Microsoft Edge         | Windows 10               |
| Opera                  | Windows 10               |
| Safari (desktop)       | macOS - latest version   |
| Safari (mobile)        | IOS - latest version     |
| Google Chrome (mobile) | Android - latest version |

### IMPORTANT!

Please be aware that FTOS-Studio is fully compatible only with Google Chrome!

We recommend that you use the latest major version available for the browser.

## System Requirements for WebRTC Components

### NOTE

Components that are using WebRTC impose a series of limitations for our services to work correctly.

**WebRTC (Web Real-Time Communication)** is a free, open-source project that provides web browsers and mobile applications with real-time communication (RTC) via simple application programming interfaces (APIs). It allows audio and video communication to work inside web pages by allowing direct peer-to-peer communication, eliminating the need to install plugins or download native apps. WebRTC is being standardized through the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF).

Its mission is to "enable rich, high-quality RTC applications to be developed for the browser, mobile platforms, and IoT devices, and allow them all to communicate via a common set of protocols".

Taking into account the [versions supported by WebRTC](#) and our 3rd party providers, please find below the list of supported browsers on different devices.

### Desktops / Laptops:

| Browser              | Recommended Version | Supported Version |
|----------------------|---------------------|-------------------|
| Google Chrome        | Latest              | 79 and greater    |
| Mozilla Firefox      | Latest              | 71 and greater    |
| Microsoft Edge       | Versions 80 - 81    | 79 and greater    |
| Safari               | 13.1 and greater    | 13.1 and greater  |
| Opera                | Latest              | 66 and greater    |
| Internet Explorer 11 | Not Supported       | Not Supported     |

### NOTE

As WebRTC is in constant development, please ensure to always use the latest version.

We recommend to use Google Chrome for the best overall experience.

### Mobile Devices:

| Operating System                  | Browser         | Supported Version            |
|-----------------------------------|-----------------|------------------------------|
| Android (version 6.0 and greater) | Google Chrome   | Latest                       |
| Android (version 6.0 and greater) | Mozilla Firefox | Not Supported                |
| Android (version 6.0 and greater) | Opera           | Latest                       |
| iOS (version 12 and greater)      | Safari          | 2 most recent major versions |
| iOS                               | Google Chrome   | Not Supported                |
| iOS                               | Microsoft Edge  | Not Supported                |
| iOS                               | Mozilla Firefox | Not Supported                |
| iOS                               | Opera           | Not Supported                |

**NOTE**

WebRTC does not support Chrome on iOS devices. On iOS you can only use the Safari engine.

## Cookie Specifications

The FintechOS HPFI uses the following cookies:

### Server-Side Encrypted Cookies

These are security cookies used for client-server authentication.

| Cookie                   | Description   |
|--------------------------|---|
| .EBSCORE\$1              | Authentication cookie.  |
| .EBSCORE\$1-CSRFToken    | CSRF prevention token cookie.   |
| .EBSCORE\$1_PartialToken | Used for partial password authentication (only available with FintechOS Identity Provider). |

### Client-Side Unencrypted Cookies

These are client-side generated cookies that store a transient state, therefore they do not require encryption.

| Cookie                                     | Description  |
|--|--|
| .EBSCORE\$1-CorrelationId                  | Browser session specific cookie used for logging.                    |
| .EBSCORE\$1-culture                        | Current language. Preserves language settings between user sessions. |
| .EBSCORE\$1-hasCollapsedHeader             | FintechOS Portal header settings.                                    |
| .EBSCORE\$1-ShowTooltipsOnForms            | Show/hide tooltips in FintechOS Portal forms.                        |
| .EBSCORE\$1-palette                        | Color palette settings for FintechOS Portal.                         |
| .EBSCORE\$1-theme                          | Theme settings for FintechOS Portal.                                 |
| .EBSCORE\$1-timezone                       | User agent's timezone (moment.js client library).                    |
| .EBSCORE\$1-appBoundaryInteractiveTutorial | Display first-time tutorial in the user interface.                   |

## Self-Hosted Installation

This page describes a simplified self-hosted deployment process of the FintechOS HPFI. This process uses automatic installers to deploy both requirements and the FintechOS HPFI.

### IMPORTANT!

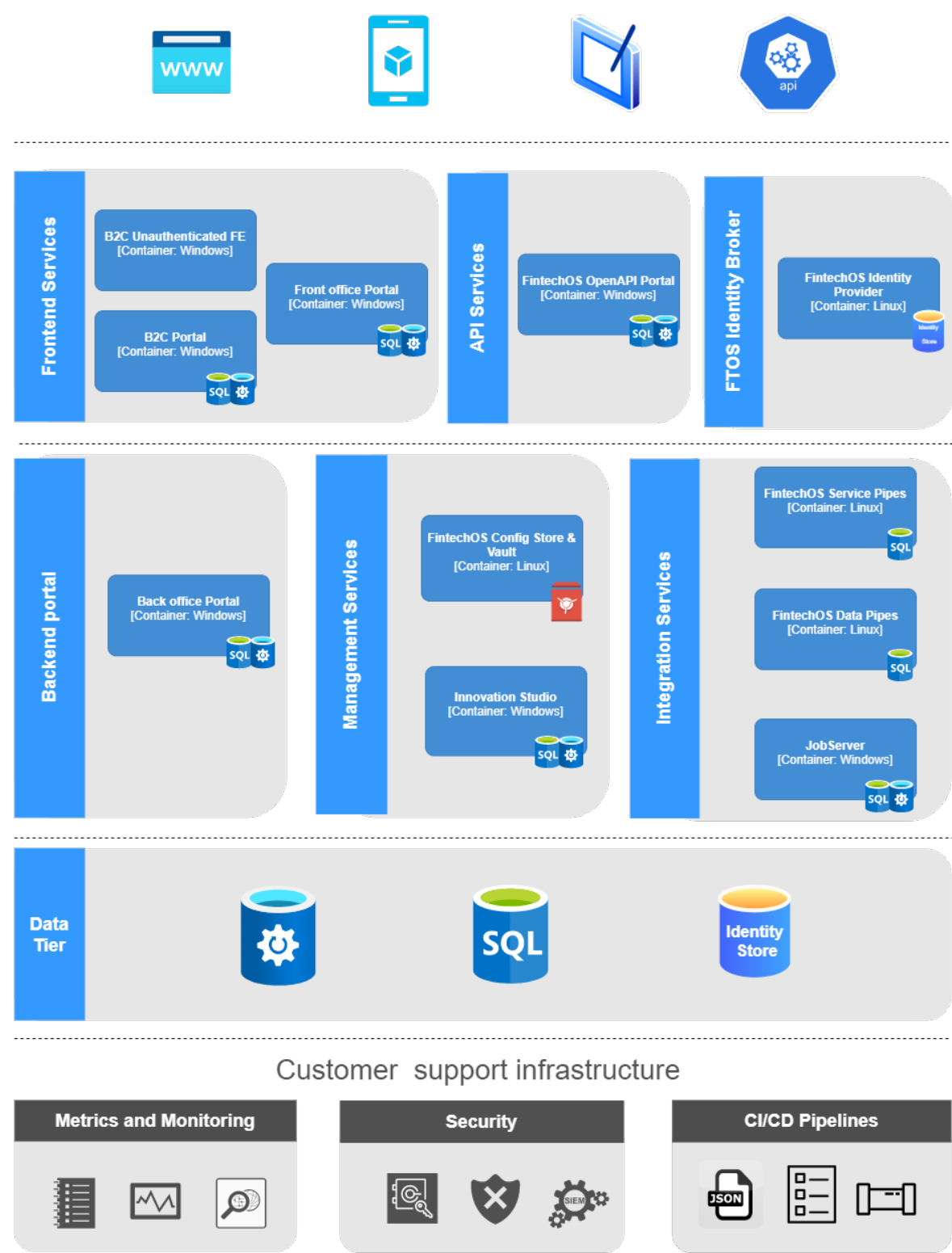
Self-hosted on-premise deployments are intended for development and evaluation purposes. For production environments, we recommend using FintechOS cloud deployments.

## System Requirements

- **OS:** Windows Server Build Number 1809 or Windows 10 Build Number 1903 or Later
- **SQL Server:** SQL Server Express 2016 or SQL Server 2016 or Later
- **Containers:** Docker Engine for Windows and WSL (for Linux Containers)
- **Networking:** Inbound access for the following ports:
  - **TCP:** main\_tcp\_port, iis\_tcp\_port, 1433 (SQL Port), 61616 (ActiveMQ, if selected)
  - **UDP:** 53 (DNS)

## Reference Architecture

Below is a representation of the main components of the FintechOS HPFI.



The automatic installers will install **ONLY** the following components:

- Prerequisites
  - Local Windows prerequisites (Docker Engine and WSL are **not** included)
  - Database configuration on existing SQL Server (SQL Server **not** included)
- Core Components
  - FintechOS Portal
  - [Innovation Studio](#)
  - FintechOS B2C Portal
  - JobServer
  - ["FintechOS Identity Provider" on page 157](#)
  - ["Configuration Manager" on page 86](#)
  - FintechOS MultiFrontEnd
  - FintechOS Proxy
- Additional Business Components
  - [FintechOS API](#)
  - ["FintechOS Service Pipes" on page 109](#)

**IMPORTANT!**

Additional FintechOS Portals are not included in the automatic installation.

## Pre-Installation Checklist

Before you begin the installation, make sure you have the following:

- Administrator privileges on the system where you wish to install the HPFI.
- The **install\_FintechOS.zip** installation scripts.

- The .zip file containing the HPFI version you wish to install available from the [Release Hub](#).

**IMPORTANT!**

The automatic installers were tested on Windows 10 and Windows Server 2019.

## Installation Steps

1. Unzip the *install\_FintechOS.zip* archive.
2. Generate an SSL certificate for your Fully Qualified Domain Name (FQDN):
  - a. Find the `fqdn` parameter in the *parameters.json* file and assign the desired FQDN.
  - b. Use LetsEncrypt ([win-acme tool](#)) to generate the certificate. For details, see ["Generate an SSL Certificate with win-acme" on page 74](#).
3. Create an entry for your FQDN by adding a local DNS lookup to the Windows *hosts* file:
  - a. Open the *C:\Windows\System32\drivers\etc\hosts* file in a text editor.
  - b. Add the IPv4 address of your machine where you want to install HPFI and

the desired FQDN. E.g.:

```
192.168.10.11    devHpfi.com
```

4. Open the parameters.json file in a text editor and edit the following parameters:

parameters.json Reference

| Parameter Section           | Parameter Name            | Parameter Type | Parameter Description  | Parameter Value Example |
|-----------------------------|---------------------------|----------------|--|-------------------------|
| <code>general_config</code> | <code>project_name</code> | String         | Project Name that will be included in IIS App Pool, IIS Application, Docker Containers, and Network, Configuration for all FintechOS components. | "MyAwesomeBankQA"       |

| Parameter Section | Parameter Name   | Parameter Type | Parameter Description  | Parameter Value Example |
|-------------------|------------------|----------------|--|-------------------------|
|                   | release_kit_path | String         | Local PATH where FintechOS release kit is located. This must be an absolute path. The path separator is "\\" (double backslash). | "D:\\FTOS-v22.1.0.0"    |

| Parameter Section | Parameter Name    | Parameter Type | Parameter Description  | Parameter Value Example |
|-------------------|-------------------|----------------|--|-------------------------|
|                   | installation_path | String         | Local PATH where FintechOS components will be installed. You do not need to create the folder in advance. This must be an absolute path. | "C:\\\\FintechOS"       |

| Parameter Section     | Parameter Name | Parameter Type | Parameter Description   | Parameter Value Example |
|-----------------------|----------------|----------------|---|-------------------------|
| additional_components | b2c            | Boolean        | <ul style="list-style-type: none"><li>• <b>true</b><br/>- install B2C components</li><li>• <b>false</b><br/>- do not install B2C components</li></ul> Default: <b>false</b> | <b>true or false</b>    |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description   | Parameter Value Example |
|-------------------|----------------|----------------|---|-------------------------|
|                   | openapi        | Boolean        | <ul style="list-style-type: none"><li>• <b>true</b><br/>- install the Fintech OS API component</li><li>• <b>false</b><br/>- do not install the Fintech OS API component</li></ul> | true or false           |

| Parameter<br>Section | Parameter<br>Name | Parameter<br>Type        | Parameter<br>Description | Parameter Value<br>Example |
|----------------------|-------------------|--------------------------|--------------------------|----------------------------|
|                      |                   | Default:<br><b>false</b> | ent<br>or false          |                            |

| Parameter Section          | Parameter Name | Parameter Type | Parameter Description      | Parameter Value Example |
|----------------------------|----------------|----------------|----------------------------|-------------------------|
| <code>service_pipes</code> | Boolean        | Boolean        | Indicates whether the "Fin | true or false           |

| Parameter Section | Parameter Name | Parameter Type                                | Parameter Description | Parameter Value Example |
|-------------------|----------------|---|-----------------------|-------------------------|
|                   |                | te ch OS<br>Server<br>vice<br>P ip es<br>" on |                       |                         |

| Parameter<br>Section | Parameter<br>Name | Parameter<br>Type   | Parameter<br>Description | Parameter Value<br>Example |
|----------------------|-------------------|---|--------------------------|----------------------------|
|                      |                   | element<br>109<br>component<br>element<br>t<br>• fa<br>l<br>s<br>e<br>- |                          |                            |

| Parameter<br>Section | Parameter<br>Name | Parameter<br>Type      | Parameter<br>Description | Parameter Value<br>Example |
|----------------------|-------------------|------------------------|--------------------------|----------------------------|
|                      |                   | do not install the "Fi |                          |                            |

| Parameter<br>Section | Parameter<br>Name | Parameter<br>Type                                  | Parameter<br>Description | Parameter Value<br>Example |
|----------------------|-------------------|--|--------------------------|----------------------------|
|                      |                   | ntechOS<br>Server<br>vice<br>ce<br>pipes<br>"<br>o |                          |                            |

| Parameter Section | Parameter Name | Parameter Type   | Parameter Description | Parameter Value Example |
|-------------------|----------------|--|-----------------------|-------------------------|
|                   |                | aggregate<br><br>109<br><br>component<br><br>Default:<br>false |                       |                         |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description | Parameter Value Example |
|-------------------|----------------|----------------|-----------------------|-------------------------|
| jobserver         | Boolean        | Boolean        | true or false         |                         |

| Parameter Section | Parameter Name | Parameter Type                   | Parameter Description | Parameter Value Example |
|-------------------|----------------|----------------------------------|-----------------------|-------------------------|
|                   |                | review component of the database |                       |                         |

| Parameter Section | Parameter Name | Parameter Type               | Parameter Description | Parameter Value Example |
|-------------------|----------------|------------------------------|-----------------------|-------------------------|
|                   |                | optional instance identifier |                       |                         |

| Parameter Section | Parameter Name | Parameter Type   | Parameter Description | Parameter Value Example |
|-------------------|----------------|--|-----------------------|-------------------------|
|                   |                | configuration  |                       |                         |
|                   |                | All JobServer services will be installed (Standard, OCB, MC). Default: false |                       |                         |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description | Parameter Value Example |
|-------------------|----------------|----------------|-----------------------|-------------------------|
| active_mq         | Boolean        | Boolean        | true or false         |                         |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description | Parameter Value Example |
|-------------------|----------------|----------------|-----------------------|-------------------------|
|                   |                | environmental  |                       |                         |

| Parameter Section | Parameter Name | Parameter Type                  | Parameter Description | Parameter Value Example |
|-------------------|----------------|---------------------------------|-----------------------|-------------------------|
|                   |                | • fail-safe - do not insist all |                       |                         |

| Parameter Section | Parameter Name | Parameter Type                   | Parameter Description | Parameter Value Example |
|-------------------|----------------|----------------------------------|-----------------------|-------------------------|
|                   |                | the Active Member for Job Server |                       |                         |

| Parameter Section | Parameter Name | Parameter Type   | Parameter Description | Parameter Value Example |
|-------------------|----------------|--|-----------------------|-------------------------|
|                   |                | This option is available only if the <code>jobs</code> <code>server</code> parameter is set to <b>true</b> . |                       |                         |

| Parameter Section  | Parameter Name | Parameter Type | Parameter Description   | Parameter Value Example |
|--------------------|----------------|----------------|---|-------------------------|
| secrets_<br>config | host_password  | String         | The password for the default user <i>host</i> .<br>Default: <b>1234567</b>                                  | "1234567"               |
|                    | idp_admin_user | String         | The admin user for the "FintechOS Identity Provider" on <a href="#">page 157</a> .<br>Default: <b>admin</b> | "admin"                 |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description  | Parameter Value Example |
|-------------------|----------------|----------------|--|-------------------------|
|                   | idp_admin_pass | String         | The admin password for the "FintechOS Identity Provider" on <a href="#">page 157</a> .<br>Default: <b>admin1</b>                   | "admin1"                |
|                   | idp_realm      | String         | The realm name for the HPFI.<br>Default: <b>fintechOS Realm</b>  | "fintechOSRealm"        |
|                   | idp_ftos_user  | String         | The user for the HPFI interaction with the "FintechOS Identity Provider" on <a href="#">page 157</a> .<br>Default: <b>mgmtuser</b> | "mgmtuser"              |

| Parameter Section | Parameter Name        | Parameter Type | Parameter Description  | Parameter Value Example |
|-------------------|-----------------------|----------------|--|-------------------------|
|                   | idp_<br>ftos_<br>pass | String         | The password for HPFI interaction with the "FintechOS Identity Provider" on <a href="#">page 157</a> . Policy: 8 characters, 1 small case, 1 capital, 1 number, and 1 special character minimum. Default: <b>Pa\$\$word123</b> | "Pcfsd433!usF\$"        |

| Parameter Section   | Parameter Name                   | Parameter Type | Parameter Description  | Parameter Value Example |
|---------------------|----------------------------------|----------------|--|-------------------------|
| database_<br>config | sql_<br>conn_<br>server_<br>name | String         | The IPv4 address of the SQL Server. Not accepted localhost or -. | "10.0.0.4"              |
|                     | sql_<br>conn_<br>server_<br>port | Number         | The TCP port of the SQL Server. Default: <b>1433</b>             | "1433"                  |
|                     | sql_<br>conn_<br>auth_<br>user   | String         | The SQL Server admin user.                                       | "sa"                    |
|                     | sql_<br>conn_<br>auth_<br>pas    | String         | The SQL Server admin password.                                   | "Pcfsd433!usF\$"        |

| Parameter Section  | Parameter Name           | Parameter Type | Parameter Description   | Parameter Value Example |
|--------------------|--------------------------|----------------|---|-------------------------|
| network_<br>config | main_<br>tcp_<br>port    | Number         | TCP port for exposing the HPFI front-end.                             | "10000"                 |
|                    | iis_<br>tcp_<br>port     | Number         | TCP port for FintechOS Portal and Innovation Studio back-end.         | "8085"                  |
|                    | iis_<br>website_<br>name | String         | The IIS website name used for FintechOS Portal and Innovation Studio. | "FintechOS"             |

| Parameter Section | Parameter Name                       | Parameter Type | Parameter Description  | Parameter Value Example |
|-------------------|--------------------------------------|----------------|--|-------------------------|
|                   | <code>docker_network_k_subnet</code> | String         | The subnet for the Docker network where all the containers will be connected. A minimum \28 CIDR Prefix network is required. Do NOT overlap with <code>sql_conn_server_name</code> IPv4 address. | "172.16.0.0/25"         |

| Parameter Section | Parameter Name                    | Parameter Type | Parameter Description   | Parameter Value Example     |
|-------------------|-----------------------------------|----------------|---|-----------------------------|
|                   | <code>fqdn</code>                 | String         | The FQDN used for configuring and exposing HPFI. <del>localhost</del> is not accepted if <a href="#">FintechOS API</a> or <a href="#">"FintechOS Service Pipes"</a> on <a href="#">page 109</a> are required. | "localhost"                 |
|                   | <code>cert_full_chain_path</code> | String         | SSL certificate chain file for FQDN. <i>.pem</i> format required.   | "D:\\certs\\cert_chain.pem" |

| Parameter Section | Parameter Name | Parameter Type | Parameter Description   | Parameter Value Example    |
|-------------------|----------------|----------------|---|----------------------------|
|                   | cert_path      | String         | SSL certificate file for FQDN. <i>.pem</i> format required.     | "D:\\certs\\cert_ crt.pem" |
|                   | cert_key_path  | String         | SSL certificate key file for FQDN. <i>.pem</i> format required. | "D:\\certs\\cert_ key.pem" |

| Parameter Section   | Parameter Name                              | Parameter Type | Parameter Description   | Parameter Value Example            |
|---|---|----------------|---|------------------------------------|
| <div><code>docker_config</code></div> <div><div>IM-<br/>PO-<br/>RT-<br/>AN-<br/>T!</div><div>Do<br/>NOT<br/>chan<br/>ge<br/>the<br/>defa<br/>ult<br/>para<br/>met<br/>ers<br/>in<br/>the<br/>dock<br/>er_<br/>confi<br/>g<br/>secti</div></div> | <code>docker_<br/>registr<br/>y_name</code> | String         | Docker Registry URL where all the images for FintechOS Identity Provider, Configuration Manager, Envoy, FintechOS APIs, and Service Pipes are stored. | <code>"crvd poc.azurecr.io"</code> |

| Parameter Section                                    | Parameter Name | Parameter Type | Parameter Description | Parameter Value Example |
|--|----------------|----------------|-----------------------|-------------------------|
| on unless you are deploying a custom advanced setup. |                |                |                       |                         |

| Parameter Section | Parameter Name                    | Parameter Type | Parameter Description  | Parameter Value Example            |
|-------------------|-----------------------------------|----------------|--|------------------------------------|
|                   | <code>docker_registry_user</code> | String         | Docker Registry user with pull rights.                                   | "crvd poc"                         |
|                   | <code>docker_registry_pass</code> | String         | Docker Registry password for user with pull rights.                      | "SkoOC7UilpJzvz8f806r1Yrr=bb6yVvF" |
|                   | <code>docker_idp_image</code>     | String         | Docker image name and tag for "FintechOS Identity Provider" on page 157. | "IDP15upg:standardthemes"          |

| Parameter Section | Parameter Name                  | Parameter Type | Parameter Description  | Parameter Value Example      |
|-------------------|---------------------------------|----------------|--|------------------------------|
|                   | <code>docker_vault_image</code> | String         | Docker image name and tag for "Configuration Manager" on <a href="#">page 86</a> . | "vaultwithcreatefile:latest" |
|                   | <code>docker_proxy_image</code> | String         | Docker image name and tag for FintechOS Proxy (Envoy).                             | "envoy2022start:latest"      |

| Parameter Section | Parameter Name                          | Parameter Type | Parameter Description   | Parameter Value Example     |
|-------------------|---|----------------|---|-----------------------------|
|                   | <code>docker_mfe_image</code>           | String         | Docker Image Name and Tag for the FintechOS Multi Frontend component.                                 | "mfe-common:latest"         |
|                   | <code>docker_openapi_image</code>       | String         | Docker image name and tag for <a href="#">FintechOS API</a> .   | "openapi:latest"            |
|                   | <code>docker_service_pipes_image</code> | String         | Docker image name and tag for <a href="#">"FintechOS Service Pipes"</a> on <a href="#">page 109</a> . | "service-pipes-core:latest" |

5. Install the Windows prerequisites using the *00\_install\_prerequisites.ps1* script:
  - a. Open a PowerShell console as Administrator and navigate to the installation folder containing the installation files.
  - b. Run

```
.\00_install_prerequisites.ps1
```
  - c. Wait until you get a success message.
6. Install the HPFI using the *01\_install\_self-hosted-platform.ps1* script:
  - a. Open a PowerShell console as Administrator and navigate to the installation folder containing the installation files.
  - b. Run

```
.\01_install_self-hosted-platform.ps1 --operation install
```
  - c. Wait until you get a success message.
7. Navigate to installation path folder and check the *installation\_summary.txt* file.

## System Digital Solution Packages Installation

The steps below describe how to perform both an automatic installation and a manual import of a FintechOS SysPack.

Depending on the FintechOS platform version that you want to install, make sure the correct SysPack type is applied:

1. For standard FintechOS infrastructure installation use Standard Syspacks.
2. For Professional/ Enterprise FintechOS infrastructure installation use the following SysPacks:
  - a. Banking environments: Professional Banking SysPacks
  - b. Insurance environments: Professional Insurance SysPacks

**NOTE**

SysPacks are mutually exclusive. The platform installation requires only one SysPack type.

Below are the components for each FintechOS SysPack.

| Package   | Description   |
|---|---|
| 01 FTOS DFP Common.zip                              | FTOS DFP Common Data Model                                |
| 02 FTOS Content Templates.zip                       | FTOS Content Templates                                    |
| 02 FTOS Foundation.zip                              | FTOS Foundation   |
| 02 FTOS Project HyperPersonalization.zip            | FTOS Hyperpersonalization Processor Data Model            |
| 02 FTOS Versioning.zip                              | FTOS Versioning PreReq                                    |
| 03 FTOS Project Campaign.zip                        | FTOS Campaign Management Data Model                       |
| FTOS Project Cognitive Processor Client.zip         | FTOS OCR and Onfido Processors Scripts                    |
| FTOS Project Cognitive Processor Operator.zip       | N/A   |
| FTOS Project Data Governance Consent Management.zip | N/A   |
| FTOS Project Data Governance Sensitive Data.zip     | FTOS Data Governance Sensitive Data Management Data Model |
| FTOS Project Digital Review.zip                     | N/A   |
| FTOS Project Esign Processor.zip                    | FTOS Esign Processor Scripts                              |
| FTOS Project Integration.zip                        | FTOS Integration Scripts                                  |

**HINT**

Details about each component are in their .zip packages.

For an automatic installation, follow the steps described in the **SysPacks Automatic Installation** section.

**IMPORTANT!**

Starting with HPFI V20.2.9, the SysPack can be imported asynchronously. When importing the packages in an Azure environment, always use `async` syntax.

## Prerequisites

In order to install the SysPacks, you need the latest FintechOS platform version installed, with the database configured. For specific steps, see the [Installation](#) page.

**NOTE**

When using **FtosSysPackageDeployer** with SQL Server Integrated Authentication make sure:

1. The Windows user running the above command has read/ write rights access to the FTOS database.
2. You run the command without the SQL username/ password parameters.

When using **FtosSysPackageDeployer** with SQL Server Build In Authentication make sure:

1. The login used has read/ write access to the FTOS database.
2. You run the above command with the SQL username/ parameters.

## Pre-Installation Checklist

The SysPack has unique constraints on some of the standard entities like: FTOS\_DFP\_FlowSettings, FTOS\_DFP\_ProcessorSettings, FTOS\_VersionSettings, FTOS\_VersionSettingsItem, FTOS\_EntityStatusSettings, FTOS\_MKT\_AudienceSegments, FTOS\_MKT\_Audience.

If you have already moved data using the Configuration Data Deployment Package menu, then you probably have already configured some unique constraints.

Before running the script, make sure you:

1. Disable the constraints that you have created on your environment, allowing the system to create the new ones after the SysPacks are imported.
2. Use the new **Configuration Data Definitions** imported with the SysPacks when you export the data.

## Automatic Installation Steps

1. Download the desired SySDigitalSolutionPackages compatible with you platform version from the [Release Hub](#).
2. Unzip the installation kits.
3. Use *FtosSysPackageDeployer* to install the Syspack as follows:
  - Locate the *FtosSysPackageDeployer* in the unzipped FintechOS installation kit at the following location: **<unzipped\_install\_archive>\Tools\FtosSysPkgDeployer**.
  - Navigate to the location where you have unzipped the SysPack and copy the *FtosSysPackageDeployer* here. Let's call this location **<pckg\_deployer\_dir>**.
  - Open `async install_SysPackDA.bat` to edit and replace the parameters described in the [install\\_SysPack.bat Parameters Explanation](#) section, with your own values.
  - Right-click `async install_SysPackDA.bat` » Run as administrator.

## install\_SysPackDA.bat Parameters Explanation

For [asynchronous import](#) run the following command:

```
FtosSysPkgDeployer.exe -i -a -s <studio_url> -u <studio_user_name> -p <studio_user_password> -z <db_Server> -v <db_server_login_username> -k <db_server_login_password> -d <db_name> -r <syspack_file_path>
```

| Field                      | Description  |
|----------------------------|--|
| <studio_url>               | The web URL of the Innovation Studio installation, for example http://localhost/ftos_studio.   |
| <studio_user_name>         | 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 |
| <studio_user_password>     | The password for the Innovation Studio user.   |
| <db_server>                | 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.   |
| <db_server_login_username> | The login username of the SQL Server user with administration rights on the FintechOS installation database.                                       |
| <db_server_login_password> | The password for the above mentioned SQL user.   |
| <db_name>                  | The name of the database where the CoreBanking_3.0.2 is deployed.  |
| <sypack_file_path>         | The physical path to the unzipped SysPack previously downloaded.   |

**HINT**

For more information about the deployment tool, please run FtosSysPackageDeployer.exe without any arguments to see the built-in help

## Post-Installation Setup

After installing the .zip packages, access the 100\_AfterImportManualCopy folder and follow the below steps:

1. Add the following images to the Upload EBS folder <portal\_EBS\_folder> (the Portal with operator flow):
  - a. <syspack\_file\_path>\**100\_AfterImportManualCopy**  
    \CopyToUploadEBS\emptyOCR.jpg
  - b. <syspack\_file\_path>\**100\_AfterImportManualCopy**\CopyToUploadEBS\emptyPhoto.png
2. Copy the following folders over the FintechOS Portal installation directory for every Portal with back-office or B2C installed.
  - a. <syspack\_file\_path>\ **100\_AfterImportManualCopy** \FTOS Project Cognitive Processor Files\dcx-sdk-version\custom
  - b. <syspack\_file\_path>\ **100\_AfterImportManualCopy** \FTOS Project Cognitive Processor Files\dcx-sdk-version\custom-on-demand
  - c. Copy any other needed js files in the corresponding js folder.
  - d. For Onfido, follow the instructions from the **InstallGuideOCRWithOnfido v1.1** file.

## Cognitive Processor Custom Folders Explanation

| Folder           | Description  |
|------------------|--|
| custom           | Contains the video custom components: <ul style="list-style-type: none"> <li>css, images, and javaScripts: dcs-sdk.js and onfido.min.js</li> </ul> |
| custom-on-demand | Contains the liveness component resources.   |









**HINT**

For any other information about the steps performed and their result, check `<pkg_deployer_dir>\Logs`.

## Manual Import Installation Steps















Follow the below steps if you choose to import the SysPack's individual deployment packages by hand.

1. Import the projects in SySDigitalSolutionPackages. Log into Innovation Studio and navigate to **Configuration Management > Deployment > Digital Solution Packages**,
2. Click **Import Digital Solution Package** and select the zip packages in the order set by their names and import them one by one.

|  |                            |
|--|----------------------------|
|  07 FTOS Project Campaign.zip             | Compressed (zipped) Folder |
|  06 FTOS Project HyperPersonalization.zip | Compressed (zipped) Folder |
|  02 FTOS Versioning.zip                   | Compressed (zipped) Folder |
|  02 FTOS Foundation.zip                   | Compressed (zipped) Folder |
|  02 FTOS Content Templates.zip            | Compressed (zipped) Folder |
|  02 FoundationPreInstall.zip              | Compressed (zipped) Folder |
|  01 FTOS DFP Common.zip                   | Compressed (zipped) Folder |
|  00 PreInstall DFP Common.zip             | Compressed (zipped) Folder |

- Run the SQL scripts found in the folders that have a part of the name of the packages.

For example **02 FTOS Foundation.zip** and **FTOS Foundation**.

|  |                            |        |    |
|--|----------------------------|--------|----|
|  02 FTOS Versioning.zip                         | Compressed (zipped) Folder | 19 KB  | No |
|  02 FTOS Foundation.zip                         | Compressed (zipped) Folder | 221 KB | No |
|  02 FTOS Content Templates.zip                  | Compressed (zipped) Folder | 42 KB  | No |
|  02 FoundationPreInstall.zip                    | Compressed (zipped) Folder | 2 KB   | No |
|  01 FTOS DFP Common.zip                         | Compressed (zipped) Folder | 51 KB  | No |
|  00 PreInstall DFP Common.zip                   | Compressed (zipped) Folder | 2 KB   | No |
|  PreInstall DFP Common                          | File folder                |        |    |
|  FTOS Versioning PreReq                         | File folder                |        |    |
|  FTOS Onfido Processor Scripts                  | File folder                |        |    |
|  FTOS OCR Processor Scripts                     | File folder                |        |    |
|  FTOS Integration Scripts                       | File folder                |        |    |
|  FTOS Hyperpersonalization Processor Data Mo... | File folder                |        |    |
|  FTOS Foundation                                | File folder                |        |    |
|  FTOS Esign Processor Scripts                   | File folder                |        |    |

#### NOTE

If you need to update certain packages from the SysPacks, import the .zip files for those packages and run the sql scripts from the folder.

## Self-Hosted Migration from v21.x to v22.x

### 1 Environment Deployment

- Back up your 21.x database and web sites.
- Go to **Control Panel > System and Security** and, from the **Administrative Tools** section, use the **Create and format hard disk partitions** tool to rename your secondary drive from E to F.
- Open the **SQL Server Configuration Manager** and make sure SQL Server (MSSQLSERVER) is running.

4. Run the `.\infra-deployments-feature-refactor-self-hosted-CI-504\change\00_install_prerequisites.ps1` PowerShell script.
5. Log in to the SQL Server Management Studio using the *sa* account (password: *Fintechos2022!*).
6. Import the database you backed up at step 1 on the new environment. The database name must match the following naming convention: *DbFintchOsProjectNameEnv* (e.g. *DbFintechOsCoreBankingDev*).
7. Upgrade the new database using the 22.x version kit.
8. Add any custom keys (values) from the 21.x web.config files (Innovation Studio and FintechOS Portal), in the corresponding 22.x web.config files.
9. Open `.\infra-deployments-feature-refactor-self-hosted-CI-504\01_install_self_hosted_platform` in a text editor and replace the `config_ftos_database -operation $operation` line with `config_ftos_database -operation "upgrade"`.
10. Generate an SSL certificate for your FQDN (see ["Generate an SSL Certificate with win-acme" on page 74](#)).
11. Copy the .pem files in the `.\scripts\certificates` folder.
12. Create an entry for your FQDN by adding a local DNS lookup to the Windows *hosts* file:
  - a. Open the `C:\Windows\System32\drivers\etc\hosts` file in a text editor.
  - b. Add the IPv4 address of your machine where you want to install HPFI and the desired FQDN. E.g.:
 

```
192.168.10.11    devHpfi.com
```
13. Open the `.\infra-deployments-feature-refactor-self-hosted-CI-504\parameters.json` file and fill in all parameters as in the following image. You should change the values for the *general\_config* parameters and for the *sql\_conn\_server\_name* parameter, and you

should use the IPv4 address of your virtual machine. For details, see "[parameters.json Reference](#)" on page 25.

14. In PowerShell (administrator mode), run the following command `.\01_install_self_hosted_platform.ps1 -operation "install"`.
15. Migrate the users and roles to the "[FintechOS Identity Provider](#)" on page 157 (see "[Migrate User Accounts and Roles to the FintechOS Identity Provider](#)" on page 77).

## 2 Install Additional FintechOS Portal Instances

If you have multiple FintechOS Portal instances installed on your environment, follow the instructions below to set up the additional portals. Let's assume that the main FintechOS Portal instance is called *ProjectNameENVAPI*.

1. Run the following command in PowerShell (administrator mode).

```
& $kit_path\PortalWebApp\PortalWebAppInstaller.ps1 -p_
MainCommand Install -p_InstallDir
$installPath\ProjectNameENVAPI-p_IisWebSite "default web
site" -p_IisApp ("ProjectNameENVAPI") -p_IisAppPool
("ProjectNameENVAPI ") -p_DbConnServer $p_DbConnServer -p_
DbConnSqlAuthUser $p_DbConnSqlAuthUser -p_DbConnSqlAuthPass
$p_DbConnSqlAuthPass -p_DbConnDb $p_DbConnDb -p_UploadBSEDir
$upload_bs_existing_path
```

2. Access the "[Configuration Manager](#)" on page 86 through the following URL `https://fqdn:10000/ui/`.
3. You can find the access token in the `.\FintechOS\Config\vault\secrets.txt` file.
4. Create an exact clone of *ProjectNameENVPortal* directory, with the name of the second portal instance (e.g.: *ProjectNameENVAPI*).
5. Change the **EBSDefaultAuthentication** app-settings value from *FTOSOIDC* to *EBS*.
6. Change all the app-settings values containing *ProjectNameENVPortal* to *ProjectNameENVAPI*.

7. Open the `.\FintechOS\Config\envoy\envoy_config.yaml` file and create a clone of the portal cluster, replacing Portal with API. Then, clone the portal route, replacing `ProjectNameENVPortal` with `ProjectNameENVAPI`.
8. In the `web.config` file, the `vault__workspace__application_name` key must have the following value: `ProjectNameENVAPI`.

```
<add key="vault__workspace__application_name" value="ProjectNameENVAPI" />
```

### 3 Final Checks

- In the "Configuration Manager" on page 86, the following app-settings should have the values:
  - `EBSDefaultAuthentication` : `FTOSOID` (for Innovation Studio and FintechOS Portal) and `EBS` (for API)
  - `feature-bundling` : `1`

- In the `web.config` file, the following app-settings should have the values:

```
<add key="use-local-configuration" value="0" />
```

- The JobServer folder must include `EBS.Core.Data.Plugins.dll` file
- If the JobServer is using local configuration, in the `FTOS.JobServer.Service.exe.config` file, the `vault__uri` key must have no assigned value:

```
<add key="vault__uri" value=" " />
```

- After changes have been made to the Envoy files or the Configuration Manager, you must run the `docker restart container_id` command for the container of that component. Also, you should restart the IIS Server.

# Upgrade Process

This section guides you through the steps required to upgrade the FintechOS HPFI.

## Prerequisites

- The .zip file containing the FintechOS HPFI version you want to upgrade to. For example, for FintechOS HPFI version 21.2.0, should look like **FTOS-CORE-RLS-v21.2.0.0-b87-GOLD.zip**. It also contains the installation script files inside the **install\_FintechOS.zip** file.
- The Deployment Package which contains your existing data. Read more about [Configuring Data Deployment Packages](#).

## Upgrade Steps

### 1 Preparing The Data Deployment Package

The first thing that needs to be done is to create the deployment package which includes all the metadata you need to migrate.

You can create and export a data deployment package as explained on [this page](#).

If, in addition to metadata, you wish to include entity data and report templates in a single deployment package, you need to [create an enhanced deployment package](#).

This generates an .xml file with all the required metadata which needs to be imported after the platform upgrade.

### 2 Upgrading Portal and Studio

#### **IMPORTANT!**

Stop/Deactivate all FTOS services (e.g., stop & switch to manual-start Windows services or Web applications) except the FTOS database.

- Extract the contents of the **install\_FintechOS.zip** located in the FintechOS HPFI installation .zip.
- Open the parameters.ini file, and modify the INSTALLATIONPATH attribute to match the location where the platform is already installed. Close and save the file.
- Run the **PortalInstaller.exe** and **StudioInstaller.exe** files. They are found in **install\_FintechOS.zip**. This upgrades the Portal and Studio components of the platform.

### 3 Upgrading The Services

#### MainDbServer

You must execute instructions from this section on a machine that can connect and execute T-SQL against **MainDbServer**.

1. Execute in cmd.exe: `"ReleaseKit\SQL\BasicDbUpgrader.exe" -s "MainDbServer" -d "MainDb"`

#### NOTE

You should be presented with a report on which scripts will be applied on **MainDb**.

2. Execute in cmd.exe: `"ReleaseKit\SQL\BasicDbUpgrader.exe" -s "MainDbServer" -d "MainDb" -g`
  - This command will upgrade **MainDb** and it will take a few minutes to complete, depending mostly on how old **MainDb\_OldVersionNo** is vs. **ReleaseKit\_VersionNo**

#### PortalWebApp

You must execute instructions from this section on **PortalWebApp\_Machine**.

1. Create a new batch file named **PortalWebAppInstaller.Upgrade.bat** in a directory of your choice and add in the following single command line:  
`powershell.exe -File "ReleaseKit\PortalWebApp\PortalWebAppInstaller.ps1" -p_MainCommand Upgrade -p_InstallDir PortalWebApp_InstallDir`
2. Execute in cmd.exe: `PortalWebAppInstaller.Upgrade.bat`

**HINT**

This command does the following:

- Overwrites all files from **PortalWebApp\_InstallDir** with **PortalWebApp** files from **ReleaseKit**, except for web.config which is not overwritten in case you customized it
- [Re]Starts **PortalWebApp\_lisAppPoolName**

3. Open in a web browser **PortalWebApp\_LoginUrl** and check the page appears as expected.

**DesignerWebApp**

You must execute instructions from this section on **DesignerWebApp\_Machine**.

1. Create a new batch file named **DesignerWebAppInstaller.Upgrade.bat** in a directory of your choice and add in the following single command line:  
`powershell.exe -File "ReleaseKit\DesignerWebApp\DesignerWebAppInstaller.ps1" -p_MainCommand Upgrade -p_InstallDir DesignerWebApp_InstallDir`
2. Execute in cmd.exe: `DesignerWebAppInstaller.Upgrade.bat`

**HINT**

This command does the following:

- Overwrites all files from **DesignerWebApp\_InstallDir** with **DesignerWebApp** files from **ReleaseKit**, except for web.config which is not overwritten in case you customized it
- [Re]Starts **DesignerWebApp\_lisAppPoolName**

3. Open in a web browser **DesignerWebApp\_LoginUrl** and check the page appears as expected.

**JobServer**

You must execute instructions from this section on **JobServer\_Machine**.

1. Execute in cmd.exe: `sc.exe stop JobServer_WinSvcName`
2. Copy with overwrite all files from **ReleaseKit\_Dir\JobServer** to **JobServer\_InstallDir** except for the following files:

- connections.config
  - FTOS.JobServer.Service.exe.config
  - schedule.config
  - services.config
  - serviceSettings.config
3. For each of the files excepted at the previous step, analyze the differences between version from **ReleaseKit\_Dir** and that from **JobServer\_InstallDir** using a text file compare tool and merge changes into the version from **JobServer\_InstallDir** without breaking existing customizations
  4. Execute in cmd.exe: `sc.exe start JobServer_WinSvcName`

#### MessageComposer

You must execute instructions from this section on **JobServer\_Machine**.

1. Follow all steps from JobServer \ Standard job configuration \ Upgrade
2. Execute in cmd.exe: `sc.exe stop JobServer_WinSvcName`
3. Copy with overwrite all files from **ReleaseKit\_Dir\JobServer.Plugins\MessageComposer** to **JobServer\_InstallDir**, except for the following files:
  - connections.config
  - schedule.config
  - services.config
  - serviceSettings.config
4. For each of the files excepted at the previous step, analyze the differences between version from **ReleaseKit\_Dir** and that from **JobServer\_InstallDir** using a text file compare tool and merge changes into the version from **JobServer\_InstallDir** without breaking existing customizations.
5. Execute in cmd.exe: `sc.exe start JobServer_WinSvcName`

## 4 Importing Deployment Packages

The final step is to import all your metadata in the upgraded environment. This makes sure all your data is migrated and ready to be used. Importing is done by running the **SysPkgDeployer** tool from the command line, using the following syntax:

```
CD /D %~dp0 "%~dp0\FtosPkgDeployer\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
```

## Generate an SSL Certificate with win-acme

This guide explains how to generate an SSL certificate for a Fully Qualified Domain Name (FDQN) on a Windows Server machine using the [win-acme](#) tool.

1. Make sure that ports 80 and 443 on your environment allow Internet connectivity.
2. Download the tool from the [official win-acme release page](#).
3. Run *wacs.exe* as administrator.
4. Type **M** to create a certificate with full options.

```
N: Create certificate (default settings)
M: Create certificate (full options)
R: Run renewals (0 currently due)
A: Manage renewals (1 total)
O: More options...
Q: Quit
Please choose from the menu: M
```

5. Type **2** to manually input the domain names included in the certificate.

```
Please specify how the list of domain names that will be
included in the
```

```

certificate should be determined. If you choose for one of
the "all bindings"
options, the list will automatically be updated for future
renewals to
reflect the bindings at that time.
1: Read bindings from IIS
2: Manual input
3: CSR created by another program
C: Abort
How shall we determine the domain(s) to include in the
certificate?: 2

```

6. Enter the FDQN you want to use. E.g.: vm-customer360-dev.westeurope.cloudapp.azure.com

```

Description:          A host name to get a certificate for.
This may be a
                        comma-separated list.
Host: vm-customer360-dev.westeurope.cloudapp.azure.com
Source generated using plugin Manual: vm-customer360-
dev.westeurope.cloudapp.azure.com
Friendly name '[Manual] vm-customer360-
dev.westeurope.cloudapp.azure.com'. <Enter> to accept or type
desired name: vm-customer360-
dev.westeurope.cloudapp.azure.com

```

7. Type **2** to serve the verification files from memory.

```

The ACME server will need to verify that you are the owner of
the domain
names that you are requesting the certificate for. This
happens both during
initial setup *and* for every future renewal. There are two
main methods of
doing so: answering specific http requests (http-01) or
create specific dns
records (dns-01). For wildcard domains the latter is the only
option. Various
additional plugins are available from https://github.com/win-acme/.
1: [http-01] Save verification files on (network) path
2: [http-01] Serve verification files from memory
3: [http-01] Upload verification files via FTP(S)
4: [http-01] Upload verification files via SSH-FTP

```

```

5: [http-01] Upload verification files via WebDav
6: [dns-01] Create verification records manually (auto-renew
not possible)
7: [dns-01] Create verification records with acme-dns
(https://github.com/joohoi/acme-dns)
8: [dns-01] Create verification records with your own script
9: [tls-alpn-01] Answer TLS verification request from win-
acme
C: Abort
How would you like prove ownership for the domain(s)?: 2

```

8. Type **2** to select the RSA key type.

```

After ownership of the domain(s) has been proven, we will
create a
Certificate Signing Request (CSR) to obtain the actual
certificate. The CSR
determines properties of the certificate like which (type of)
key to use. If
you are not sure what to pick here, RSA is the safe default.
1: Elliptic Curve key
2: RSA key
C: Abort
What kind of private key should be used for the certificate?:
2

```

9. Type **2** to store the certificate as PEM encoded files.

```

When we have the certificate, you can store in one or more
ways to make it
accessible to your applications. The Windows Certificate
Store is the default
location for IIS (unless you are managing a cluster of them).
1: IIS Central Certificate Store (.pfx per host)
2: PEM encoded files (Apache, nginx, etc.)
3: PFX archive
4: Windows Certificate Store
5: No (additional) store steps
How would you like to store the certificate?: 2

```

10. Type **2** to insert the path where you wish to store the certificates from the console.  
 E.g.: C:\Users\john.doe\Documents)

```

Description:      .pem files are exported to this folder.
File path:      .
Description:      Password to set for the private key .pem
file.
1: None
2: Type/paste in console
3: Search in vault
Choose from the menu: 2

```

11. Type **1** to disable password protection for the private key file.

```

Description:      Password to set for the private key .pem
file.
1: None
2: Type/paste in console
3: Search in vault
Choose from the menu: 2

```

12. Type **5** to decline any additional store steps.

```

1: IIS Central Certificate Store (.pfx per host)
2: PEM encoded files (Apache, nginx, etc.)
3: PFX archive
4: Windows Certificate Store
5: No (additional) store steps
Would you like to store it in another way too?: 5
Installation plugin IIS not available: This step cannot be
used in combination with the specified store(s)

```

## Migrate User Accounts and Roles to the FintechOS Identity Provider

Starting with release 22.1, the FintechOS HPFI uses the ["FintechOS Identity Provider"](#) on [page 157](#) for identity and access management. In order to facilitate user accounts and roles migration from legacy systems to the FintechOS Identity Provider, an automatic user migration tool has been provided.


**IMPORTANT!**

The FintechOS Identity Provider requires each user to have a unique email address. Users who don't have a unique email address will not be migrated.

To use the migration tool, follow the instructions below:

1. Make sure [.NET 5.0 Desktop Runtime](#) is installed on your system.
2. From the HPFI installation kit, copy the User Migration Tool archive and extract it on your system.

3. In the extracted folder, run the User Migration Tool executable (FTOS.Tools.UserMigrationToIdpUi.exe).



FTOS IDP User Migration Tool

**Local database:**

DB server:

DB user:

DB password:

DB initial catalog:

☐ Use a connection string instead

DB Conn. string:

**IDP Instance:**

IDP URL:

IDP realm:

IDP client id:

IDP client secret:

Save output to file:

Logs: [Copy](#) [Save...](#)

☐ Migrate roles

☐ Overwrite roles

☐ Migrate users

☐ Overwrite users

Test Running will not commit the results!

Test Run!

## 4. Fill in the following information:

| Field                                 | Description   |
|---------------------------------------|---|
| DB server                             | Required if "Use a connection string instead" below is not checked. Database server address of your legacy FintechOS system.  |
| DB user                               | Required if "Use a connection string instead" below is not checked. Database username used to connect to the legacy FintechOS database to download user accounts and user roles.          |
| DB password                           | Required if "Use a connection string instead" below is not checked. Database user password for the above user account.  |
| DB initial catalog                    | Required if "Use a connection string instead" below is not checked. Database name of your legacy FintechOS system.  |
| Use a connection string instead       | Uses the database connection string provided in the "DB Conn. string" below field to connect to the legacy FintechOS system database instead of the above settings.                       |
| DB Conn. string                       | Required if "Use a connection string instead" above is checked. Allows you to provide a database connection string which will be used to connect to the legacy FintechOS system database. |
| IDP URL                               | Required. URL of the FintechOS Identity Provider instance.  |
| IDP realm                             | Required. FintechOS Identity Provider realm set up for the FintechOS HPFI.  |
| IDP client id                         | Required. Client ID set up in the FintechOS Identity Provider for the Innovation Studio.  |
| IDP client secret (v22.1.1 and later) | Required. The client secret associated with the client ID.  |

| Field                                   | Description  |
|---|--|
| IDP admin API username (v22.1 only)     | A user account set up in the FintechOS Identity Provider admin console with realm management role.   |
| IDP admin API secret (v22.1 only)       | Password for the above user account.   |
| Save output to file (v22.1.1 and later) | Specify a .csv file where information regarding the user/roles that have been migrated (together with the migration status) will be saved. If not specified, the output will be directed to the text area below. |
| Command line & results (v22.1 only)     | You can use the automatically generated command to run the users and roles migration from the command line.  |
| Save to log file (v22.1 only)           | Allows you to define a text file where the migration results will be logged.   |

| Field                            | Description  |
|----------------------------------|--|
| Migrate roles                    | <p>Specify whether or not the tool should migrate user roles.</p> <div> <p><b>IMPORTANT!</b></p> <p>In order to properly migrate the users, the roles must be migrated first. Either check both <b>Migrate roles</b> and <b>Migrate users</b> (the tool will first migrate the roles, and then the users) or run the tool first with the <b>Migrate roles</b> option checked, then run it again with the <b>Migrate users</b> option checked. If the tool tries to migrate the users before the roles are migrated, it will fail.</p> </div> |
| Migrate users                    | <p>Specify whether or not the tool should migrate user accounts.</p> <div> <p><b>IMPORTANT!</b></p> <p>User accounts without a valid email address will not be migrated. If another user account with an identical email address already exists in the FintechOS Identity Provider, the user will not be migrated (the legacy database authentication does not enforce unique email addresses for user accounts).</p> </div>   |
| Skip external users (v22.1 only) | Does not migrate external user accounts (users with a valid entry in the MembershipProviderUserId attribute).  |

| Field           | Description  |
|-----------------|--|
| Overwrite roles | Specify whether or not the tool should overwrite matching user roles that already exist in the FintechOS Identity Provider.    |
| Overwrite users | Specify whether or not the tool should overwrite matching user accounts that already exist in the FintechOS Identity Provider. |

5. (v22.1.1 and later) Click **Test Run!**. This will not commit the results. This step performs a dry migration (the tool simulates the actual migration), but the results are not saved.
6. (v22.1.1 and later) After the test run is finished, check the output .csv file and look for roles/users that have the ERROR status. This indicates that the corresponding users/roles will have issues during the real migration. Check the Reason column to determine why a user/role is in an error state and fix the issue.
7. (v22.1.1 and later) If, during the previous step, you had to fix any errors, you can close the tool and run it again to make sure there are no more issues.
8. Click **Run Migration!** This action commits the results and the users/roles are migrated to the FintechOS Identity Provider.

## Check the Migration Output

The test migration cannot identify all the possible errors that might occur during the actual migration. Even if, based on the test migration output, all the users/roles should be migrated successfully, make sure to also check the actual migration output file and verify the statuses of the users/roles that have been migrated. If additional errors (that couldn't be identified in the test run) are present, try to fix them based on the information in the Reason column.

### NOTE

The migration is idempotent and can be run multiple times.

## Troubleshooting

During the test run, the following errors may occur:

- A role that needs to be migrated is found, but the **Overwrite roles** option is not checked. In this case, during the actual migration, the role will be skipped.
- There are users with duplicate emails or usernames in the legacy database. The legacy database and applications support multiple users with identical email addresses or usernames, but the FintechOS Identity Provider doesn't! To fix this, make sure that there are no duplicate email addresses or usernames in the database. If the errors are not fixed, during the actual migration, only the first such encountered user account will be migrated successfully.
- The user does not have an email address. This is a required field in the FintechOS Identity Provider, so users that do not have a valid email address will not be migrated. Make sure that all the users have a valid email address prior to running the actual migration.
- The user does not have a username. This is a required field in the FintechOS Identity Provider, so users that do not have a username will not be migrated. Make sure that all the users have unique usernames assigned prior to running the actual migration.
- A matching username has been found in the FintechOS Identity Provider, but the email address is different.
- A matching email address has been found in the FintechOS Identity Provider, but the username is different.
- Two matches are found in the FintechOS Identity Provider for the user to be migrated: one that matches the username and one that matches the email address.

- A matching user is found in the FintechOS Identity Provider, but the **Overwrite users** option is not checked. During the actual migration, this user will be skipped.
- There is an issue connecting to the database or the FintechOS Identity Provider. This will be notified in the text area of the tool. The users/roles migration will not proceed.
- There is an issue connecting to the database or the FintechOS Identity Provider. This will be notified in the text area of the tool. The users/roles migration will not proceed.
- Neither the **Migrate roles** nor the **Migrate users** option is checked. The migration will stop, as there is no actual migration to be done.
- The test run has finished (the Run Migration button is visible), then one of the migration options changes. The migration tool will reset and the Test Run button will be displayed again. This is to make sure that the test run results are accurate.

# Configuration Manager

Starting with release 22.1, the FintechOS HPFI uses the [HashiCorp Vault](#) secrets management system to store system configurations in a secure and controlled environment. This protects sensitive data, such as system parameters, environment variables, services credentials, or API keys and simplifies user access and environment management.

**NOTE**

When changing system parameters from Innovation Studio, the corresponding Vault secrets will be updated accordingly.

The Vault Agent can be installed either as an Azure web app for cloud deployments or as a Windows service for on-premise deployments.

## Manage Vault Secrets

### Directory Structure

The directory structure of a vault path is described below:

```
kv/<environmentName>/<applicationName>/<node>
```

| Directory         | Designation    | Description  |
|-------------------|----------------|--|
| kv                | Secrets Engine | FintechOS uses Vault's KV version 2 secrets engine to store system configurations as key-value pairs.  |
| <environmentName> | Environment    | You can define different sets of configurations specific to various environments such as development, testing, or production. This allows you to change the configurations for your system in one go by switching the environment directory. |

| Directory         | Designation | Description   |
|-------------------|-------------|---|
| <applicationName> | Application | Directory indicating the type of FintechOS system component such as an Innovation Studio instance, a FintechOS Portal instance, or a FintechOS Identity Provider. |
| <node>            | Node        | Nodes allow you to group your secrets for easy classification and access management.  |

For example:

`kv/production/fintech-os-portal/app-settings`

## Secrets

Within each node, you can define multiple Vault secrets in the form of key-value pairs.

Navigation: Secrets Access Policies Tools | Status

Breadcrumb: < kv < rc < studio < app-settings

### rc/studio/app-settings

Secret Metadata

JSON Delete Copy Version 20 Create new version

| Key  | Value |
|--|-------|
| ADPath                                     | -     |
| APPINSIGHTS_INSTRUMENTATIONKEY             | ***** |
| APPLICATIONINSIGHTS_CONNECTION_STRING      | ***** |
| ApplicationInsightsAgent_EXTENSION_VERSION | ***** |
| AutoAuthDefaultBusinessUnit                | ****  |
| AutoAuthOneTimeKeyExpiration               | ****  |
| AutoAuthRequestExpiration                  | ****  |
| ChangePasswordUnsuccessful                 | ***** |
| ClientValidationEnabled                    | ****  |

Version created Mar 25, 2022 06:04 PM

## Vault Connection

To configure the connection between the Vault Agent and an Innovation Studio or Portal instance, open its *web.config* file in a text editor and, in the `app-settings` node, edit the following keys:

```
<app-settings>
  <add key="vault__uri" value="https://myVaultWebApp" />
  <add key="vault__token" value="myVaultAuthToken" />
  <add key="vault__workspace__application_
environment" value="test" />
  <add key="vault__workspace__application_name" value="fintech-
os-portal" />
</app-settings>
```

| Key  | Value  |
|--|--|
| vault__uri                                 | Address of the Vault Agent web app or Windows service.   |
| vault__token                               | Authentication token created by Vault for the operator used to access the system configurations. |
| vault__workspace__application__name        | Type of FintechOS ecosystem component. See <a href="#">"Application" on the previous page</a> .  |
| vault__workspace__application__environment | Type of environment. See <a href="#">"Environment" on page 86</a> .                              |

## Enable web.config Override

### IMPORTANT!

The Vault secrets management system is the default method for storing system configurations. The web.config override is only intended for development and testing purposes, not for production use.

To control your Innovation Studio or FintechOS Portal application settings from the web.config file instead of Vault, open the *web.config* file in a text editor and, in the `app-settings` node, add or enable the following key:

```
<app-settings>
...
<add key= "feature-development-mode" value="1" />
...
</app-settings>
```

| Key                      | Value  |
|--------------------------|--|
| feature-development-mode | <ul style="list-style-type: none"><li>• 1 - enables web.config override. If an application setting is defined both as a Vault secret and web.config key, the web.config key will take precedence.</li><li>• 0 - disables web.config override</li></ul> |

**IMPORTANT!**  
The `feature-development-mode` key should never be enabled in production, as it has multiple purposes targeted for developers (i.e. extra logging).

# DevOps

DevOps is a set of processes that unifies development (Dev) and processes (Ops) to complement software development. Unlike traditional software development methodologies, DevOps enables companies to create and improve products and go to market at a faster pace.

This section covers the following topics:

---

## Configure the File Upload Folder

When building a web application that requires users to upload or download files (documents, images, etc.), file storage can be an important aspect of the application architecture.

### Where Should I Store Files?

HPFI platform supports multiple storage providers for storing the uploaded or generated user files. When building web applications using HPFI technology, you've got a few choices for where to store your files:

- ["Local File System Storage" on the next page](#)
- ["Azure Blob Storage" on page 93](#)
- ["Amazon S3 Buckets Storage" on page 94](#)

The local file system refers to either a local path on the application server or a shared folder on the network containing the application server. While it is the default storage provider, you might be running out of disk space or you might find it a very challenging task to ensure that files are properly backed up and available at all times.

If you'll be storing large blobs of content, you might want to consider one of the other options. Storing files in a file storage service like Amazon S3 Buckets or Azure Blob is a great option if you'll be storing large blobs of content. Not only you stay rest assured that your data is replicated and backed up, but they also ensure scalability and high availability.

This section walks you through the steps needed to configure the "UploadEbs" storage provider /location as needed.

## Local File System Storage

There are no special configurations that have to be made in order to use it other than setting the name of the root folder.

To set the name of the root folder, go to Vault and add the below secret:

| Key Path                                    | Key Name     | Key Value           |
|---|--------------|---------------------|
| kv/<environment>/<application>/app-settings | UploadFolder | path_to_root_folder |

## (Deprecated) Add key in web.config

Go to the **web.config** file, open it and to the **app-settings** node, add the application setting **UploadFolder**, as described below:

```
<configuration>
...
<app-settings>
...
  <add key="UploadFolder" value="path_to_root_
folder" />
</app-settings>
</configuration>
```

Depending on where the root folder resides, make sure that you properly set the value of the UploadFolder setting:

- subfolder of the application folder: "~/path/to/uploadfolder/";
- local folder on application server, the full path to local folder, like:  
"c:\path\to\uploadfolder"
- network shared folder: "\\server\path\to\uploadfolder";

**NOTE**

If in the **web.config** file you do not set the **UploadFolder** setting, it is automatically set to the default value, that is, "~/UploadEBS/".

## Automatically Create File Upload Subfolders

**IMPORTANT!**

This feature is available only for local file system storage. It is not available for Azure Blob Storage or Amazon S3 Buckets Storage.

You can automatically group uploaded files into folders based on the last three characters in their file name (excluding the file extension). To do so, add an **feature-uploadfolder-autocreate-subfolders** key with a value of **1** in the web.config file:

```
<add key="feature-uploadfolder-autocreate-subfolders" value="1">
```

This will save each uploaded file in a `-.files\xyz` subfolder of the upload folder, where `xyz` represents the last three characters of the file name. For example, a file called `MyDoc_0caf99b6-549d-48f7-8747-5e3eb82753fd.txt` will be saved in a folder structure similar to:

```
...\  
  UploadEBS\  
    -.files\  
      3fd\  
        MyDoc_0caf99b6-549d-48f7-8747-5e3eb82753fd.txt
```

Setting the `feature-uploadfolder-autocreate-subfolders` key value to **0** disables the feature.

This feature is backward compatible. If a requested file is not stored in the above folder structure, it will be read from the main upload folder or the entity specific upload folder respectively.

## Azure Blob Storage

To configure HPFI to store user files in Azure Blob, follow these steps:

1. Go to the web.config file and open it.
2. Add a ftosStorageService section to the **<configSections>** element:

```
<configuration>
  <configSections>
    ...
    <section
name
=
"ftosStorageService"

type
="EBS.Core.Utills.Services.Config.StorageServiceConfigSection,
EBS.Core.Utills"/>
  </configSections>
</configuration>
```

3. Add a ftosStorageService section (note the AzureBlob type) as child of **<configuration>** element:

```
<configuration>
  ...
  <ftosStorageService type="AzureBlob">
    <settings>
      <setting
name="connectionString" value="connection_string"/>
      <setting name="rootContainer" value="root_
container"/>
    </settings>
  </ftosStorageService>
</configuration>
```

where:

- *connectionString* is the connection string FintechOS is using to connect to an Azure Blob container;
- *rootContainer* is the root container name where the user files will be stored.

## Azure Resource Manager templates support

To enable automatic deployment through ARM templates, the `connectionString` and `rootContainer` settings must be configured in the **<app-settings>** element of the `web.config` file:

```
<app-settings>
...
  <add key="ftosStorageService-AzureBlob-
connectionString" value="connection_string" />
  <add key="ftosStorageService-AzureBlob-
rootContainer" value="root_container" />
</app-settings>
```

### IMPORTANT!

Values set in the **<app-settings>** keys take precedence over the values set in the **<ftosStorageService>** settings node.

## Amazon S3 Buckets Storage

To configure HPFI to store user files in Amazon S3 Buckets, follow these steps:

1. Go to the `web.config` file and open it.
2. To the **<configSections>** element, add the following two sections: `ftosStorageService` and `aws`, as described below:

```
<configuration>
  <configSections>
    ...
```

```

        <section
name
=
"ftosStorageService"

type
="EBS.Core.Utills.Services.Config.StorageServiceConfigSection,
EBS.Core.Utills"/>
        <section name="aws" type="Amazon.AWSSection,
AWSSDK.Core"/>
    </configSections>
</configuration>

```

3. Add <ftosStorageService> tag (note the AmazonS3Bucket type) as child of configuration element:

```

<configuration>
    ...
    <ftosStorageService type="AmazonS3Bucket">
        <settings>
            <setting name="AWSAccessKey" value="access_
key" />
            <setting name="AWSSecretKey" value="secret_
key" />
            <setting name="BucketName" value="bucket_
name"/>
        </settings>
    </ftosStorageService>
</configuration>

```

where:

**AWSAccessKey** and **AWSSecretKey** are used by FTOS to sign the requests made to AWS. For more information, see [Access Keys \(Access Key ID and Secret Access Key\)](#).

**BucketName** is the root bucket name where the user files will be stored.

4. Add the **aws** section as child of the configuration element:

```

<configuration>
    ...
    <aws region="aws_region">

```

```
</aws>  
</configuration>
```

**NOTE**

The only required attribute is **region**. For a complete list of available regions, see Amazon documentation, section *Regions, Availability Zones, and Local Zones*. The region attribute must have one of the values from the column "Region". E.g.: `<aws region="eu-central-1"></aws>`

For a list of allowed elements in the AWS section, see [Configuration Files Reference for AWS SDK for .NET](#).

## Importing and Exporting Deployment Packages

In HPFI, users with elevated privileges (admin users) can export metadata from an environment and import it into another environment, using deployment packages.

Deployment packages are text-based so they can be version controlled to have their history inspectable with text-diff tools.

In HPFI, you have three options for importing and exporting deployment packages, as follows:

- In Innovation Studio, from the DevOps menu > Deployment Packages. For more information, see the Innovation Studio, section [Deployment Packages](#).
- Using the customization set methods of the API. For information on how to import and export packages using customization sets), see [HPFI API documentation](#).
- From the command line by using the **FtosPkgDeployer** tool.

The **FtosPkgDeployer** tool is available in the release subdirectory, `\Tools\FtosPkgDeployer`, It allows you to do the following from the command prompt:

- list the customization sets found in a FTOS server.
- import / export in / from server a customization set from / in a local file.

**NOTE** :In order to use the tool, make sure to run the command prompt as admin.

For information on how to use the **FtosPkgDeployer** tool , see the built-in help by running the command prompt as admin and executing `FtosPkgDeployer.exe` without arguments. The tool is using the POST CUSTOMIZATION SET method of the HPFI API.

## File-Type Upload Control

In HPFI, you can control what types of files users can upload into the system.

This feature is particularly useful in preventing users from uploading wrong file types, thus saving time from investigating what went wrong and having to resubmit the files.

**NOTE** The file-type upload control feature has been added to the previous existing validations: file extension validation, content size validation etc. For a content to be uploaded all validations must pass.

### Enable the file-type upload control

By default, the file-type upload control is disabled. To enable it, add the following secret in Vault:

| Key Path                                    | Key Name                      | Key Value |
|---|-------------------------------|-----------|
| kv/<environment>/<application>/app-settings | feature.upload.filetype-check | true      |

(Deprecated) Add configuration in web.config files:

```
<app-settings>
  ....
  <add key="feature.upload.filetype-check" value="true" />
</app-settings>
```

## File-Type Upload Processing

Once the File-Type Upload control is enabled, upon file uploads using client scripts (using the `ebs.upload` function) or server automation scripts (using the `uploadFile` function), the system verifies the uploaded content against the file extension. The system will try to match the uploaded content (the bytes) with the provided file extension based on a list of files signatures.

Files signatures are available for the following file types: pdf, docx, xlsx, pptx, odt, ods, jpg/jpeg, doc, xls, ppt, rtf, xml, png, gif, bmp.

### No match, the file is uploaded

If the matching process does not find any match between the file content and the available file signatures then the upload is allowed.

The user uploads an Autocad file.

### Match, but the signature's extension is not what the file says it is

if the matching process finds a match between the file extension and the available file signature, the system further checks the file internal type (that's is, MIME type) which serves as an integrity check. If there is a mismatch between the two, that means that the internal type of the file does not correspond to what the file extension says it is and the file upload is not allowed. An error will be returned.

The user tries to upload a PNG file (the content has a PNG signature) that has a ".jpg" extension

### Executable files

By design, if the matching process identifies that the uploaded content has an EXE or DLL signature then the upload is not allowed. An error will be returned.

## HPFI API a Standalone Web App

HPFI gives you the ability to set the HPFI API as a standalone web app, which means that the API it will work in exclusive API server mode. This is particularly useful when you want to get data from HPFI using API calls and use it within your own web apps. The following controllers are available via the API standalone app: WCF Services, API and Authorization.

**NOTE** The Portal functionality is disabled, that means that the API web app will not be available to end users.

To configure the API as a standalone app, go to the **web.config** file and enable the API server, as provided below:

| Key Path                                    | Key Name           | Key Value |
|---|--------------------|-----------|
| kv/<environment>/<application>/app-settings | feature-api-server | 1         |

### (Deprecated) Add configuration in web.config files:

```
<configuration>
  <app-settings>
    ...
    <add key="feature-api-server" value="1" />
  </app-settings>
  ...
</configuration>
```

## Activating Localization Debug Mode

HPFI supports the localization of static and dynamic elements, as well as the localization of customized messages and metadata.

Before localizing in a new language, you need to prepare your environment to easily identify the resources to be localized. To do so, open Vault secrets change the value of the following:

From:

| Key Path                                    | Key Name              | Key Value |
|---|-----------------------|-----------|
| kv/<environment>/<application>/app-settings | ebs:debugLocalization | 0;        |

To:

| Key Path                                    | Key Name              | Key Value |
|---|-----------------------|-----------|
| kv/<environment>/<application>/app-settings | ebs:debugLocalization | 1;        |

The table below lists the localization keys:

| Key                               | Description  |
|-----------------------------------|--|
| ebs:uiLocalization                | Toggles dynamic UI localization (HTML templates, after generate JavaScript).<br>The immediate result visible in the user interface is the dynamic generation of HTML templates in the optimum format for localization. |
| ebs:dataLocalization              | Toggles metadata (data) localization and automatically creates database support for each language.   |
| ebs:debugLocalization             | Toggles the debug mode.  |
| ebs:localizationSchedulerTimeSpan | Interval in milliseconds when checking for external resource updates in the database.  |

Once the localization debug mode is activated, the following markers are displayed for both the user interface and the metadata localized values:

✓ - To indicate the localized values.

❓ - To indicate that values are localizable, but no localization has been provided.

🔑 - To indicate that the fields allow localization of the data inside the field.

## (Deprecated) Configuring keys in web.config files

In web.config files, change the value for the below keys:

From:

```
<add key="ebs:debugLocalization" value="0" />
```

To:

```
<add key="ebs:debugLocalization" value="1" />
```

## Configure Notifications for Operations

In the FintechOS HPFI it is possible to receive notifications for operations such as scheduled jobs. This is done when options such as **Send Notification On Error** or **Send Notification On Success** are available in Innovation Studio. For this to properly function, the email settings need to be configured in the **mail.config** file, as follows:

- Navigate to the **JobServer** folder inside the FintechOS HPFI installation folder (for example: `C:\FintechOS\MyProject\JobServer`). Please note that the **JobServer** folder is in the folder with the name of your project.

- Open the **mail.config** file with a text editor (such as Notepad++). It needs to contain the following:

```
<mailSettings>
  <server>test@fintechos.com</server>
  <port>587</port>
  <auth>true</auth>
  <user>test@fintechos.com</user>
  <password>insertPasswordHere</password>
  <from>test@fintechos.com</from>
  <to>test@fintechos.com</to>
  <cc>test@fintechos.com</cc>
  <bcc></bcc>
  <replyTo>test@fintechos.com</replyTo>
</mailSettings>
```

- Fill in the parameters, as follows:
  - `<server>FintechServer</server>` - the server from which FintechOS operates
  - `<port>587</port>` - the port corresponding to the server
  - `<auth>true</auth>` - set the value to **true** if the authentication requires username and password
  - `<user>test@fintechos.com</user>` - the username corresponding to the server
  - `<password>insertPasswordHere</password>` - the password of the provided user
  - `<from>test@fintechos.com</from>` - the email address which appears in the **From** field
  - `<to>test@fintechos.com</to>` - the email address to send the notification to
  - `<cc>test@fintechos.com</cc>` - optional; an additional address to send the notification to
- Save and close the **mail.config** file.

# Observability

Observability allows the HPFI platform to generate log events and push those logs to various destinations such as system consoles, files, log servers, or Application Performance Management (APM) services.

## Send log messages to the system console

Configuration in Vault:

| Key Path                                    | Key Name        | Key Value                     |
|---|-----------------|-------------------------------|
| kv/<environment>/<application>/app-settings | console-logging | enabled=0;<br>logLevel=Debug; |

| Parameter | Description   |
|-----------|---|
| enabled   | <ul style="list-style-type: none"><li>• 0 - disable console logging</li><li>• 1 - enable console logging</li></ul>      |
| logLevel  | Minimum severity level for the logged messages. Available options are: Verbose, Debug, Info, Warning, Error, and Fatal. |

## (Deprecated) Configuration using web.config keys:

```
<app-settings>
...
<add key="console-logging" value="enabled=0;
logLevel=Debug;" />
...
</app-settings>
```

## Send log messages to local file storage

Configuration in Vault:

| Key Path                                    | Key Name     | Key Value  |
|---|--------------|--|
| kv/<environment>/<application>/app-settings | file-logging | enabled=1; logLevel=Debug; flushInterval=1s; fileName=trace_roll_.log; retainedFileCount=99; rollSizeBytes=31457280; |

| Parameter         | Description  |
|-------------------|--|
| enabled           | <ul style="list-style-type: none"> <li>0 - disable file logging</li> <li>1 - enable file logging</li> </ul>  |
| logLevel          | Configures the minimum severity level for the logged messages. Available options are: Verbose, Debug, Info, Warning, Error, and Fatal.   |
| flushInterval     | If provided, a full disk flush will be performed periodically at the specified interval in seconds. E.g: 1s, 5s, 20s   |
| fileName          | Name of the log file. The file will be stored in the folder where the application starts.  |
| retainedFileCount | Maximum number of log files that will be retained, including the current log file. For unlimited retention, set it to null. The default value is 31.   |
| rollSizeBytes     | If defined, a new log file is created when the log file size reaches the designated number of bytes. New log files will have an index counter appended in the format _NNN, with the initial log file given no index counter. |

## (Deprecated) Configuration using web.config keys:

```
<app-settings>
...
  <add key="file-logging" value="enabled=1;
logLevel=Debug; flushInterval=1s; fileName=trace_roll_.log;
retainedFileCount=99; rollSizeBytes=31457280;" />
...
</app-settings>
```

## Send log messages to a Seq structured log server

Configuration in Vault:

| Key Path                                    | Key Name    | Key Value   |
|---|-------------|---|
| kv/<environment>/<application>/app-settings | seq-logging | enabled=0; apiKey={seq instrumentation key}; logLevel=Debug; flushInterval=1s; serverUrl=http://localhost:5341; maxEventCount=100000; |

| Parameter     | Description   |
|---------------|---|
| enabled       | <ul style="list-style-type: none"> <li>0 - disable Seq logging</li> <li>1 - enable Seq logging</li> </ul>   |
| apiKey        | Seq API key that authenticates the client to the Seq server.  |
| logLevel      | Minimum severity level for the logged messages. Available options are: Verbose, Debug, Info, Warning, Error, and Fatal.                             |
| flushInterval | Time to wait (in seconds) between checking for event batches.   |
| serverUrl     | Base URL of the Seq server that log events will be sent to.   |
| maxEventCount | Maximum number of events that will be held in-memory while waiting to ship them to Seq. Beyond this limit, events will be dropped. Default: 100000. |

## (Deprecated) Configuration using Web.config keys:

```
<app-settings>
...
  <add key="seq-logging" value="enabled=0; apiKey={seq
instrumentation key}; logLevel=Debug; flushInterval=1s;
serverUrl=http://localhost:5341; maxEventCount=100000;" />
...
</app-settings>
```

## Send log messages to an Azure Application Insights service

Configuration in Vault:

| Key Path                                    | Key Name                  | Key Value   |
|---|---------------------------|---|
| kv/<environment>/<application>/app-settings | azure-appinsights-logging | enabled=0; apiKey={app insights instrumentation key}; logLevel=Verbose; flushInterval=1m; |

| Parameter     | Description   |
|---------------|---|
| enabled       | <ul style="list-style-type: none"> <li>0 - disable Azure Application Insights logging</li> <li>1 - enable Azure Application Insights logging</li> </ul>                                   |
| apiKey        | Instrumentation key value that it is used by default by all Microsoft.ApplicationInsights.TelemetryClient instances created in the logger.  |
| logLevel      | Minimum severity level for the logged messages. Available options are: Verbose, Debug, Info, Warning, Error, and Fatal.   |
| flushInterval | Maximum telemetry batching interval. Once the interval expires, Microsoft.ApplicationInsights.WindowsServer.TelemetryChannel serializes the accumulated telemetry items for transmission. |

### (Deprecated) Configuration using web.config keys:

```
<app-settings>
...
  <add key="azure-appinsights-logging" value="enabled=0;
  apiKey={app insights instrumentation key}; logLevel=Verbose;
  flushInterval=1m;" />
...
</app-settings>
```

## Configure Azure Application Insights telemetry

Configuration in Vault:

| Key Path                                    | Key Name                            | Key Value   |
|---|-------------------------------------|---|
| kv/<environment>/<application>/app-settings | azure-appinsights-telemetry-logging | <ul style="list-style-type: none"> <li>• 0 - disable telemetry</li> <li>• 1 - enable telemetry</li> </ul> |

## (Deprecated) Configuration using web.config keys:

```

<app-settings>
  ...
  <add key="azure-appinsights-telemetry-logging" value="0"
/>
  ...
</app-settings>

```

## Logging context

In addition to the actual log event, developers can track the context in which a log event occurred (machine name, portal profile, user context regional settings, thread ID, etc.), as well as define their own custom log context properties via server-side scripting. For more information, see the [Innovation Studio User Guide](#).

## Prevent Sequencer Infinite Loops

The Innovation Studio [sequencers](#) allow you to define complex sequences of codes for uses such as invoice numbers. Among the customizations you can apply to your sequencers, is the Filter JS script which allows you to skip specific sequence numbers. Depending on your Filter JS code, this may lead to situations where the sequencer enters an infinite loop. To prevent this, you can limit the sequence numbers your sequencers can skip until an infinite loop error is thrown. To do so, in the ["Configuration Manager" on page 86](#), set up the following key:

| Key Path  | Key Name                    | Description   |
|---|-----------------------------|---|
| kv/<environment>/<Portal Instance>/app-settings | SequencerFilterInfiniteLoop | Numeric value indicating the maximum number of sequence numbers that can be skipped until an infinite loop error is thrown. |

## (Deprecated) Configuration using web.config keys

```
<app-settings>
...
<add key="SequencerFilterInfiniteLoop" value="20"/>
...
</app-settings>
```

# Integrations

This section explains how to integrate the HPFI platform with third party services, such as payment processors, electronic signature providers, or digital profile reviews:

---

## FintechOS Service Pipes

Service Pipes are the integration layer of the FintechOS HPFI. It uses [Apache Camel](#) as routing and mediation engine to integrate the HPFI with external systems. Apache Camel is an integration framework that allows easy implementation of routing and mediation logic using a variety of domain-specific languages (DSLs) .

The FintechOS Service Pipes can be installed either as an Azure web service for cloud deployments or as a Docker container for on-premise deployments.

## App Service Configuration

To configure the Service Pipes on your Azure environment, log in to the **Azure Portal** and navigate to your Service Pipes app service blade. In the Configuration section, set up the following settings:

| Setting                             | Description  |
|-------------------------------------|--|
| app.loglevel.application (optional) | Minimum severity level for the logged messages for the Service Pipes app.<br>Available values are: DEBUG, ERROR, FATAL, INFO, OFF, TRACE, WARN.<br>Default: INFO |

| Setting                                     | Description   |
|---|---|
| app.loglevel.root<br>(optional)             | Minimum severity level for the logged messages for all the packages. For the Service Pipes app specifically, the app.loglevel.application setting will take precedence over app.loglevel.root.<br>Available values are: DEBUG, ERROR, FATAL, INFO, OFF, TRACE, WARN.<br>Default: INFO |
| app.loglevel.security<br>(optional)         | Minimum severity level for security messages (authentication or authorization).<br>Available values are: DEBUG, ERROR, FATAL, INFO, OFF, TRACE, WARN.<br>Default: INFO  |
| spring.profiles.active<br>(optional)        | Leave empty to use the <a href="#">"FintechOS Identity Provider" on page 157</a> for authentication. This is the default behavior and it is the recommended setting. Set it to <i>basic</i> if you wish to use the legacy platform credentials for authentication.                    |
| app.context.path<br>(optional)              | The servlet path used for the Service Pipes app service, which is going to be appended to the app service URL. By default, the app will be available at the <i>/services</i> servlet path, for example:<br><code>https://app-myApp.azurewebsites.net/services/</code>                 |
| app.vault.url                               | URL of the <a href="#">"Configuration Manager" on page 86</a> app service or Windows service.   |
| app.vault.token                             | Access token for the <a href="#">"Configuration Manager" on page 86</a> .   |
| app.vault.secrets.engine<br>(optional)      | Secrets engine used by the <a href="#">"Configuration Manager" on page 86</a> .<br>Default: kv  |
| app.vault.required.secrets.path             | Path to services pipes properties in the <a href="#">"Configuration Manager" on page 86</a> (i.e. kv/{environment name}/service-pipes). The properties from this secrets path are required for the application startup and runtime.   |
| app.vault.custom.secrets.path<br>(optional) | Path to additional properties in the <a href="#">"Configuration Manager" on page 86</a> that are loaded in the service pipes cache.   |

## Configuration Manager Settings

The settings used to associate the Service Pipes with a FintechOS Portal instance are stored in the "[Configuration Manager](#)" on [page 86](#). Currently, a single Service Pipes instance can be associated with a single FintechOS Portal instance. The corresponding secrets are stored in the Configuration Manager at the `kv/{environment name}/service-pipes` path.

## Configuration for Environments Using the FintechOS Identity Provider

| Key           | Value  | Description  |
|---------------|--|--|
| openid-config | <pre>{   "realm": "fintechOSRealm",   "auth-server-url":     "https://myServer.azurewebsites.net/auth",   "ssl-required": "external",   "resource": "admin-servicepipes-dev",   "principal-attribute": "preferred_username",   "credentials": {     "secret":       "TkATkOrKeTuubnZqo7ecEKGAq6UXEvW"   },   "use-resource-role-mappings": "false",   "autodetect-bearer-only": "true" }</pre> | <ul style="list-style-type: none"><li>• realm - The FintechOS realm configured in the "FintechOS Identity Provider" on page 157.</li><li>• auth-server-url - The "FintechOS Identity Provider" on page 157</li></ul> |

| Key | Value | Description   |
|-----|-------|---|
|     |       | <div>discovery endpoint.</div> <div><div><div>• ssl-require -</div><div>The default value is <i>external</i> meaning that HTTPS is required by default for external requests. In production</div></div></div> |

| Key | Value | Description  |
|-----|-------|--|
|     |       | <p>enviro<br/>nment<br/>s, this<br/>should<br/>be set<br/>to <i>all</i>.</p> <ul style="list-style-type: none"><li>• resour<br/>ce -<br/>Name<br/>of the<br/>Servic<br/>e<br/>Pipes<br/>resour<br/>ce as<br/>define<br/>d in<br/>the<br/>Fintec<br/>hOS<br/>realm<br/>config<br/>ured<br/>in the<br/>"Finte<br/>chOS<br/>Identit<br/>y</li></ul> |

| Key | Value | Description   |
|-----|-------|---|
|     |       | <p>Provider" on page 157.</p> <ul style="list-style-type: none"><li>• principal-attribute - OpenID Connect ID Token attribute used to populate the UserPrincipal name. Possible values are:</li></ul> |

| Key | Value | Description  |
|-----|-------|--|
|     |       | <p><i>sub,</i></p> <p><i>prefer</i></p> <p><i>red_</i></p> <p><i>userna</i></p> <p><i>me,</i></p> <p><i>email,</i></p> <p><i>name,</i></p> <p><i>nickna</i></p> <p><i>me,</i></p> <p><i>given_</i></p> <p><i>name,</i></p> <p>and</p> <p><i>famil</i></p> <p><i>y_</i></p> <p><i>name.</i></p> <p>Default</p> <p>t:</p> <p><i>prefer</i></p> <p><i>red_</i></p> <p><i>userna</i></p> <p><i>me.</i></p> <ul style="list-style-type: none"><li>• secret</li><li>-</li></ul> <p>Secret</p> <p>key</p> <p>set up</p> <p>in the</p> <p>"Finte</p> |

| Key | Value | Description   |
|-----|-------|---|
|     |       | <p>chOS</p> <p>Identit</p> <p>y</p> <p>Provid</p> <p>er" on</p> <p>page 1</p> <p>57 for</p> <p>the</p> <p>Servic</p> <p>e</p> <p>Pipes.</p> <ul style="list-style-type: none"><li>• use-<br/>resou<br/>rce-<br/>role-<br/>mapp<br/>ings -<br/>When<br/><i>true</i>,<br/>the<br/>adapt<br/>er<br/>retrie<br/>ves<br/>the<br/>user's<br/>applic<br/>ation<br/>level<br/>role<br/>mapp<br/>ings<br/>from</li></ul> |

| Key | Value | Description  |
|-----|-------|--|
|     |       | <p>the token. When <i>false</i>, it looks at the realm level role mappings. This should be set up in accordance with your OpenID configuration. Default: <i>false</i>.</p> <ul style="list-style-type: none"><li>• autodetect-bearear-only - Set it to <i>true</i>. Do</li></ul> |

| Key | Value | Description        |
|-----|-------|--------------------|
|     |       | not<br>chang<br>e. |

| Key                 | Value   | Description   |
|---------------------|---|---|
| openid-token-config | <pre>{   "client-id": "admin-portal-dev",   "client-secret":     "hWpSm6OPPzqUTBLW0Zk9013AEXanBTWe",   "token-url":     "https://myServer.azurewebsites.net/auth/realms/fintechOSRealm/protocol/openid-connect/token" }</pre> | <ul style="list-style-type: none"><li>• client-id - The Client ID configured in the "FintechOS Identity Provider" on page 157 for the Service Pipes.</li><li>• client-secret - The client secret set up in the "Finte</li></ul> |

| Key | Value | Description  |
|-----|-------|--|
|     |       | <div>chOS</div> <div>Identit</div> <div>y</div> <div>Provid</div> <div>er" on</div> <div>page 1</div> <div>57 for</div> <div>the</div> <div>Servic</div> <div>e</div> <div>Pipes.</div> <div><div>• token-</div><div>url -</div><div>URL</div><div>used</div><div>to</div><div>retriev</div><div>e the</div><div>access</div><div>token</div><div>from</div><div>the</div><div>"Finte</div><div>chOS</div><div>Identit</div><div>y</div><div>Provid</div><div>er" on</div></div> |

| Key           | Value   | Description   |
|---------------|---|---|
|               |   | <a href="#">page 1</a><br><a href="#">57.</a>   |
| path-to-roles | <code>/api/**=service-pipes-user;/specifications/**=service-pipes-user</code> | User roles used to provide access to the <i>api</i> and <i>specifications</i> paths in your Service Pipes projects. The <i>service-pipes-user</i> role is available by default, but additional roles can be set up. |
| portal-url    | <code>https://myServer.azurewebsites.net/myPortal</code>                      | URL of the FintechOS Portal instance associated with the Service Pipes.   |

## Configuration for Environments Using Legacy Authentication (non-FintechOS Identity Provider)

| Key        | Value  |  |
|------------|--|--|
| username   | {username}   | User name used by the associated Portal instance to authenticate when calling the Service Pipes. A matching key-value pair must be set up in the Configuration Manager on the associated Portal environment at <code>kv/&lt;environment&gt;/&lt;portalName&gt;/app-settings</code> . |
| password   | {password}   | Password used by the associated Portal instance to authenticate when calling the Service Pipes. A matching key-value pair must be set up in the Configuration Manager on the associated Portal environment at <code>kv/&lt;environment&gt;/&lt;portalName&gt;/app-settings</code> .  |
| portal-url | <code>https://myServer.azurewebsites.net/myPortal</code> | URL of the FintechOS Portal instance associated with the Service Pipes.  |

| Key              | Value   |   |
|------------------|---|---|
| ebs-token-config | <pre>{<br/>  "client-id": "id",<br/>  "username": "test",<br/>  "password": "secret",<br/>  "token-url":<br/>    "https://myServer.azurewebsites.net/<br/>    /myPortal/api/Authorize/GetToken"<br/>}</pre> | <ul style="list-style-type: none"><li>• client-id - For future developments. Use any value.</li><li>• username - User name used by the Service Pipes to authenticate when calling the associated Portal instance. A matching user account must be set up on the associated Portal instance.</li></ul> <div><b>HINT</b><br/>You can use this user account to</div> |

| Key | Value |  |
|-----|-------|--|
|     |       | <div>restric<br/>t<br/>acces<br/>s<br/>only<br/>to<br/>the<br/>endp<br/>oints<br/>that<br/>are<br/>going<br/>to be<br/>calle<br/>d via<br/>Servi<br/>ce<br/>Pipe<br/>s.</div> <ul style="list-style-type: none"><li>password - Password used by the Service Pipes to authenticate when calling the associated Portal instance. A matching user</li></ul> |

| Key | Value |   |
|-----|-------|---|
|     |       | <p>account password must be set up on the associated Portal instance.</p> <ul style="list-style-type: none"> <li>• token-url - API endpoint used to retrieve the access token for the FintechOS Portal instance.</li> </ul> |

## User Roles

When using the ["FintechOS Identity Provider" on page 157](#) for identity and access management, make sure you assign the following user roles accordingly:

- service-pipes-admin - Users responsible for monitoring the Service Pipes. This will provide them access to the Service Pipes monitoring tool available at the `<Service Pipes URL>/actuator/hawtio` path. E.g.:  
`https://myServer.azurewebsites.net/services/actuator/hawtio`
- service-pipes-user - Users that need to authenticate to the Service Pipes, such as user accounts that will run digital journeys that make calls to the Service Pipes server.

For advanced configurations, roles can be further customized using the ["Configuration Manager" on page 86](#). *service-pipes-admin* is the default role and is preset inside the app, giving access to every URL or page exposed by the service pipes.

## Connect to Azure Notification Hubs

The server SDK [sendMobileNotifications](#) function allows you to send notifications to subscribed user devices via the Azure Notifications Hub push engine. To connect HPFI with the Azure Notifications Hub, follow the steps below:

1. Configure your notifications hub on the Microsoft Azure cloud computing service. For details, see the [Azure Notification Hubs documentation](#).

### NOTE

You have to create one notification hub per mobile app, per environment.

2. In Vault, add secrets for the hub name and endpoint settings of each client application, based on the model below:

| Key Path                                    | Key Name                                   |
|---|--|
| kv/<environment>/<application>/app-settings | azure-mobile-notifications-myApp1-hubname  |
| kv/<environment>/<application>/app-settings | azure-mobile-notifications-myApp1-endpoint |
| kv/<environment>/<application>/app-settings | azure-mobile-notifications-myApp2-hubname  |
| kv/<environment>/<application>/app-settings | azure-mobile-notifications-myApp2-endpoint |

| Key Name   | Key Value   |
|--|---|
| azure-mobil<br>e-notific<br>ation<br>s-myAp<br>p1-hubna<br>me  | xxxhubname1   |
| azure-mobil<br>e-notific<br>ation<br>s-myAp<br>p1-endpo<br>int | Endpoint=sb://xxxnamespace1.servicebus.windows.net/;SharedAccessKey<br>yName=DefaultFullSharedAccessSignature1;SharedAccessKey=xxxxxxxxx1 |

| Key Name   | Key Value  |
|--|--|
| azure-mobil<br>e-notific<br>ation<br>s-myAp<br>p2-hubna<br>me  | xxxhubname2  |
| azure-mobil<br>e-notific<br>ation<br>s-myAp<br>p2-endpo<br>int | Endpoint=sb://xxxnamespace2.servicebus.windows.net/;SharedAccessKeyName=DefaultFullSharedAccessSignature2;SharedAccessKey=xxxxxxxxx2 |

In the example above:

- We set up two client applications that will receive notifications: **myApp1** and **myApp2**.
- The notifications are sent using the **xxxhubname1** and **xxxhubname2** Azure notifications hubs respectively.

- The endpoints for the two hubs are **sb://xxxnamespace1.servicebus.windows.net/** and **sb://xxxnamespace2.servicebus.windows.net/**.
- The shared access key names are **DefaultFullSharedAccessSignature1** and **DefaultFullSharedAccessSignature2**.
- The shared access keys are **xxxxxxxxx1** and **xxxxxxxxx2**.

## (Deprecated) Add configuration in web.config:

```
<app-settings>
  ...
  <add key="azure-mobile-notifications-myApp1-
hubname" value="xxxhubname1">
  <add key="azure-mobile-notifications-myApp1-
endpoint" value=
"Endpoint=sb://xxxnamespace1.servicebus.windows.net/;SharedA
ccessKeyName=DefaultFullSharedAccessSignature1;SharedAccessK
ey=xxxxxxxxx1">
  ...
  <add key="azure-mobile-notifications-myApp2-
hubname" value="xxxhubname2">
  <add key="azure-mobile-notifications-myApp2-
endpoint" value=
"Endpoint=sb://xxxnamespace2.servicebus.windows.net/;SharedA
ccessKeyName=DefaultFullSharedAccessSignature2;SharedAccessK
ey=xxxxxxxxx2">
</app-settings>
```

## Push Notifications Log

Sent notifications are saved in the FTOS\_DPA\_MessageQueue table. The table contains an entry for each notification sent to each user. This message queue can be viewed at the [http://localhost:57123/Main#/entity/FTOS\\_DPA\\_MessageQueue/list](http://localhost:57123/Main#/entity/FTOS_DPA_MessageQueue/list) link:

| Attribute | Description                                    |
|-----------|--|
| ToAddress | Mobile app name.                               |
| UserId    | ID of the user that received the notification. |

| Attribute             | Description  |
|-----------------------|--|
| Subject               | Message queue subject set in the <a href="#">sendMobileNotifications</a> function call that initiated the notification push. |
| Body                  | Message received by the recipient.   |
| ChannelProvider       | Provider with the same name as the Mobile App Name.  |
| CommunicationChannel  | Hardcoded to <b>AzureNotificationHub</b> .   |
| ChannelProviderParams | Recipient filter used when sending the notification (example: "role: developer").  |
| MessageStatus         | Hardcoded to <b>Sent</b> . If notifications were not successfully received, check the logging in the Azure Portal.           |

## FAQs

### What happens if the messages are added to the FintechOS message queue, but are not sent by the Azure notification hub?

The messages remain in the Sent status. The main purpose of the message queue logging is to track notification attempts. For troubleshooting, the main place to check statuses is the Azure portal.

### Is there a limit to the number of notifications sent at once?

No. The Azure Notification Hubs support a maximum of 20 tags on a notification command, but HPFI automatically splits the notification into batches if this number is exceeded.

### If my role contains 5 users, but only 3 are registered to the notification hub, how many push notifications are going to be sent?

In the HPFI message queue, there are going to be 5 entries (one for each user). On the Azure notification hub, after sending the request from HPFI, the outcome will be: 3 Success, 0 Failed.

### Can I have only one notification hub for 2 client apps?

It is recommended to have 1 hub per app. In the Azure portal > Notification Hub setup, you can set only one key, which is normally associated to only one app.

There are exceptions such as Android-FCM where the API key corresponds to the Firebase project. In this case, you can:

- Add multiple apps on the same Firebase project, resulting in a single API key.
- Create a project for each App, resulting in separate API keys/hubs.

## Collect Logging Data in Azure Application Insights

You can use the Azure Application Insights application performance management service to collect logging information independently from the *trace.log* local file. If you want to centralize your logs, you can configure multiple machines on the same cluster to send their logging information to the same Application Insights subscription.

### NOTE

Logs saved to the local *trace.log* file are visible instantly. Messages sent to Azure Application Insights might be visible after a short delay ranging from seconds to minutes.

## Prerequisites

An Azure Application Insights subscription.

## Configuration

### Azure Application Insights Logging Vault Configuration

| Key Path                                    | Key Name                  | Key Value  |
|---|---------------------------|--|
| kv/<environment>/<application>/app-settings | azure-appinsights-logging | enabled=1; apiKey=API_KEY; logLevel=Warning; flushInterval=1m; |

| Values        | Data Type   | Default Value | Description  |
|---------------|-------------|---------------|--|
| enabled       | boolean     | 0             | Enables or disables Azure Application Insights logging: true/false, on/off, 0/1.   |
| apiKey        | guid string | empty         | Instrumentation Key from your Azure Application Insights subscription.   |
| logLevel      | string      | Error         | <p>Minimum logging level. Possible values: Fatal, Error, Warning, Information, Debug, Verbose.</p> <div> <p><b>IMPORTANT!</b></p> <p>Specifying a log level greater than Warning, may flood Azure Application Insights with irrelevant information. It is not recommended to enable Azure Application Insights logging on development environments.</p> </div> |
| flushInterval | timespan    | 1m            | Delay between message batches sent to Azure, to be specified in .NET Timespan format or in Jira timespan format: 1h, 1d, 1h30m, 00:00:05, etc.   |

(Deprecated) Configuration using **web.config** files:

To configure Application Insights logging, use the `feature-logging-azure-appinsights` key in the `<app-settings>` node in the `web.config` file:

```
<app-settings>
...
  <add key="feature-logging-azure-
appinsights" value="enabled=1; apiKey=API_KEY;
logLevel=Warning; flushInterval=1m">
...
</app-settings>
```

## Azure Application Insights Telemetry Configuration

Telemetry is enabled by default. To disable Azure Application Insights telemetry, use the below secret:

| Key Path                                    | Key Name                            | Key Value |
|---|-------------------------------------|-----------|
| kv/<environment>/<application>/app-settings | azure-appinsights-telemetry-logging | 0;        |

### NOTE

Deactivating telemetry does not affect logging.

## (Deprecated) Configuration using web.config files:

Use the `feature-logging-azure-appinsights-telemetry-disabled` key in the `<app-settings>` node in the `web.config` file:

```
<app-settings>
...
  <add key="feature-logging-azure-appinsights-telemetry-
disabled" value="1"/>
...
</app-settings>
```

## Collected Data

The following application specific information is saved for each log entry as part of custom dimensions:

- MachineName
- CorrelationId
- Language
- AutomationScript
- AutomationScriptLibrary
- SQL
- ExtraErrorDetails

## Severity Levels

Log entries with severity levels Warning, Information, Debug, or Verbose are saved as **traces**.

Log entries with severity levels Exception or Fatal are saved as **exceptions**.

### Information Severity Level Examples

- All calls to the [log](#) Server SDK function executed from automation scripts.
- Success/Error status for http get/post calls to external APIs executed from automation scripts.
- Authentication errors and information for AD and EBS providers.

### Warning Severity Level Example

Missing or mismatched configuration.

**Error Severity Level Examples**

- Runtime exceptions with full stack trace.
- Errors in automation scripts. These errors include the automation script name, line, and column from the JavaScript code.
- Application custom errors.
- Authentication errors and information for OpenID providers.

## CertSign Integration for electronic signature

Certsign is a digital certification for digital signatures. It will provide the user with the capability to use the ESign processor in the Studio and Portal. This makes possible to sign contracts and other documents by a customer. The existing integration provides two types of signature:

- Remote signature (with authorization code sent through sms)
- Automatic signature (with an existing certificate)
- Automatic signature with qualified electronic sign.

After the installation of the ESign provider package, you should add the following configuration in Vault, or in JobServer serviceSettings.config:

In Vault

| Key Path  | Key Name                           |
|---|------------------------------------|
| kv/<environment>/<FintechOS Portal instance>/app-settings | FTOSServicesESignProvider2Endpoint |
| kv/<environment>/<FintechOS Portal instance>/app-settings | FTOSServicesESignProvider2AppId    |
| kv/<environment>/<FintechOS Portal instance>/app-settings | ESignProvider2CertName             |

- For the FTOSServicesESignProvider2Endpoint key, add as value the URL to the environment.
- For the FTOSServicesESignProvider2AppId key, add as value the subscription key.
- For the ESignProvider2CertName key, add as value the mapping for the certificate provided by FTOS.

In JobServer serviceSettings.config:

```
<add
key
=
"FTOSServicesESignProvider2Endpoint"
  value="https://aztestapi01.azure-api.net/certSign"/> <!-- This is
the test env url -->
<add key="FTOSServicesESignProvider2AppId" value=""/><!-- the
subscription key -->
<add key="ESignProvider2CertName" value="certSignTest"/> <!-- the
mapping for the certificate provided by FTOS-->
```

If you have to configure the automatic signature also, add the following secrets in Vault:

| Key Path  | Key Name                            |
|---|-------------------------------------|
| kv/<environment>/<FintechOS Portal instance>/app-settings | ESign2AutomaticNumber_{ProfileName} |
| kv/<environment>/<FintechOS Portal instance>/app-settings | ESign2AutomaticName_{ProfileName}   |

- For ESign2AutomaticNumber\_{ProfileName}, add as value the serial number provided for the specific profile
- For ESign2AutomaticName\_{ProfileName}, add as value the issuer information for the profile

## (Deprecated) Add keys in web.config

```
<add key="ESign2AutomaticNumber_
{ProfileName}" value=""/> <!--this will contain the
serial number provided for the specific profile-->
<add key="ESign2AutomaticName_
{ProfileName}" value="cn=certSIGN CA Class 2
G2,ou=certSIGN CA Class 2 G2,o=certSIGN,c=RO"/> <!--
this will contain the issuer information for the
profile-->
```

**IMPORTANT!**

The token {ProfileName} must be replaced with a profile name that will be used when requesting the signature process.

## Set up for the automatic signature with qualified electronic sign

After the installation of the ESign provider package, you should add the following configuration in JobServer serviceSettings.config:

```
<add
key
="FTOSServicesESignProvider2Endpoint"
value="https://aztestapi01.azure-api.net/certSign"/> <!-- This is
the test env url -->
  <add key="FTOSServicesESignProvider2AppId" value=""/><!-- the
subscription key -->
  <add
key
="ESignProvider2AutomaticQESCertName"
value="certSignTestAutomatic"/> <!-- the mapping for the
certificate provided by FTOS for the Automatic QES signature-->
```

Insert a record in the business entity **FTOS\_DDM\_ESignQueue**, this record will contain the configurations that will be used for automatic qualified electronic sign.

ProfileName, choose a name for this automatic profile, make sure it is unique if you have multiple configurations:

- ExternalId, this value should be provided CertSign, it will be the externalId of the user that is enrolled to sign with automatic QES
- Seed, this value will be read by the agent from his CertSign account (he will receive an email with steps to follow)
- WorkstepsBulkNo, this represents the number of worksteps that will be sent in the request to be signed with automatic QES.

## Calling the automatic signature with qualified electronic sign

The request for automatic QES will be sent together with the rest of the worksteps. The signing processes will be made in the order provided in the request. It is recommended that this signature to stay at the end because it will be processed async by Job Server.

The workstep with automatic QES should be like:

```
{
  "signatureTag": "#tagAgentQES#",
  "signatureType": FTOSServices.DDM.signatureTypeAutomaticQES,
  "automaticProfile": "ProfileNameDefinedInQueue", //defined in
  FTOS_DDM_ESignQueue
  "signatureStamp": { //this is for the signature stamp
    "SignerName": "Sign name",
    "Reason": "Credit loan", //this will appear in signature
    details
    "Subject": "Bank signature",
    "ShowTimeStamp": true, //show date in signature stamp
    "FontSize": "12"
  }
}
```

## Example

```

var signRequest = {
  "workstepConfigs": [
    {
      "signatureTag": "#tagClient#",
      "signatureType":
      :FTOSServices.DDM.signatureType.QualifiedElectronicSign,
      "recipient": {
        "Country": "RO",
        "Email": "@fintechos.com",
        "ExternalId": "", //an unique id representing the
customer (ex: Accoountid)
        "FirstName": "M",
        "LastName": "C",
        "PhoneMobile": "+407",
        "SocialSecurityNumber": "", //PIN
        "IdPhoto": "", // ftos file attribute value
representing the id picture
      },
      "signatureStamp":{
        "Reason": "Client reason",
        "Subject": "Credit loan",
        "SignerName": "TestFirstName TestLastName",
        "FontSize": "12"
      }
    },
    {
      "signatureTag": "#tagAgentQES#",
      "signatureType":
      FTOSServices.DDM.signatureTypeAutomaticQES,
      "automaticProfile": "ProfileNameDefinedInQueue",
      "signatureStamp": { //this is for the signature stamp
        "SignerName": "Sign name",
        "Reason": "Credit loan", //this will appear in
signature details
        "Subject": "Bank signature",
        "ShowTimeStamp": true, //show date in signature
stamp
        "FontSize": "12"
      }
    }
  ],
  "signedDocumentName": "test.pdf",
  "files": [

```

```

    {
      "ftosFile": "" //ftos file attribute value
representing the pdf that needs to be signed
    }
  ]
}

```

**NOTE**

The automatic QES signing has to be processed by Job Server, so you must configure a schedule trigger with a server side script. Please make sure that you won't use the same ExternalId on multiple instances of FintechOS. Also, the scheduled trigger should be configured to run with a frequency of at least 30 seconds. The recommended cron expression should be to start at second 0 or at second 30 and to run from 30s in 30s (or a multiple of 30s = 1min, 2min, etc).

In the server side script you have to call the following method with the parameter `ProfileName` that you've defined in `FTOS_DDM_ESignQueue`:

```

//This method returns the list with processed eSignId and the
workstepId that has been finished
//As input you should pass the Name that you've configured in FTOS_
DDM_ESignQueue
var result = FTOSServices.DDM.ESign2.processPendingAutomaticQESSign
("ProfileNameDefinedInQueue");

// if success the result will look like:
/*
{
  "isSuccess": true,
  "isFinished": true,
  "eSignProcesses": [
    {
      "eSignId": "e2f2bc20-de1f-41f1-851b-5c0279cc4cb7",
      "eSignWorkstepId": "13b2ea4e-f5cb-491a-a32a-268741e7da2a"
    },
    {
      "eSignId": "ac42d532-88a3-4db6-932c-9d0888e2fd0e",
      "eSignWorkstepId": "6830d22a-9309-4a11-9e9a-05ab7ac8807b"
    }
  ]
}

```

```

    }
  ]
}
*/

```

## FTOS ESign Services API

In order to sign a document you must call the following methods:

1. RequestSign (for the configuration of the automatic signature)
 

For client signature with remote method with authorization code sent through sms:
2. AcceptTermsAndConditions
3. Authorize signature
4. Resend code

## RequestSign

Firstly, add a reference to the library FTOSServices. To request with qualified electronic signature and automatic, use the following example:

```

var signRequest = {
  "workstepConfigs": [{
    "signatureTag": "#tagClient#",
    "signatureType":
FTOSServices.DDM.signatureType.QualifiedElectronicSign,
    "recipient": {
      "Country": "RO",
      "Email": "@fintechos.com",
      "ExternalId": "", //an unique id representing the
customer (ex: Accountid)
      "FirstName": "M",
      "LastName": "C",
      "PhoneMobile": "+407",
      "SocialSecurityNumber": "", //PIN
    }
  ]
}

```

```

        "IdPhoto": "", // ftos file attribute value
        representing the id picture
    },
    "signatureStamp": {
        "Reason": "Client reason",
        "Subject": "Credit loan",
        "SignerName": "TestFirstName TestLastName"
    }
}, {
    "signatureTag": "#tagBank#",
    "signatureType":
FTOSServices.DDM.signatureTypeAutomaticSign,
    "automaticProfile": "Profile1",
    "signatureStamp": { //this is for the signature stamp
        "Reason": "Credit loan", //this will appear in
signature details
        "Subject": "Bank signature",
        "ShowTimeStamp": true //show date in signature stamp
    }
}],
    "signedDocumentName": "test.pdf",
    "files": [{
        "ftosFile": "" //ftos file attribute value representing the
pdf that needs to be signed
    }]
}

```

Optionally, you can add to the recipient property the following information. It will appear in terms and conditions file.....

```

"workstepConfigs": [{
    "signatureTag": "#tagClient#",
    "signatureType":
FTOSServices.DDM.signatureTypeQualifiedElectronicSign,
    "recipient": {.....
        "DocumentIssuedBy": "splcid", "DocumentIssuedOn":
"2020-01-20", "DocumentExpiryDate": "2050-01-25", "DocumentNumber":
"123456", "DocumentSeries": "xa", "County": "Braila", "City":
"Braila", "Street": "asd", "StreetNo": "12", "Block": "asd",
"Entrance": "q", "ApartmentNo": "123", "ZipCode": "123453",
    }
}

```

**HINT**

If you have a request with multiple signatures, please keep in mind that the signing process is sequentially and the client signature must have manual input (accept terms and conditions and authorize signing using the code received via sms).

You should save the eSignId in order to track the status of the eSign process using the method GetESignStatus.

## AcceptTermsAndConditions

Add a reference to the client script library FTOS\_DDM\_ESignProvider2.

```
/**
 * Accepts terms and conditions for emitting the certificate
 * @param termsId is the id returned by requestSign method
 * (entityId property)
 * @param accepted should be set on true, if the user accepts
 * the terms and conditions
 * @return documentId that will be used for authorize signing
 */
acceptTermsAndConditions(termsId: string, accepted:
boolean): Promise<any>
```

Example:

```
var ddmESign = ebs.importClientScript("FTOS_DDM_
ESignProvider2");
ddmESign.acceptTermsAndConditions(termsId, true).then
(function(result) {
    console.log(result);
    //output should be {isSuccess: true, entityId: "12437-
34873"}
}, function(error) {
    console.log(error);
});
```

## Authorize signature

Add a reference to the client script library FTOS\_DDM\_ESignProvider2.

```
// @param documentId is the id returned by
acceptTermsAndConditions method (entityId property)
// @param code should be the code sent via sms for the
signing process

authorizeSign(documentId: string, code: string): Promise <
any > ;
```

```
// Example:

var ddmESign = ebs.importClientScript("FTOS_DDM_
ESignProvider2");
ddmESign.authorizeSign(documentId, code).then(function
(result) {
    console.log(result);
    //output should be {isSuccess: true}
}, function(error) {
    console.log(error);
});
```

Example:

```
var ddmESign = ebs.importClientScript("FTOS_DDM_
ESignProvider2");
ddmESign.authorizeSign(documentId, code).then(function
(result) {
    console.log(result);
    //output should be {isSuccess: true}
}, function(error) {
    console.log(error);
});
```

## Resend sms code

As before, add a reference to the client script library FTOS\_DDM\_ESignProvider2.

```
/**
 * @param documentId is the id returned by
 * acceptTermsAndConditions method (entityId property)
 */
resendCode(documentId: string): Promise<any>
```

Example:

```
var ddmESign = ebs.importClientScript("FTOS_DDM_
ESignProvider2");
ddmESign.resendCode(ebs.getCurrentEntityId()).then(function
(result) {
    console.log(result);
});
```

```
//output should be {isSuccess: true}
}, function(error) {
  console.log(error);
});
```

Async methods, can be used with JobServer:

## Update statuses

This method should be called using Job Server scheduler. It will update the status of the in progress eSign processes.

```
FTOSServices.DDM.ESign2.updateStatusESignProcess();
```

## ProcessAutomaticSign (sign with automatic signature)

This method should be called using Job Server scheduler. It gets the in progress esign processes that must be signed with automatic signatures.

```
FTOSServices.DDM.ESign2.processPendingAutomaticSign();
```

## Get ESign Status

This method returns the status of the eSign process. If the status is Finished, then you can get the signed document name to use it in your digital journey.

```
/**
 * @param eSignId is the id returned by requestSign
 */
FTOSServices.DDM.ESign2.getESignStatus(eSignId): any
Example: var eSignStatus =
FTOSServices.DDM.ESign2.getESignStatus(eSignId);
log(toJson(eSignStatus));
/*should print:
{
  "isSuccess": true,
  "status": "Finished",
```

```

"documents": "[\r\n {\r\n   \ "Name\ ":
\ "contract1.pdf\ ",\r\n   \ "RealName\ ": \ "contract1_
42dfa5ba-c1f4-4328-b095-
b6075d0c12ee.pdf\ ",\r\n   \ "IsSuccess\ ":
true,\r\n   \ "Message\ ": null,\r\n   \ "ClientScript\ ":
null,\r\n   \ "Serialized\ ": null,\r\n   \ "ErrorCode\ ":
0,\r\n   \ "UIResult\ ": null\r\n } \r\n]"
}
/*

```

Example:

```

var eSignStatus = FTOSServices.DDM.ESign2.getESignStatus
(eSignId);
log(toJson(eSignStatus));
/*should print:
{
  "isSuccess": true,
  "status": "Finished",
  "documents": "[\r\n {\r\n   \ "Name\ ":
\ "contract1.pdf\ ",\r\n   \ "RealName\ ": \ "contract1_
42dfa5ba-c1f4-4328-b095-
b6075d0c12ee.pdf\ ",\r\n   \ "IsSuccess\ ":
true,\r\n   \ "Message\ ": null,\r\n   \ "ClientScript\ ":
null,\r\n   \ "Serialized\ ": null,\r\n   \ "ErrorCode\ ":
0,\r\n   \ "UIResult\ ": null\r\n } \r\n]"
}
/*

```

## Configure the CData Sync Service

The CData Sync service is required for the HPFI Data Pipes data replication feature. CData Sync must be installed on the same machine as the HPFI platform. The service is shared between HPFI instances. If you have multiple platform instances running on the same machine, install the CData Sync service only once.

## System Requirements

- Windows Vista/Windows Server 2008 or higher.
- .NET Framework 4.5 or higher.
- 500 MB RAM required. 1+ GB recommended.
- Adequate free disk space for job logging.

## Installation

1. Copy the CData Sync installation kit provided by HPFI to your local machine.
2. Open Windows PowerShell as administrator and navigate to the installation kit folder.
3. Run the following command in Windows PowerShell:

```
.\FtosCDataSyncInstaller.ps1 -p_MainCommand Install -p_InstallDir <installation path>
```

4. This will start the installer. At the command line prompt, type **GO!** and press **Enter**.

The CData Sync server will be installed in the specified directory. The default credentials are:

- username: admin
- password: admin

For more details about managing the CData Sync server, see the [CData official documentation](#).

## Upgrade

To upgrade the CData Sync server, follow the same ["Installation" on the previous page](#) instructions, but replace the Windows PowerShell command with:

```
.\FtosCDataSyncInstaller.ps1 -p_MainCommand Upgrade -p_InstallDir  
<installation path>
```

## Uninstall

To uninstall the CData Sync server, follow the same ["Installation" on the previous page](#) instructions, but replace the Windows PowerShell command with:

```
.\FtosCDataSyncInstaller.ps1 -p_MainCommand Uninstall -p_InstallDir  
<installation path>
```

# Configure the Payment Processor Service Provider

If you wish to enable online payments in your digital journeys, you need to partner with a payment processor and configure the link to their service in the Innovation Studio *web.config* file.

## 1 Define a new type of section in the web.config file for the payment processor

Open the Innovation Studio *web.config* file in a text editor and add a new entry inside the **<configSections>** node:

```
<section
name
="ftosPaymentProcessor"

type="EBS.Core.Utills.Services.Config.PaymentProcessorConfigSection,
EBS.Core.Utills"/>
```

## 2 Add the connection settings for your payment processor

Open the Innovation Studio *web.config* file in a text editor and add a new entry inside the **<configuration>** node (after **<configSections>**):

```
<ftosPaymentProcessor type="Netopia">
  <definition>
    <settings alias="conf1">
      <setting
name="environment" value="http://sandboxsecure.mobilpay.ro"/>
      <setting name="publicCertificate" value="C:\\PATH_TO_
CERT\\sandbox.cer"/>
      <setting name="privateKey" value="C:\\PATH_TO_
CERT\\sandbox.key"/>
      <setting name="signature" value="XXX"/>
      <setting
name="redirectUrl" value="http://localhost/test_redirect.html"/>
      <setting
name="confirmUrl" value="http://localhost/test_confirm.html"/>
    </settings>
  </definition>
</ftosPaymentProcessor>
```

### NOTE

- Currently, only the *Netopia* payment processor type is supported, which will link to a mobilPay service provided by Netopia Payments. Additional payment processors may be available in the future.

- The *alias* will identify the payment processor service when initiating payments using the `getPaymentToken` function.

## Configure the FTOSApiSMS Service

The FTOSApiSMS service allows the HPFI platform to send SMS messages. Follow the instructions below to enable the service.

### 1 Add a new section in the web.config file for the FTOSApiSMS service

Open the *web.config* file in a text editor and add a new entry in the `<configSections>` node:

```
<configSections>
...
  <section name="ftosApiSmsProvider" type=
    "EBS.Core.Data.Services.CommunicationProviders.Sms.FtosApiSmsProvid
    erConfig, EBS.Core.Data.Services"/>
</configSections>
```

### 2 Add the configuration settings for the FTOSApiSMS service

Add the following configuration element in the *web.config* file:

```
<ftosApiSmsProvider
  xmlns="urn:EBS.Core.Data.Services.CommunicationProviders.Sms"
  serviceUrl="insertyourURL"
```

```

        subscriptionKey="insertyourkey"
        from="SantaClaus"
    />

```

where:

- serviceUrl - the URL to FintechOS SMS gateway;
- subscriptionKey - subscription key for the service;
- from - a text representing the sender of the message.

## Customize the SMS messages sent for Multi-Factor Authentication

- web.config - add a new attribute on multiFactorAuthentication/providers/provider section.  
Name: "messageTemplate". Its value should be one of the FTOS\_CMB\_ActionTemplate records.
- identify the correct template - find FTOS\_CMB\_ActionTemplateContent child of FTOS\_CMB\_ActionTemplate (with name equal to the value of messageTemplate attribute) that has a FTOS\_CMB\_CommunicationChannel (Channel) that has FTOS\_DPA\_ChannelProvider (Bus Communication Provider) with name equal to the name of the channel provider set on the MFA provider. The template also depends on user culture → try to find the template using the user culture (from EbsMetadata.UserSettings with fallback to default system culture);
- if a proper template cannot be identified → error;
- the message (the body of the sms) will be customizable with 2 tokens: {{otp}} (the generated OTP) and {{user\_display\_name}} (DisplayName of the current user);

- if “messageTemplate” is missing the message will contain just the OTP code (as it is now);

## Configure the OneyTrust Digital Review service

The OneyTrust Digital Review service analyzes the information in a user's profile (email, telephone number, address, etc.) and calculates a reliability score for that information. This allows companies to detect potentially problematic profiles and act accordingly (for instance, by deciding to direct them to a manual review process instead of accepting them automatically).

To set up a connection to the OneyTrust service, add the following keys in Vault:

| Key Path                                    | Key Name                      | Key Value            |
|---|-------------------------------|----------------------|
| kv/<environment>/<application>/app-settings | FTOSServicesOneyTrustEndpoint | exposed endpoint url |
| kv/<environment>/<application>/app-settings | FTOSServicesOneyTrustAppId    | the subscription key |

Once the connection to the OneyTrust service is set up, you can use the [createReview](#) and [getReview](#) Server SDK functions to review user profiles.

### (Deprecated) Configuration using **web.config** files:

```
<add key="FTOSServicesOneyTrustEndpoint" value="exposed  
endpoint url"/>  
<add key="FTOSServicesOneyTrustAppId" value="the  
subscription key"/>
```

# Security

HPFI was built on 4 pillars of security: data encryption, authentication, authorization and logging. With a keen focus on security critical aspects, such as: access rights, segregation of duties, data ownership, it also provides you with comprehensive audit trail of what happened at any given time and who performed the action.

For all cloud deployment types, you own your data and identities. You are responsible for protecting the security of your data and identities, on-premises resources, and the cloud components you control (which varies by service type). We recommend you to implement security best practices provided by your cloud provider.

This section covers the following topics:

---

## Data Encryption and Security

One of the keys to data protection is accounting for the possible states in which your data may occur, and what controls are available for that state:

- **Data in transit.** When data is being transferred between components, locations or programs, such as over the network, across a service bus, or during an input/output process, it is thought of as being in-transit.
- **Data at rest.** This includes all information storage objects, containers, and types that exist statically on physical media, be it magnetic or optical disk.

Data in transit is encrypted using the industry standard TLS min. 1.2 encryption algorithm.

Data at rest is encrypted using the AES-256 encryption algorithm.

To establish identity and trust between HPFI web-based platform and the web browser, the connection is secured via SSL certificates.

The SSL-secured communication between HPFI and the client is done using the symmetric encryption keys that are established during the authentication process.

The data model and all scripts defined within HPFI can be exposed through REST APIs to enable integration with 3rd party systems / solutions. HPFI APIs are secured through OAuth 2.0 and follow the OWASP security standards.

You can encrypt the data at rest using security best practices provided by the infrastructure provider of choice where you install and deploy HPFI (Microsoft Azure, AWS, IBM Cloud, other).

## XSS Prevention

To prevent Cross-Site Scripting (XSS) and keep HPFI users safe, all user input data is sanitized by default, except for the following attributes: JavaScript, HTML and XML.

In HPFI, the XSS prevention secures your web apps by escaping user input of type JavaScript, HTML and XM. It censors the data received by the web pages in a way which disallows the following characters: "<", "</", ">", "&lt;" and "&gt;" (e.g., <text, </text, &lt;text or &gt;text) from being rendered.

**IMPORTANT!** When importing deployment packages or adding new metadata in HPFI versions which have XSS prevention enabled, you have to eliminate the following tags from metadata and packages: "<", "</", ">", "&lt;" and "&gt;"; otherwise, you will get an error message and you will not be able to import them.

### XSS prevention when upgrading to HPFI 20.1

When upgrading HPFI to version 20.1, you should enable the request validation to the latest version; otherwise you will be vulnerable to cross-site scripting attacks. To do so, go to the **web.config** file and set the request validation version to **4.5**:

```
<httpRuntime targetFramework="4.6.2" requestValidationMode="4.5"
... />
```

# Authentication

Authentication is the process of verifying the identity of a user based on a set of credentials.

FintechOS HPFI provides the following authentication mechanisms:

|  |            |
|--|------------|
| <b>FintechOS Identity Provider</b>               | <b>157</b> |
| <b>Deprecated Identity Providers</b>             | <b>183</b> |
| <b>Browser Based Multi-Factor Authentication</b> | <b>224</b> |
| <b>Email/SMS/IVR Multi-Factor Authentication</b> | <b>228</b> |
| <b>Deprecated Multi-Factor Authentication</b>    | <b>235</b> |

## FintechOS Identity Provider

**IMPORTANT!**

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services. Alternate authentication methods are provided only for backward compatibility. For more information, see ["Deprecated Identity Providers" on page 183](#).

The FintechOS Identity Provider is an OpenID compliant identity and access management solution based on the [Keycloak](#) authentication server. All HPFI components, such as Innovation Studio, FintechOS Portal, or FintechOS API, are represented in the FintechOS Identity Provider as different clients of the same FintechOS realm.

**NOTE**

User credentials and roles set up in Innovation Studio are stored by the FintechOS Identity Provider. When creating, updating, or deleting a user account in Innovation Studio, the changes are automatically propagated in the FintechOS Identity Provider.

## Identity Brokering

The FintechOS Identity Provider supports identity brokering, allowing users to log in to FintechOS applications and services using any external identity provider that supports the OpenID Connect standard. You can find examples for common external identity providers configurations below:

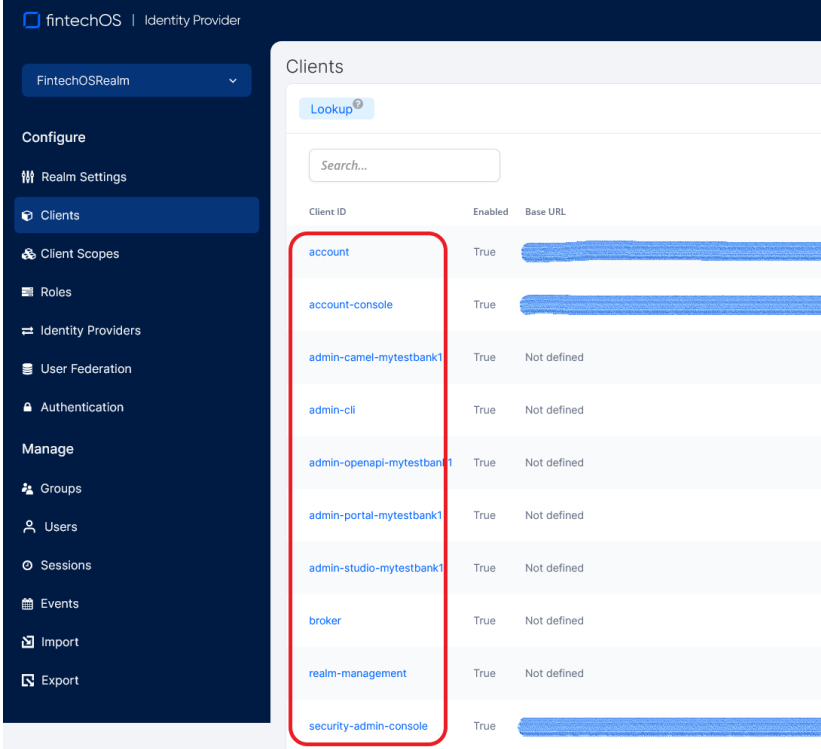
- ["Using Azure AD as External Identity Provider" on page 167](#)
- ["Using Okta as External Identity Provider" on page 172](#)
- ["Using AWS Cognito as External Identity Provider" on page 178](#)

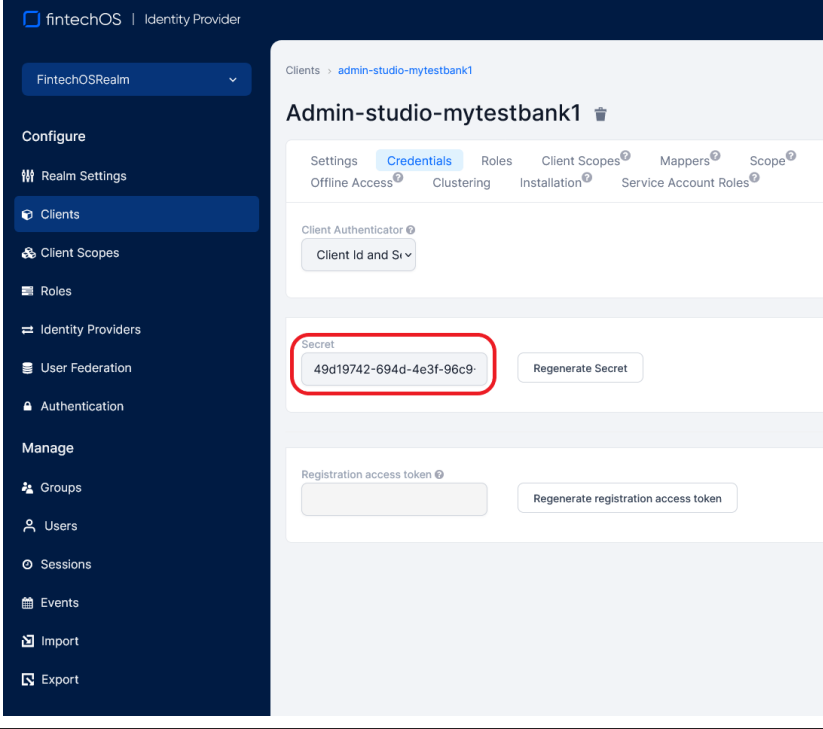
## FintechOS Identity Provider Settings

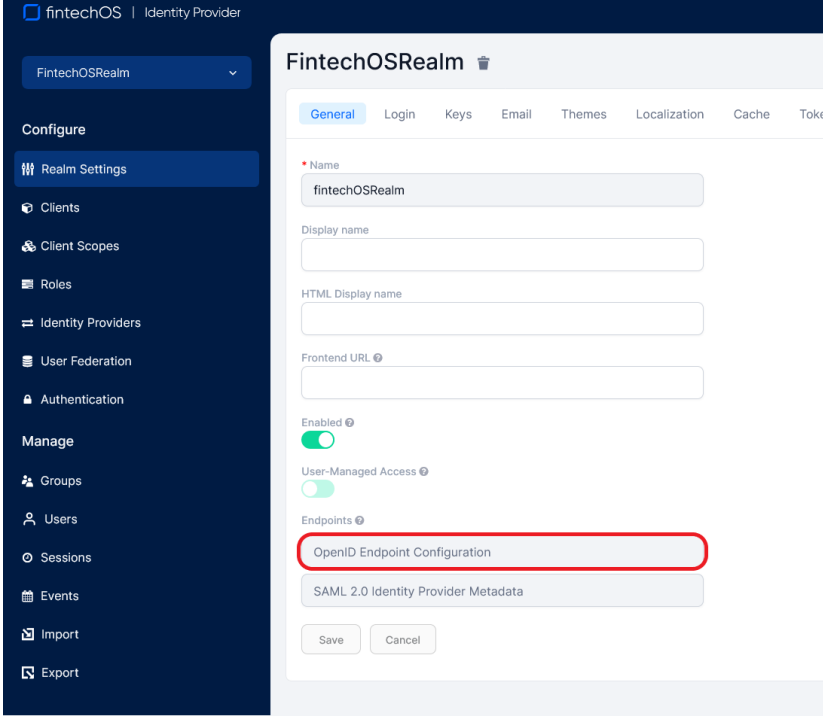
In the FintechOS Cloud Configuration Manager, set up the following secrets:

| Key Path                                    | Key Name                                    |
|---|---|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication                    |
| kv/<environment>/<application>/app-settings | core-setting-external-auth-provider-key-url |
| kv/<environment>/<application>/app-settings | core-setting-external-auth-provider-issuer  |
| kv/<environment>/<application>/app-settings | openid-client-id                            |
| kv/<environment>/<application>/app-settings | openid-client-secret                        |
| kv/<environment>/<application>/app-settings | openid-discovery-endpoint                   |
| kv/<environment>/<application>/app-settings | openid-callback-url                         |

| Key Name                                    | Key Description  |
|---|--|
| EBSDefaultAuthentication                    | <p>Specifies the identity provider:</p> <ul style="list-style-type: none"> <li>• FTOSOIDC - FintechOS Identity Provider</li> <li>• EBS - Legacy FintechOS HPFI authentication (deprecated)</li> </ul> <div style="background-color: #f9c78d; padding: 10px; margin: 10px 0;"> <p><b>IMPORTANT!</b> In a non-standard scenario where a regular portal using FintechOS Identity Provider is linked to a B2C portal using legacy authentication, it is recommended to have each portal reside on a separate domain. The requests will pass successfully only if coming from FTOS IDP towards B2C (the other way around, errors may occur).</p> </div> <ul style="list-style-type: none"> <li>• AD - "<a href="#">Microsoft Active Directory Authentication</a>" on <a href="#">page 183</a> (deprecated).</li> <li>• AzureAD - "<a href="#">Azure Active Directory Authentication</a>" on <a href="#">page 189</a> (deprecated).</li> <li>• Okta - "<a href="#">Authentication with Okta</a>" on <a href="#">page 195</a> (deprecated)</li> <li>• ADFS - "<a href="#">Authentication with Active Directory Federation Services</a>" on <a href="#">page 204</a> (deprecated).</li> <li>• AWSCognito - "<a href="#">Authentication with AWS Cognito</a>" on <a href="#">page 220</a> (deprecated)</li> </ul> |
| core-setting-external-auth-provider-key-url | <p>Link to the FintechOS Identity Provider public keys used to validate the digital signatures of the access tokens.<br/>E.g.:<br/><a href="https://myHPFI.myDomain.com/auth/realms/fintechOSRealm/protocol/openid-connect/certs">https://myHPFI.myDomain.com/auth/realms/fintechOSRealm/protocol/openid-connect/certs</a></p>   |
| core-setting-external-auth-provider-issuer  | <p>FintechOS Identity Provider instance identifier as provided in the issue field of the authentication token. This value is case sensitive.<br/>E.g.: <a href="https://myHPFI.myDomain.com/auth/realms/fintechOSRealm">https://myHPFI.myDomain.com/auth/realms/fintechOSRealm</a></p>   |

| Key Name                  | Key Description  |             |             |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
|---------------------------|--|-------------|-------------|----------|---------|---------|------|------------|-------------|-----------------|------|------------|-------------|-------------------------|------|-------------|-------------|-----------|------|-------------|-------------|---------------------------|------|-------------|-------------|--------------------------|------|-------------|-------------|--------------------------|------|-------------|-------------|--------|------|-------------|-------------|------------------|------|-------------|-------------|------------------------|------|------------|
| openid-client-id          | <p>Your HPFI component's corresponding Client ID as defined in the FintechOS Identity Provider.</p> <p>E.g.: admin-portal, myInnovationStudio.</p> <p>In the FintechOS Identity Provider admin console, you can find the list of Client IDs in the Clients section of your FintechOS realm.</p>  |             |             |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
|                           |  <p>The screenshot shows the FintechOS Identity Provider admin console. On the left is a dark sidebar with a menu including 'Configure', 'Realm Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', 'Authentication', 'Manage', 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The 'Clients' menu item is selected. The main area is titled 'Clients' and contains a 'Lookup' button and a search bar. Below these is a table with columns 'Client ID', 'Enabled', 'Base URL', and 'Actions'. The table lists several clients, with the 'account' client ID highlighted by a red box. The 'Enabled' column for all listed clients is 'True'. The 'Base URL' column contains either a redacted value or 'Not defined'. The 'Actions' column contains 'Edit' and 'Export' links for each client.</p> <table><tr><th>Client ID</th><th>Enabled</th><th>Base URL</th><th>Actions</th></tr><tr><td>account</td><td>True</td><td>[Redacted]</td><td>Edit Export</td></tr><tr><td>account-console</td><td>True</td><td>[Redacted]</td><td>Edit Export</td></tr><tr><td>admin-camel-mytestbank1</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>admin-cli</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>admin-openapi-mytestbank1</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>admin-portal-mytestbank1</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>admin-studio-mytestbank1</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>broker</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>realm-management</td><td>True</td><td>Not defined</td><td>Edit Export</td></tr><tr><td>security-admin-console</td><td>True</td><td>[Redacted]</td><td>Edit Export</td></tr></table> | Client ID   | Enabled     | Base URL | Actions | account | True | [Redacted] | Edit Export | account-console | True | [Redacted] | Edit Export | admin-camel-mytestbank1 | True | Not defined | Edit Export | admin-cli | True | Not defined | Edit Export | admin-openapi-mytestbank1 | True | Not defined | Edit Export | admin-portal-mytestbank1 | True | Not defined | Edit Export | admin-studio-mytestbank1 | True | Not defined | Edit Export | broker | True | Not defined | Edit Export | realm-management | True | Not defined | Edit Export | security-admin-console | True | [Redacted] |
| Client ID                 | Enabled  | Base URL    | Actions     |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| account                   | True   | [Redacted]  | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| account-console           | True   | [Redacted]  | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| admin-camel-mytestbank1   | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| admin-cli                 | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| admin-openapi-mytestbank1 | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| admin-portal-mytestbank1  | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| admin-studio-mytestbank1  | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| broker                    | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| realm-management          | True   | Not defined | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |
| security-admin-console    | True   | [Redacted]  | Edit Export |          |         |         |      |            |             |                 |      |            |             |                         |      |             |             |           |      |             |             |                           |      |             |             |                          |      |             |             |                          |      |             |             |        |      |             |             |                  |      |             |             |                        |      |            |

| Key Name             | Key Description  |
|----------------------|--|
| openid-client-secret | <p>Your HPFI component's corresponding client secret generated by the FintechOS Identity Provider.</p> <p>In the FintechOS Identity Provider admin console, you can find the client secret in the Credentials tab of your client's page.</p>  <p>The screenshot shows the FintechOS Identity Provider admin console. On the left is a dark blue sidebar with a menu. The main content area is titled 'Admin-studio-mytestbank1' and has a 'Credentials' tab selected. Under the 'Credentials' tab, there is a section for 'Client Authenticator' with a dropdown menu set to 'Client Id and Si'. Below this, a 'Secret' is displayed as '49d19742-694d-4e3f-96c9-' and is highlighted with a red rectangle. To the right of the secret is a 'Regenerate Secret' button. Below the secret section is a 'Registration access token' section with a 'Regenerate registration access token' button.</p> |

| Key Name                  | Key Description   |
|---------------------------|---|
| openid-discovery-endpoint | <p>FintechOS Identity Provider endpoint.<br/>E.g.:<br/>https://myHPFI.myDomain.com/auth/realms/fintechOSRealm/.well-known/openid-configuration<br/>In the FintechOS Identity Provider admin console, you can find the discovery endpoint address in the Realm Settings section.</p>  |

| Key Name            | Key Description  |
|---------------------|--|
| openid-callback-url | URL where the user agent is redirected after a successful login. The default value is {<Studio/Portal base URL>/Account/LogonCallback}. A matching entry must be configured in the FintechOS Identity Provider as a valid redirect URI for the client. |

| Key Name | Key Description   |
|----------|---|
|          | <div><div><div><div><div><div>fintechOS   Identity Provider</div></div></div><div><div>FintechOSRealm</div></div></div><div><div>Configure</div><div>Realm Settings</div><div>Clients</div><div>Client Scopes</div><div>Roles</div><div>Identity Providers</div><div>User Federation</div><div>Authentication</div><div>Manage</div><div>Groups</div><div>Users</div><div>Sessions</div><div>Events</div><div>Import</div><div>Export</div></div></div><div><div>Clients &gt; admin-studio-mytestbank1</div><div>Admin-studio-mytestbank1</div><div><div>Settings</div><div>Revocation</div><div>Credentials</div><div>Sessions</div><div>Roles</div><div>Offline Access</div><div>Client Scopes</div><div>Clustering</div><div>Mappers</div><div>Installation</div><div>Scope</div><div>Authorization</div><div>Service Account Roles</div></div><div><div>Client ID</div><div>admin-studio-mytestbank1</div></div><div><div>Name</div><div>admin-studio-mytestbank1</div></div><div><div>Description</div><div>rest-api</div></div><div><div>Enabled</div><div></div></div><div><div>Always Display in Console</div><div></div></div><div><div>Consent Required</div><div></div></div><div><div>Login Theme</div><div></div></div><div><div>Client Protocol</div><div>openid-connect</div></div><div><div>Access Type</div><div>confidential</div></div><div><div>Standard Flow Enabled</div><div></div></div><div><div>Implicit Flow Enabled</div><div></div></div><div><div>Direct Access Grants Enabled</div><div></div></div><div><div>Service Accounts Enabled</div><div></div></div><div><div>OAuth 2.0 Device Authorization Grant Enabled</div><div></div></div><div><div>OIDC CIBA Grant Enabled</div><div></div></div><div><div>Authorization Enabled</div><div></div></div><div><div>Root URL</div><div></div></div><div><div>Valid Redirect URIs</div><div><div>Account/LogonCallback</div><div></div><div></div><div></div></div></div><div><div>Base URL</div><div></div></div><div><div>Admin URL</div><div></div></div><div><div>Web Origins</div><div></div></div><div><div>Backchannel Logout URL</div><div></div></div><div><div>Backchannel Logout Session Required</div><div></div></div></div></div> |

## (Deprecated) Configuration using web.config keys:

```
<app-settings>
  ...
  <!-- Set FintechOS Identity Provider authentication-->
  <add key="EBSDefaultAuthentication" value="FTOSOIDC" />

  <!-- External authentication provider settings-->
  <add key="core-setting-external-auth-provider-key-
url" value="{AccessTokenPublicKey}" />
  <add key="core-setting-external-auth-provider-
issuer" value="{AccessTokenIssuer}" />

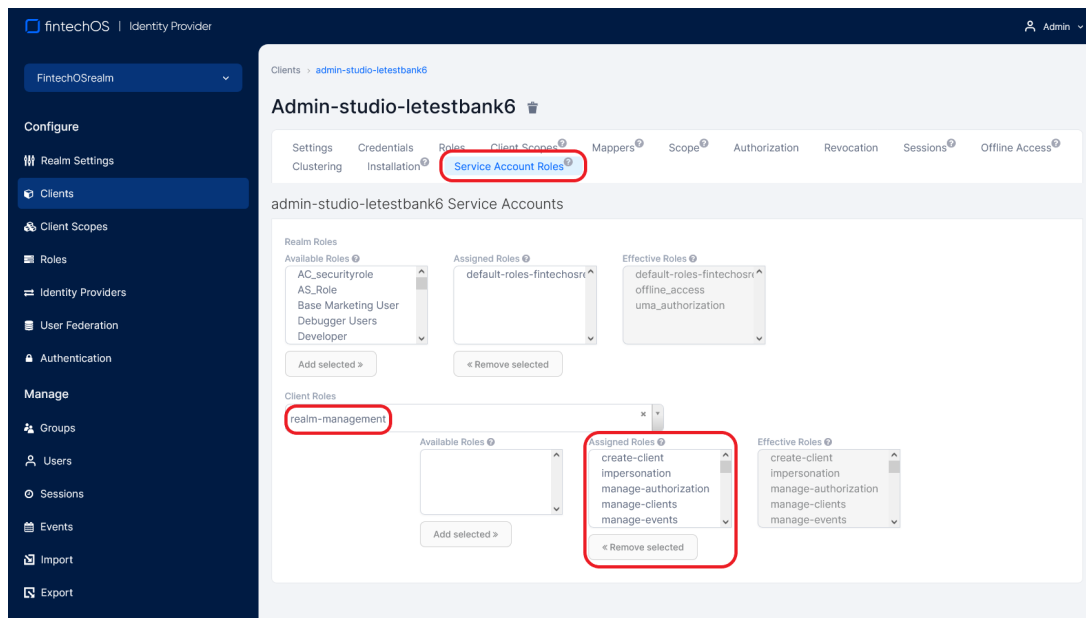
  <!-- Replace these values with your FintechOS Identity
Provider configuration: -->
  <add key="openid-client-id" value="{ClientId}" />
  <add key="openid-client-secret" value="{ClientSecret}"
/>
  <add key="openid-discovery-endpoint" value="
{DiscoveryEndpointUrl}" />
  <add key="openid-callback-url" value="{CallbackUrl}" />
  ...
</app-settings>
```

### Set up Service Account Roles for the Innovation Studio Client

Innovation Studio clients require full privileges for membership management (CRUD operations with users, password change/reset, etc.). For this purpose, a realm management service account role must be assigned to Innovation Studio clients:

1. Log in to the FintechOS Identity Provider admin console.
2. Select your HPFI realm.
3. Select the **Clients** blade.
4. Open the Innovation Studio client.
5. Go to the **Service Account Roles** tab.

6. In the Client Roles drop-down, select **realm-management** and assign all the available roles.



## How users log in the HPFI Portal or Innovation Studio

When accessing the HPFI Portal or Innovation Studio, users who have an active OpenID session are logged in automatically. Otherwise, they are displayed the FintechOS Identity Provider single sign-on login page and will use the OpenID account credentials to log in to the HPFI Portal or Innovation Studio.

## HPFI user account automatic synchronization

When a user logs in to HPFI Portal or Innovation Studio using the FintechOS Identity Provider single sign-on, the following information stored in the corresponding HPFI user account is updated automatically based on the FintechOS Identity Provider account settings:

- First Name
- Last Name
- Email

- Security roles
- External user ID (uniquely identifies the user by an external identity provider)

## Using Azure AD as External Identity Provider

If your organization is using Azure Active Directory (AD) for identity and access management, you can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing Azure AD credentials.

### 1 Register the FintechOS Identity Provider as an Azure App

1. Log in to your **Azure Portal** and navigate to the **Azure Active Directory** blade.
2. Select **App Registrations**.
3. Click **+New registration**.
4. Enter a name for your app and choose who can access it.
5. Add a **Redirect URI** for the FintechOS Identity Provider. The Redirect URI is based on the Identity Provider alias you will set up for the app in the FintechOS Identity Provider and has the following structure:

```
https://{HPFI base URL}/auth/realms/{realm name}/broker/
{alias}/endpoint
```

E.g.:

```
https://myHpfi.myDomain/auth/realms/FintechOSRealm/broker/AzureAD
/endpoint
```

6. **Save** your changes.
7. In the newly created app, select **Certificates and Secrets**.
8. In the **Client secrets** section, click **+New client secret** to generate a secret string for the FintechOS Identity Provider identification.

**(Optional) Configure Access for Azure AD Users**

By default, user assignment is not required, allowing any user to access the app. You should restrict access to the app only to specific assigned users or groups. To do so:

1. In the newly created app, select **Manage Application in Local Directory**.
2. Select **Properties**, and toggle **User assignment required** to Yes.
3. Select **Users and Groups** and assign the desired app users or groups.

**Grant Consent to Access APIs**

Apps are authorized to call APIs when they are granted permissions as part of the consent process. In order to authorize the app:

1. In the newly created app, select **API Permissions**.
2. In the **Grant admin consent for** enter the Azure Directory (tenant) ID.

**2 Set up the Azure AD App as Identity Provider in the FintechOS Identity Provider**

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. From the **Add provider...** drop down, select **OpenID Connect v1.0**.

4. Fill in the following configuration settings for the Azure AD app.

| Setting               | Value   |
|-----------------------|---|
| Alias                 | Identity provider alias you set for the Azure AD app (see " 1 Register the FintechOS Identity Provider as an Azure App" on page 167). |
| Display Name          | User friendly name for the Azure AD app.  |
| Enabled               | ON  |
| Trust Email           | ON  |
| First Login Flow      | Leave the default value or select a custom login flow.  |
| Sync Mode             | Force   |
| Authorization URL     | Use your Azure Tenant ID (found in the App Registration details section of your Azure AD app).  |
| Token URL             | Use your Azure Tenant ID (found in the App Registration details section of your Azure AD app).  |
| Client Authentication | Client secret sent as post.   |
| Client ID             | Use your App Registration Client ID (found in the App Registration details section of your Azure AD app).                             |
| Client Secret         | Use the client secret set up previously (see " 1 Register the FintechOS Identity Provider as an Azure App" on page 167).              |
| Default Scopes        | Scopes to be sent when asking for authorization. Default: openid email.   |

5. Click **Save**.

### **3** Map Azure AD Security Groups to FintechOS Security Roles

When users log in, information about their security roles must be retrieved from Azure AD. For this purpose, you must set up an automatic mapping between Azure AD security groups and FintechOS Identity Provider security roles.

**Set Up the ID Tokens Sent by Azure AD to Include Security Groups Information**

You can include security groups information in the ID tokens sent by Azure AD as an optional claim. To do so, in the Azure app you created earlier (see "[1 Register the FintechOS Identity Provider as an Azure App](#)" on page 167), modify the app registration manifest based on the following model:

```
"optionalClaims": {  
  "idToken": [  
    {  
      "name": "groups",  
      "source": null,  
      "essential": false,  
      "additionalProperties": []  
    }  
  ],  
  "accessToken": [  
    {  
      "name": "groups",  
      "source": null,  
      "essential": false,  
      "additionalProperties": []  
    }  
  ],  
  "saml2Token": [  
    {  
      "name": "groups",  
      "source": null,  
      "essential": false,  
      "additionalProperties": []  
    }  
  ]  
}
```

**Define Mappings between Azure AD Security Groups and FintechOS Security Roles**

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.

3. Select the Azure AD app you created earlier (see " 2 Set up the Azure AD App as Identity Provider in the FintechOS Identity Provider" on page 168).
4. Open the **Mappers** tab.
5. For each security role, do the following:
  - a. Click **Create**.
  - b. In the **Add Identity Provider Mapper** window, fill in the following fields:

| Setting            | Value   |
|--------------------|---|
| Name               | Enter a descriptive name for the mapper.          |
| Sync Mode Override | force   |
| Mapper Type        | Claim to Role                                     |
| Claim              | groups  |
| Claim Value        | GUID of the security group set up in Azure AD.    |
| Role               | Select the corresponding FintechOS security role. |

- c. Click **Save**.

#### **4** Disable User Account Editing in Innovation Studio

Users who authenticate in HPFI via an external identity provider cannot have their user account information edited in Innovation Studio as modifications cannot be propagated back to the external identity provider.

In order to protect the user name, first name, last name, display name, email, and phone number fields, as well as the password reset button in the Innovation Studio interface, a hardcoded *ftos-third-party-brokered-auth-provider* attribute mapping must be provided by the FintechOS Identity Provider for such user accounts:

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. Open your external identity provider and select the **Mappers** tab.

4. Click **Create** to create a new mapper.

5. Fill in the following fields:

- **Name** - Provide a name for your mapper
- **Sync Mode Override** - force
- **Mapper Type** - Hardcoded attribute
- **User attribute** - ftos-third-party-brokered-auth-provider
- **User attribute value** - Any non-null value will work, but it is recommended to use a value that is meaningful for your external identity provider, such as AzureAD or Okta.

6. Click **Save**.

## Using Okta as External Identity Provider

If your organization is using Okta for identity and access management, you can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing Okta credentials.

### 1 Create an Okta App Integration for the FintechOS Identity Provider

Sign in to your Okta admin console with your administrator account and [create an app integration](#) for the FintechOS Identity Provider. The **Sign-in redirect URI** must match the identity provider alias you will set up for Okta in the FintechOS Identity Provider

and has the following structure:

```
https://{HPFI base URL}/auth/realms/{realm name}/broker/  
{alias}/endpoint
```

E.g.:

```
https://myHpfi.myDomain/auth/realms/FintechOSRealm/broker/Okta/e  
ndpoint
```

Make a note of the following Okta settings which you will have to provide in the FintechOS Identity Provider for integration:

- **Okta client ID** (from the Okta app, General tab)
- **Okta client secret** (from the Okta app, General tab)
- **Well-known configuration** - This is an endpoint that returns the OpenID Connect metadata related to the Okta authorization server and has the following URL template:  

```
https://{oktaDomain}.okta.com/oauth2/{oktaServer}/.well-  
known/openid-configuration
```

## **2** Set up the Okta server as Identity Provider in the FintechOS Identity Provider

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. From the **Add provider...** drop down, select **OpenID Connect v1.0**.

## 4. Fill in the following configuration settings for the Okta server.

| Setting               | Value   |
|-----------------------|---|
| Alias                 | Identity provider alias you set for the Okta server (see " <a href="#">1 Create an Okta App Integration for the FintechOS Identity Provider</a> " on page 172). |
| Display Name          | User friendly name for the Okta server.   |
| Enabled               | ON  |
| First Login Flow      | first broker login  |
| Sync Mode             | force   |
| Authorization URL     | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}/authorize</code>  |
| Token URL             | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}/token</code>  |
| Logout URL            | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}/logout</code>   |
| User Info URL         | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}/userinfo</code>   |
| Client Authentication | Client secret sent as post.   |
| Client ID             | Use your Okta client ID (from the Okta app, General tab).   |
| Client Secret         | Use your Okta client secret (from the Okta app, General tab).   |
| Issuer                | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}</code>  |
| Default Scopes        | Scopes to be sent when asking for authorization. Default: openid email.   |
| Prompt                | unspecified   |

| Setting             | Value   |
|---------------------|---|
| Validate Signatures | ON  |
| Use JWKS URL        | ON  |
| JWKS URL            | <code>https://{oktaDomain}.okta.com/oauth2/{oktaServer}/keys</code> |

5. Click **Save**.

### 3 Map Okta User Groups to FintechOS Security Roles

When users log in, information about their security roles must be retrieved from the Okta server. For this purpose, you must set up an automatic mapping between Okta user groups and FintechOS Identity Provider security roles.

#### Set Up the ID Tokens Sent by Okta to Include Security Groups Information

You can include user groups information in the ID tokens sent by Okta as an optional claim. To do so, in the in the Okta portal:

1. Hover over the **API** menu item and select **Authorization Servers**.
2. Select your Okta authorization server.
3. Open the **Claims** tab and click **Add Claim**. Fill in the following fields:

| Field                 | Value             |
|-----------------------|-------------------|
| Name                  | groups            |
| Include In token type | ID Token   Always |
| Value Type            | Groups            |
| Filter                | Regex   .*        |
| Disable claim         | Uncheck           |
| Include In            | Any scope         |

Edit Claim

Name

groups

Include In token type

ID Token

Always

Value type

Groups

Filter

Only include groups that meet the following condition.

Regex

.\*

Disable claim

☐ Disable claim

Include In

☒ Any scope

☐ The following scopes:

Save

Cancel

- Click **Save**.

#### Define Mappings between Okta Groups and FintechOS Security Roles

- Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
- Open the **Identity Providers** section.
- Select the Okta server you added earlier (see " 2 Set up the Okta server as Identity Provider in the FintechOS Identity Provider" on page 173).
- Open the **Mappers** tab.

5. For each security role, do the following:

a. Click **Create**.

b. In the **Add Identity Provider Mapper** window, fill in the following fields:

| Setting            | Value   |
|--------------------|---|
| Name               | Enter a descriptive name for the mapper.          |
| Sync Mode Override | legacy  |
| Mapper Type        | Claim to Role                                     |
| Claim              | groups  |
| Claim Value        | Name of the Okta group set up on the Okta server. |
| Role               | Select the corresponding FintechOS security role. |

c. Click **Save**.

#### **4** Disable User Account Editing in Innovation Studio

Users who authenticate in HPFI via an external identity provider cannot have their user account information edited in Innovation Studio as modifications cannot be propagated back to the external identity provider.

In order to protect the user name, first name, last name, display name, email, and phone number fields, as well as the password reset button in the Innovation Studio interface, a hardcoded *ftos-third-party-brokered-auth-provider* attribute mapping must be provided by the FintechOS Identity Provider for such user accounts:

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. Open your external identity provider and select the **Mappers** tab.
4. Click **Create** to create a new mapper.

5. Fill in the following fields:

- **Name** - Provide a name for your mapper
- **Sync Mode Override** - force
- **Mapper Type** - Hardcoded attribute
- **User attribute** - ftos-third-party-brokered-auth-provider
- **User attribute value** - Any non-null value will work, but it is recommended to use a value that is meaningful for your external identity provider, such as AzureAD or Okta.

Identity Providers > keycloak-oidc > Identity Provider Mappers > Create Identity Provider Mapper

### Add Identity Provider Mapper

Name ⚠ 🔍  
ftos-third-party-brokered-auth-provider

Sync Mode Override ⚠ 🔍  
force

Mapper Type 🔍  
Hardcoded Attribute

User Attribute 🔍  
ftos-third-party-brokered-auth-provider

User Attribute Value 🔍  
externalIDP

**Save** **Cancel**

6. Click **Save**.

## Using AWS Cognito as External Identity Provider

If your organization is using AWS Cognito for identity and access management, you can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing AWS Cognito credentials.

### 1 Create an AWS Cognito App for the FintechOS Identity Provider

Sign in to your AWS Cognito console with your administrator account and [create an app](#) for the FintechOS Identity Provider. The **Callback URL** must match the identity provider alias you will set up for AWS Cognito in the FintechOS Identity Provider and

has the following structure:

```
https://{HPFI base URL}/auth/realms/{realm name}/broker/
{alias}/endpoint
```

E.g.:

```
https://myHpfi.myDomain/auth/realms/FintechOSRealm/broker/awsCog
nito/endpoint
```

Make a note of the following AWS Cognito app settings which you will have to provide in the FintechOS Identity Provider for integration:

- **AWS Cognito client ID**
- **AWS Cognito client secret**
- **AWS Cognito domain**
- **Pool ARN** - The region and pool ID will be extracted from the pool ARN to determine the discovery endpoint. This is an endpoint that returns the OpenID Connect metadata related to the AWS Cognito authorization server and has the following URL template:  

```
https://cognito-idp.{region}.amazonaws.com/{poolId}/.well-known/openid-configuration
```

## 2 Set up the AWS Cognito server as Identity Provider in the FintechOS Identity Provider

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. From the **Add provider...** drop down, select **OpenID Connect v1.0**.
4. Fill in the following configuration settings for the AWS Cognito server.

| Setting | Value   |
|---------|---|
| Alias   | Identity provider alias you set for the AWS Cognito server (see " 1 Create an AWS Cognito App for the FintechOS Identity Provider" on the previous page). |

| Setting               | Value   |
|-----------------------|---|
| Display Name          | User friendly name for the AWS Cognito server.  |
| Enabled               | ON  |
| First Login Flow      | first broker login  |
| Sync Mode             | force   |
| Authorization URL     | <code>https://{Amazon Cognito domain}/oauth2/authorize</code><br>E.g: <code>https://myDomainPrefix.auth.eu-west-1.amazonaws.com/oauth2/authorize</code> |
| Token URL             | <code>https://{Amazon Cognito domain}/oauth2/token</code><br>E.g: <code>https://myDomainPrefix.auth.eu-west-1.amazonaws.com/oauth2/token</code>         |
| User Info URL         | <code>https://{Amazon Cognito domain}/oauth2/userinfo</code><br>E.g: <code>https://myDomainPrefix.auth.eu-west-1.amazonaws.com/oauth2/userinfo</code>   |
| Client Authentication | Client secret sent as post.   |
| Client ID             | Use your AWS Cognito client ID.   |
| Client Secret         | Use your AWS Cognito client secret.   |
| Issuer                | <code>https://cognito-idp.{region}.amazonaws.com/{poolId}</code>  |
| Default Scopes        | Scopes to be sent when asking for authorization. Default: <code>aws.cognito.signin.user.admin email openid phone profile</code> .                       |
| Prompt                | unspecified   |
| Validate Signatures   | ON  |
| Use JWKS URL          | ON  |

| Setting  | Value  |
|----------|--|
| JWKS URL | <code>https://cognito-idp.{region}.amazonaws.com/{poolId}/.well-known/jwks.json</code> |

- Click **Save**.

### 3 Map AWS Cognito User Groups to FintechOS Security Roles

When users log in, information about their security roles must be retrieved from the AWS Cognito server. For this purpose, you must set up an automatic mapping between AWS Cognito user groups and FintechOS Identity Provider security roles.

- Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
- Open the **Identity Providers** section.
- Select the AWS Cognito server you added earlier (see "2 Set up the AWS Cognito server as Identity Provider in the FintechOS Identity Provider" on page 179).
- Open the **Mappers** tab.
- For each security role, do the following:
  - Click **Create**.
  - In the **Add Identity Provider Mapper** window, fill in the following fields:

| Setting               | Value   |
|-----------------------|---|
| Name                  | Enter a descriptive name for the mapper.                        |
| Sync Mode<br>Override | force   |
| Mapper Type           | Claim to Role   |
| Claim                 | cognito:groups  |
| Claim Value           | Name of the AWS Cognito group set up on the AWS Cognito server. |
| Role                  | Select the corresponding FintechOS security role.               |

- c. Click **Save**.

#### **4** Disable User Account Editing in Innovation Studio

Users who authenticate in HPFI via an external identity provider cannot have their user account information edited in Innovation Studio as modifications cannot be propagated back to the external identity provider.

In order to protect the user name, first name, last name, display name, email, and phone number fields, as well as the password reset button in the Innovation Studio interface, a hardcoded *ftos-third-party-brokered-auth-provider* attribute mapping must be provided by the FintechOS Identity Provider for such user accounts:

1. Log in to the FintechOS Identity Provider admin console and select your FintechOS realm.
2. Open the **Identity Providers** section.
3. Open your external identity provider and select the **Mappers** tab.
4. Click **Create** to create a new mapper.
5. Fill in the following fields:
  - **Name** - Provide a name for your mapper
  - **Sync Mode Override** - force
  - **Mapper Type** - Hardcoded attribute
  - **User attribute** - ftos-third-party-brokered-auth-provider
  - **User attribute value** - Any non-null value will work, but it is recommended to use a value that is meaningful for your external identity provider, such as

AzureAD or Okta.

Identity Providers > keycloak-oidc > Identity Provider Mappers > Create Identity Provider Mapper

### Add Identity Provider Mapper

Name ⓘ  
ftos-third-party-brokered-auth-provider

Sync Mode Override ⓘ  
force

Mapper Type ⓘ  
Hardcoded Attribute

User Attribute ⓘ  
ftos-third-party-brokered-auth-provider

User Attribute Value ⓘ  
externalIDP

Save Cancel

6. Click **Save**.

## Deprecated Identity Providers

### IMPORTANT!

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services. Alternate identity providers are supported only for backward compatibility. For more information, see "[FintechOS Identity Provider](#)" on [page 157](#).

## Microsoft Active Directory Authentication

If your organization is using Microsoft Active Directory (AD) as central user repository, you can configure HPFI to give users the possibility to log in HPFI using their existing AD credentials.

HPFI supports interoperability with AD using two configurations: AD standard configuration and AD configuration using a configuration file in which you map the business units and the security roles from HPFI with the ones in AD.

To avoid unnecessary traffic across domains and return results promptly with maximum speed, you can limit the scope of Active Directory queries. For more information, see [Limiting scope of the query on Active Directory](#).

This section covers the following topics:

---

### AD Standard Login Configuration

In order to change the default HPFI authentication with the Microsoft Active Directory authentication, add/edit the following secret:

| Key Path                                    | Key Name                 | Key Value |
|---|--------------------------|-----------|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication | AD        |

You are still able to log in using the administrator host credentials (using the password from HPFI authentication).

#### NOTE

- When adding system users in HPFI who will be using AD credentials for logging in, in the **UserName** field, you should provide the username in the following format: **[Domain]\[Username]**. When logging in HPFI, users should provide the username in the format previously mentioned.
- Every AD has different security roles, so make sure that the Application Pool Identity of the HPFI WebApp has the privileges to search into the directory entry nodes, otherwise, when trying to log in HPFI using AD credentials, privileges related errors might occur.

## (Deprecated) Add key in the **web.config** file:

Go to the web.config file of your WebApp (Portal/Designer) and add/edit the following setting:

```
<app-settings>
...
<add key="EBSDefaultAuthentication" value="AD"/>
```

```
...
</app-settings>
```

### Automatically Adding Users from AD

You can automatically create / update users from Microsoft AD in HPFI using a configuration file.

**NOTE** Automatically creating users from AD will remove the existing business units and security roles from HPFI and add the ones from AD as provided in the configuration file. If you want to keep the system user as is, you should make additional settings. For information on the additional settings, see [Preserve System User's Business Unit and Security Roles](#).

**IMPORTANT!** In FintechOS versions prior 18.2.8, getting from the Active Directory (AD) the groups to whom a user belongs to did not work smoothly; therefore, there might be situations in which wrong security roles were applied to users. With version 18.2.8, the existing configurations for mapping AD groups-roles (specified in the `~\ADUserConfiguration.xml` file) might not work as it worked in previous versions of FintechOS.

To automatically create/update users in HPFI using a configuration file, follow these steps:

1. Add the following secret in **Vault**:

| Key Path                                    | Key Name                 | Key Value |
|---|--------------------------|-----------|
| kv/<environment>/<application>/app-settings | EBSADAuthAutoCreateUsers | true      |

(Deprecated) Add key in the **web.config** files:

```
<app-settings>
...
<add key="EBSADAuthAutoCreateUsers" value="true"/>
...
</appSetting
```

2. In an xml file, create the mapping between the AD groups and the security roles and business units from HPFI. Name the file **ADUserConfiguration.xml**.

Overwrite the Business Unit from HPFI with the business unit from AD.

```
<ADUserConfiguration>
  <SecurityGroup>
    <Name>`AD Group Name`</Name>
    <DefaultBusinessUnitName>`FTOS Business Unit
Name`</DefaultBusinessUnitName>
    <SecurityRoleName>`FTOS Security Role
Name`</SecurityRoleName>
  </SecurityGroup>
</ADUserConfiguration>
```

3. In the root of the WebApp, add the **ADUserConfiguration.xml** file previously created.

### Preserving System Users

To preserve the system user's business unit from HPFI, go to **Vault** and add the following:

| Key Path                                    | Key Name                | Key Value |
|---|-------------------------|-----------|
| kv/<environment>/<application>/app-settings | ADOverwriteBusinessUnit | false     |

### (Deprecated) Adding key in the **web.config** file:

```
<app-settings>
...
<add key="ADOverwriteBusinessUnit" value="false"/>
...
</app-settings>
```

To preserve the system user's security roles from HPFI and merge them with the ones provided :

| Key Path                                    | Key Name             | Key Value |
|---|----------------------|-----------|
| kv/<environment>/<application>/app-settings | ADOverwriteUserRoles | false     |

## (Deprecated) Add key in the **web.config** file:

```
<app-settings>
...
<add key="ADOverwriteUserRoles" value="false"/>
...
</app-settings>
```

### Limiting Query Scope on AD

By default, the Lightweight Directory Access Protocol (LDAP) queries are performed on the entire Active Directory (AD).

To avoid unnecessary traffic across domains and return results promptly with maximum speed, limit the scope of active directory queries by adding the following app-settings keys in Vault:

- for queries related to users, add the key core-setting-adauth-users-container
- for queries related to groups, add the key core-setting-adauth-groups-container.

When AD authentication is enabled, the HPFI platform will use the values provided in the app-settings keys.

The keys are optional, if they are not provided the search will be performed on the entire directory.

Setting the users and groups containers in **Vault** secrets:

| Key Path                                    | Key Name                            | Key Value                    |
|---|-------------------------------------|------------------------------|
| kv/<environment>/<application>/app-settings | core-setting-adauth-users-container | OU=Utilizatori,DC=acme,DC=ro |

| Key Path                                    | Key Name   | Key Value                |
|---|--|--------------------------|
| kv/<environment>/<application>/app-settings | core-setting-<br>adauth-<br>groups-<br>container | OU=Grupuri,DC=acme,DC=ro |

In the example above, the LDAP queries will be performed against the following AD containers:

#### Users:

- Organizational Unit (OU): Utilizatori
- Domain Component (DC): ro

#### Groups:

- Organizational Unit (OU): Grupuri
- Domain Component (DC): ro

## (Deprecated) Setting the users and groups containers in the `web.config` file:

```
<app-settings>
  <add key="core-setting-adauth-users-
container" value="OU=Utilizatori,DC=acme,DC=ro"/>
  <add key="core-setting-adauth-groups-
container" value="OU=Grupuri,DC=acme,DC=ro"/>
  ....
</app-settings>
```

### Customizing Group Membership Checks

To comply with the querying rights set up for your configuration, you can customize how the platform checks if an Active Directory user belongs to a specific Active Directory group.

To do so, add the following secrets in **Vault**:

| Key Path                                    | Key Name                             | Key Value |
|---|--------------------------------------|-----------|
| kv/<environment>/<application>/app-settings | core-setting-adauth-group-query-mode | 1 or 2    |

- Setting the value to **1** uses the `user.GetAuthorizationGroups()` method to retrieve all the groups the user account belongs to, then loops them to see if the target group is among them.
- Setting the value to **2** uses the `user.IsMemberOf(group)` method to query directly if the user account is part of the target group.

Default value: 1.

## (Deprecated) Add key in the web.config files

In the **web.config** file add the `core-setting-adauth-group-query-mode` key in the `<app-settings>` section:

```
<app-settings>
...
<add key="core-setting-adauth-group-query-mode" value="1 or 2"/>
...
</app-settings>
```

## Azure Active Directory Authentication

### IMPORTANT!

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services. You can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing Azure AD credentials. For more information, see ["Using Azure AD as External Identity Provider" on page 167](#).

This authentication method is provided only for backward compatibility.

If your organization is using Azure Active Directory (Azure AD) for identity and access management, you can map Azure groups to HPFI ["Security Roles" on page 254](#) using the OpenID authentication protocol. This allows users to log in to HPFI using their existing Azure AD credentials.

### Configure OpenID Settings

Configuration using **Vault** secrets:

| Key Path                                    | Key Name                 | Key Value |
|---|--------------------------|-----------|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication | AzureAD   |

Azure openid configuration:

| Key Path                                    | Key Name                  | Key Value   |
|---|---------------------------|---|
| kv/<environment>/<application>/app-settings | openid-client-id          | Azure directory (tenant) id (GUID)  |
| kv/<environment>/<application>/app-settings | openid-application-id     | Azure application id (GUID)   |
| kv/<environment>/<application>/app-settings | openid-client-secret      | Azure application secret  |
| kv/<environment>/<application>/app-settings | openid-callback-url       | http://\${portalRoot}/Account/LogonCallback                                     |
| kv/<environment>/<application>/app-settings | openid-discovery-endpoint | https://login.microsoftonline.com/\${tenantId}/.well-known/openid-configuration |

User mapping settings:

|   |                               |                                  |
|---|-------------------------------|----------------------------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-roles        | Guest,Developer,Registered Users |
| kv/<environment>/<application>/app-settings | openid-auto-user-organization | ebs                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-businessunit | root                             |

|   |                                    |             |
|---|------------------------------------|-------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-type              | Back Office |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-add  | 0 1         |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-sync | 0 1         |

**Configuration Keys**

| Parameter                          | Value   |
|------------------------------------|---|
| openid-auto-user-roles             | Platform role names, separated by commas. These roles will be added automatically when the Azure AD user is mapped to a HPFI user.  |
| openid-auto-user-organization      | Platform organization name. The mapped user will be added in this organization.   |
| openid-auto-user-businessunit      | Platform business unit name. The mapped user will be added in this business unit.   |
| openid-auto-user-remote-roles-add  | When set to 1, the roles from Azure AD will be added to the mapped user on user creation, adding the roles found in the values for web.config key="openid-auto-user-roles" (has effect only at user creation). See below how to expose the Azure AD roles in custom claims consumable by HPFI. Azure AD will be added to the mapped user, |
| openid-auto-user-remote-roles-sync | When set to 1, the roles from Azure AD and the default roles are always synchronized at login. Any roles manually added to Azure AD user are lost.  |

**Parameters**

| Parameter      | Value                              |
|----------------|------------------------------------|
| \${portalRoot} | Root URL for the HPFI web service. |
| \${tenantId}   | Azure tenant ID.                   |

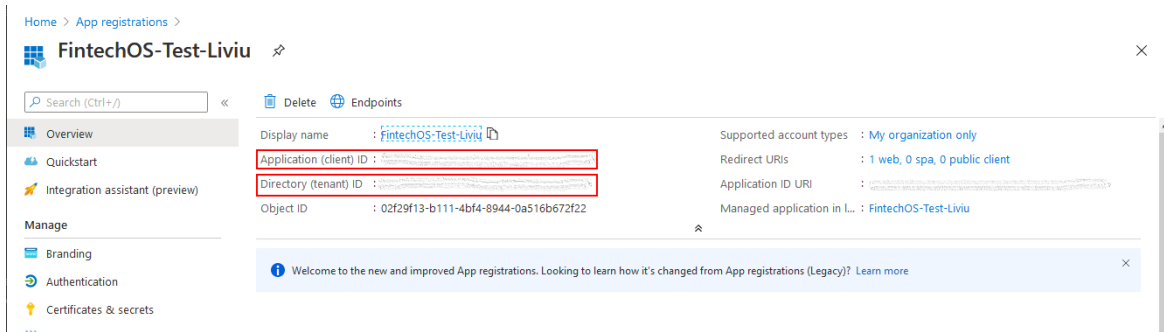
(Deprecated) Configuration using **web.config** files:

Add the following keys to the <app-settings> node in the *web.config* file of your web service (Portal/Studio):

```
<add key="EBSDefaultAuthentication" value="AzureAD" />
<!-- BEGIN AzureAD IDOPEN ID CONFIGURATION -->
<add key="openid-client-id" value="Azure directory (tenant)
id (GUID)" />
<add key="openid-application-id" value="Azure application id
(GUID)"/>
<add key="openid-client-secret" value="Azure application
secret"/>
<add key="openid-callback-
url" value="http://${portalRoot}/Account/LogonCallback" />
<add key="openid-discovery-
endpoint"
value="https://login.microsoftonline.com/${tenantId}/.well-
known/openid-configuration" />
<!-- USER MAPPING SETTINGS -->
<add key="openid-auto-user-roles" value="Registered User,My
default role" />
<add key="openid-auto-user-organization" value="ebs" />
<add key="openid-auto-user-businessunit" value="root" />
<add key="openid-auto-user-type" value="Back Office" />
<add key="openid-auto-user-remote-roles-add" value="0 or
1"/>
<add key="openid-auto-user-remote-roles-sync" value="0 or
1"/>
<!-- END AzureAD IDOPEN ID CONFIGURATION -->
```

To find the *Azure directory (tenant) id (GUID)* and the *Azure application id (GUID)*:

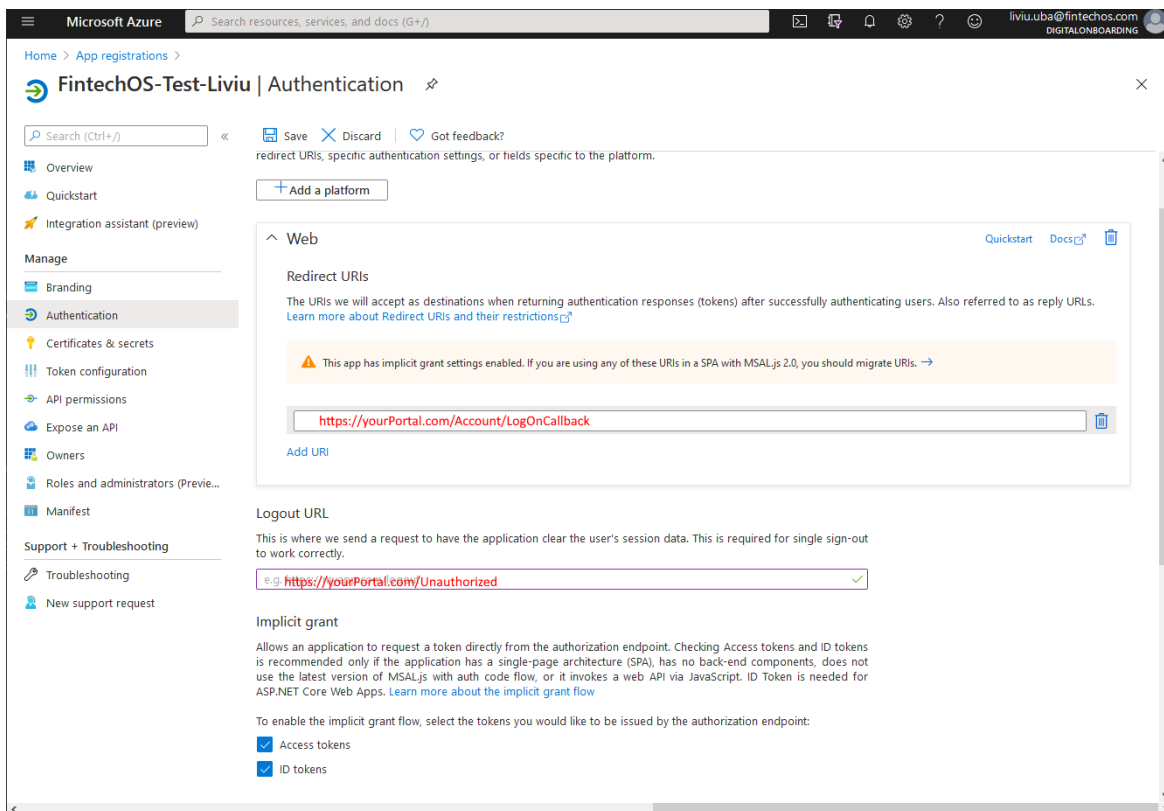
1. Open the **Azure Portal**.
2. Select the **App registrations** service.
3. Select the application you wish to use as a source for identity credentials.
4. The *Azure directory (tenant) id (GUID)* and the *Azure application id (GUID)* will be displayed in the Overview section of the application.



## Set up Login/Logout Redirect URIs

In the Azure Portal, in the Authentication section of your registered application, fill in the:

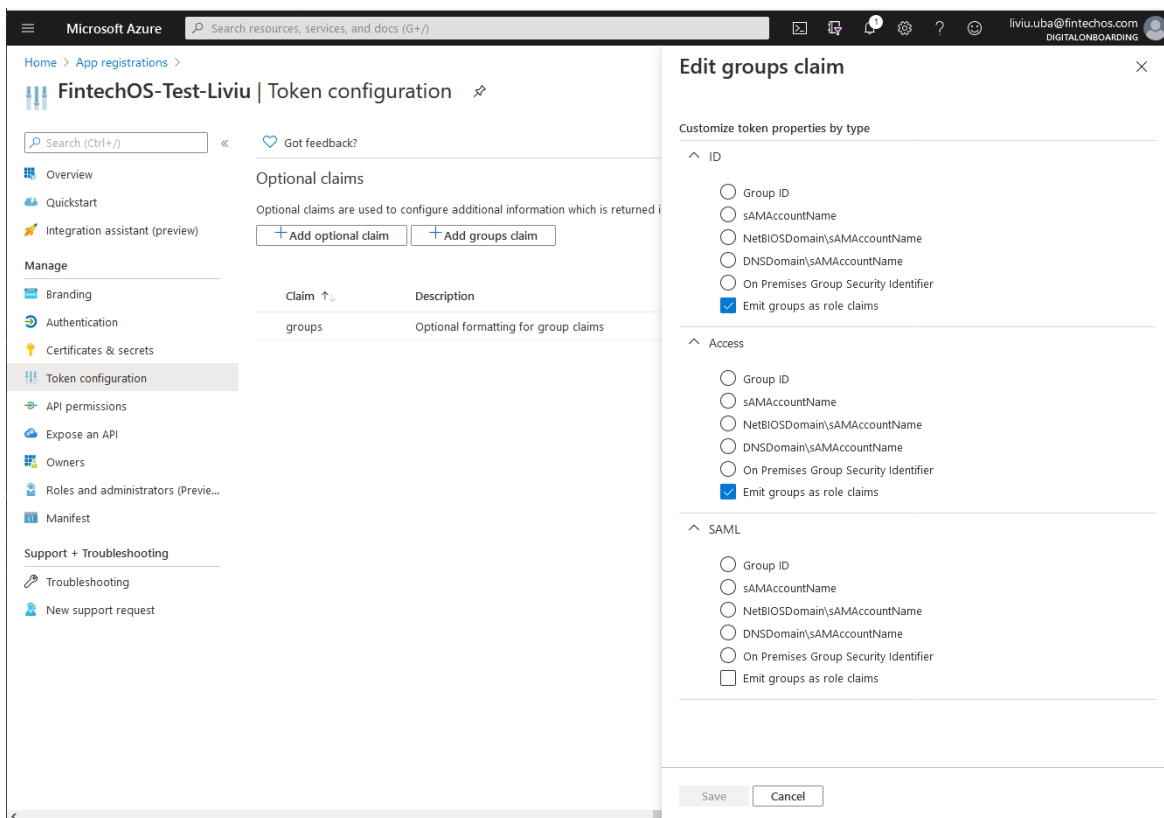
- Login redirect URI: `{ $portalRoot }/Account/LogonCallback`
- Logout redirect URI: `{ $portalRoot }/Unauthorized`



## Groups Mapping

When a user is authorized with Azure AD, a corresponding system user is created in HPFI. Default roles for this user, organization, business unit, and user type can be configured in *web.config*. Any Security Role which has not already been created in the system and is mentioned in *OpenIdUserConfiguration.xml* it will be automatically created.

To synchronize groups created in Azure AD with HPFI, the administrator of the Azure AD application must include an optional claim named **groups** in the token configuration.



To configure the mappings, an XML file named *OpenIdUserConfiguration.xml* must be placed in the root folder of the web application. Azure AD sends group IDs with the OpenID token, so the mapping must be done between the Azure Group ID and QWPlatform security roles.

```
<root>
  <SecurityGroup>
```

```

    <Name>b681734a-5601-435c-b817-465f8e20b7fb</Name>
    <DisplayName>GROUP1</DisplayName>
    <DefaultBusinessUnitName>root</DefaultBusinessUnitName>
    <SecurityRoleName>Registered Users</SecurityRoleName>
  </SecurityGroup>
  ...

  <SecurityGroup>
    <Name>84d47d54-0956-4f9a-b37d-81880374fd46</Name>
    <DisplayName>GROUP2</DisplayName>
    <DefaultBusinessUnitName>root</DefaultBusinessUnitName>
    <SecurityRoleName>Developer, Registered
Users</SecurityRoleName>
  </SecurityGroup>
</root>

```

**IMPORTANT!**

Any changes to `OpenIdUserConfiguration.xml` require a manual Application Domain restart.

## Authentication with Okta

**IMPORTANT!**

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services. You can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing Okta credentials. For more information, see ["Using Okta as External Identity Provider" on page 172](#).

This authentication method is provided only for backward compatibility.

Okta is a standards-compliant OAuth 2.0 authorization server and a certified OpenID Connect provider.

HPFI built-in integration with Okta enables users to log in to the Digital Experience Portal using the Okta single-sign on (SSO).

### How to Set up the Okta Authentication

To set up the Okta authentication for your Experience Portal, follow these steps:

**Step 1. Create and configure the Okta app**

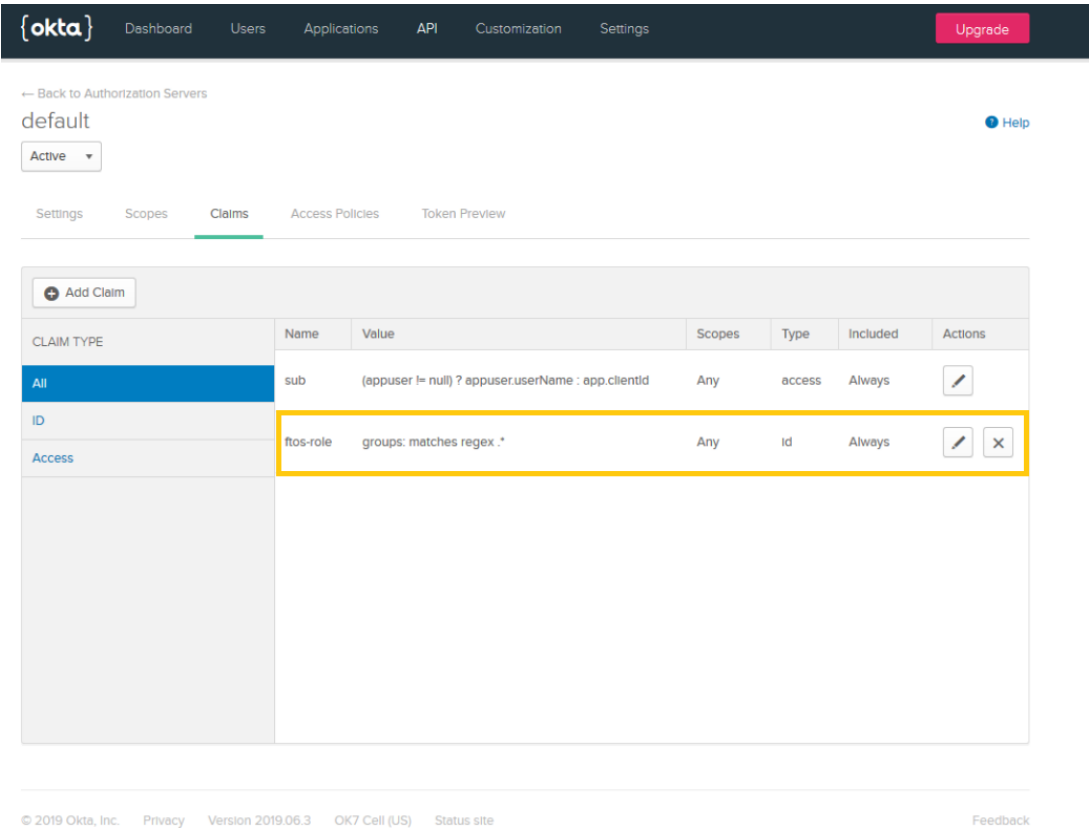
1. Using an Okta admin account, log into Okta and create an Okta application (Application tab > Web > OpenID Connect).
2. From the Applications tab > General > Login, set up the HPFI callbacks by configuring both the login and the logout redirect URLs, as follows:

login redirect uri                      **`{portalRoot}/Account/LogonCallback`**

logout redirect uri                    **`{portalRoot}/Unauthorized`**

3. From the **API** tab > **Authorization Servers**, create an authorization server for the Okta application.
4. Expose the Okta roles in custom claims consumable by HPFI. To do so, synchronize the user groups created in Okta with HPFI by creating a custom claim named **ftos-role** mapped to the group metadata in Okta. For more information on how to create a

custom claim in the Okta app, see [Okta Documentation](#).



When a user is authorized with Okta, a corresponding system user will be created in HPFI . In the **web.config** file you can configure default roles for this user, organization, business unit and user type.

**Step 2. Configure the Experience Portal**

**Prerequisite:**

Make sure that you know the following values:

- Client ID (from the Okta app, General tab)
- Client Secret (from the Okta app, General tab)
- Discovery Endpoint (from the Okta app, API section > Authorization Servers > Metadata URL)

Configuration using **Vault** secrets:

Set Okta authentication:

| Key Path                                    | Key Name                 | Key Value |
|---|--------------------------|-----------|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication | Okta      |

Replace the keys' value with your Okta configuration:

| Key Path                                    | Key Name                  | Key Value   |
|---|---------------------------|---|
| kv/<environment>/<application>/app-settings | openid-client-id          | {ClientId}  |
| kv/<environment>/<application>/app-settings | openid-client-secret      | {ClientSecret}  |
| kv/<environment>/<application>/app-settings | openid-callback-url       | http://\${portalRoot}/Account/LogonCallback   |
| kv/<environment>/<application>/app-settings | openid-discovery-endpoint | https://\${oktaApplication}.okta.com/oauth2/\${authServerId}/.well-known/oauth-authorization-server |

User mapping settings:

| Key Path                                    | Key Name                           | Key Value                        |
|---|------------------------------------|----------------------------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-roles             | Guest,Developer,Registered Users |
| kv/<environment>/<application>/app-settings | openid-auto-user-organization      | ebs                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-businessunit      | root                             |
| kv/<environment>/<application>/app-settings | openid-auto-user-type              | Back Office                      |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-add  | 0 1                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-sync | 0 1                              |

The table below describes the Okta app configuration keys:

| Key                 | Description   |
|---------------------|---|
| \${portalRoot}      | The root URL of the Experience Portal.  |
| \${authServerId}    | The ID of the authorization server associated with the Okta application (default value is default). |
| \${oktaApplication} | The ID of the Okta application.   |
| Key                 | Description   |

The table below describes the user mapping configuration keys.

| Parameter                          | Value  |
|------------------------------------|--|
| openid-auto-user-roles             | The platform role names, separated by colon. These roles will be added automatically when the Okta user is mapped to a platform user.        |
| openid-auto-user-organization      | The platform organization name. The mapped user will be added in this organization.  |
| openid-auto-user-businessunit      | The platform business unit name. The mapped user will be added in this business unit.  |
| openid-auto-user-remote-roles-add  | If set to 1, the roles from the Okta app will be added to the mapped user.   |
| openid-auto-user-remote-roles-sync | If value is 1, the roles from Okta and the default roles are always synchronized at login. Any roles manually added to a Okta user are lost. |

## (Deprecated) Configuration using **web.config** files:

Go to the <app-settings> section and add the configuration of your Okta application:

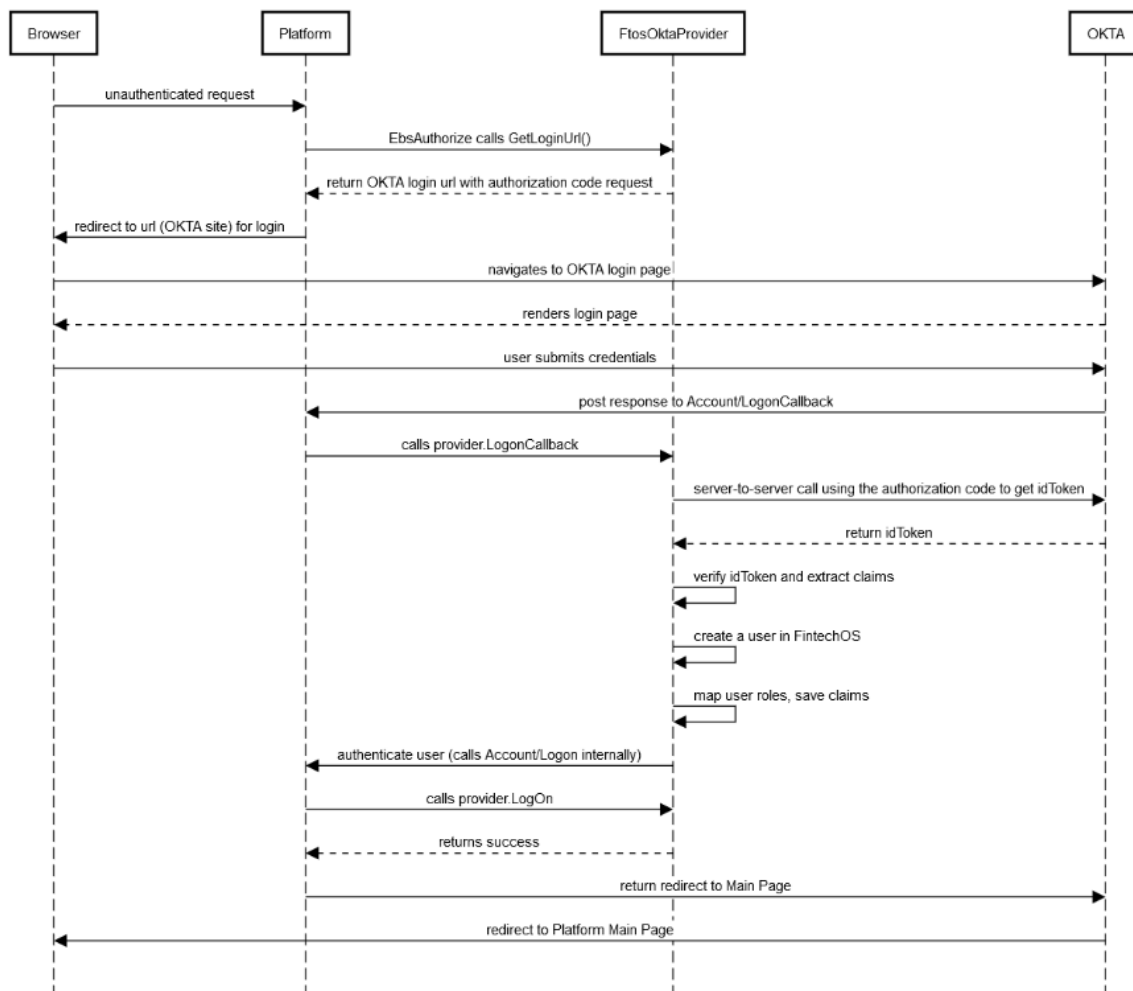
```
<!-- 1. Set Okta authentication-->
<add key="EBSDefaultAuthentication" value="Okta" />

<!-- 2. Replace these values with your Okta configuration: -
->
<add key="openid-client-id" value="{ClientId}" />
<add key="openid-client-secret" value="{ClientSecret}" />
```

```
<add key="openid-callback-  
url" value="http://${portalRoot}/Account/LogonCallback" />  
<add key="openid-discovery-endpoint" value=  
"https://${oktaApplication}.okta.com/oauth2/${authServerId}/  
.well-known/oauth-authorization-server" />  
  
<!-- 3. Map user settings: -->  
<add key="openid-auto-user-  
roles" value="Guest,Developer,Registered Users" />  
<add key="openid-auto-user-organization" value="ebs" />  
<add key="openid-auto-user-businessunit" value="root" />  
<add key="openid-auto-user-type" value="Back Office" />  
  
<add key="openid-auto-user-remote-roles-add" value="0|1"/>  
<add key="openid-auto-user-remote-roles-sync" value="0|1"/>
```

### How it Works

The diagram below describes the HPFI login flow when using Okta authentication.





### Group mapping in FintechOS

When a user is authorized with Okta, a corresponding system user is created in FintechOS. In web.config file of the FintechOS instance, default roles for this user, organization, business unit and user type are added.

Create a custom claim named **ftos-role** mapped to the group metadata in Okta. This configuration is done for the authorization server associated with the Okta application.

### How users log in the Portal

When accessing the Digital Experience Portal URL, users will be redirected to the URL of the authorization server associated with the Okta app. The Okta login page appears.

  
  
**Sign In**  
  
  
☐ Remember me  
**Sign In**  
[Need help signing in?](#)

Once they provide Okta account credentials, they will be logged into the Digital Experience Portal.

When new users are created, they will receive an email notification from Okta which contains instructions and Okta credentials.

#### **Troubleshooting Okta Redirect Error**

##### **Error**

UnhandledException: System.Web.HttpException (0x80004005): Server cannot set status after HTTP headers have been sent.

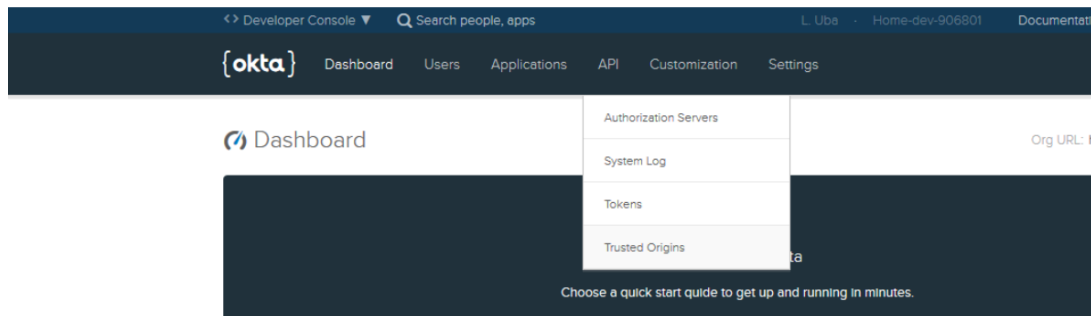
### Cause

FTOS OpenID provider does not redirect when the user is still logged in due to the OpenID cookie not being expired too.

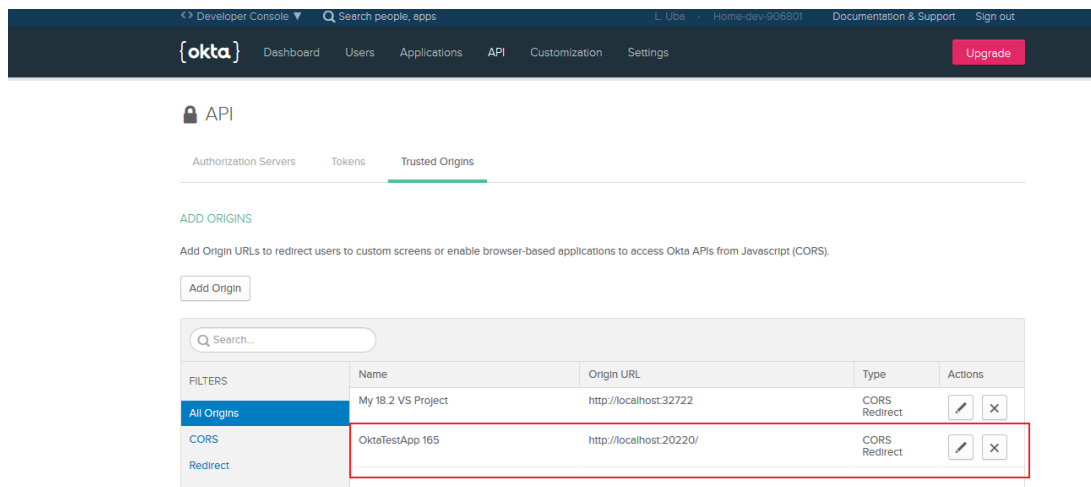
### What should I do?

For session expiration to work seamlessly, follow these steps:

1. Using an Okta admin account, log into Okta.
2. Click API tab > Trusted Origins.



3. Allow CORS and Redirect for FTOS portal host.



## Authentication with Active Directory Federation Services

This service provided by Microsoft manages the user sign-in information for members of a platform. If your organization is using ADFS for identity and access management of your users, it is possible to map the users already existing in ADFS to FintechOS [Security Roles](#). When a user is authorized with ADFS, a corresponding system user is created in FintechOS. Through ADFS OpenId, users to log in to FintechOS using their existing ADFS credentials.

### Add keys to Vault secrets

| Key Path                                    | Key Name                 | Key Value |
|---|--------------------------|-----------|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication | ADFS      |

### ADFS configuration:

| Key Path                                    | Key Name                  | Key Value   |
|---|---------------------------|---|
| kv/<environment>/<application>/app-settings | openid-client-id          | Client identifier configured in ADFS                    |
| kv/<environment>/<application>/app-settings | openid-application-id     | this value is not used                                  |
| kv/<environment>/<application>/app-settings | openid-client-secret      | ADFS Web API shared secret                              |
| kv/<environment>/<application>/app-settings | openid-callback-url       | http://{portalRoot}/Account/LogonCallback               |
| kv/<environment>/<application>/app-settings | openid-discovery-endpoint | {adfs server uri}/adfs/.well-known/openid-configuration |

### User mapping settings:

| Key Path                                    | Key Name                      | Key Value                        |
|---|-------------------------------|----------------------------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-roles        | Registered User, My default role |
| kv/<environment>/<application>/app-settings | openid-auto-user-organization | ebs                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-businessunit | root                             |

| Key Path                                    | Key Name                           | Key Value   |
|---|------------------------------------|-------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-type              | Back Office |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-add  | 0 1         |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-sync | 0 1         |

**Configuration Keys:**

| Key                                | Value  |
|------------------------------------|--|
| openid-auto-user-roles             | Platform role names, separated by colon. These roles will be added automatically when the AD user is mapped to a platform user                                     |
| openid-auto-user-organization      | Platform organization name. The mapped user will be added in this organization   |
| openid-auto-user-businessunit      | Platform business unit name. The mapped user will be added in this business unit   |
| openid-auto-user-remote-roles-add  | when value is 1 the roles from AD will be added to the mapped user on user creation. See below how to expose the AD roles in custom claims consumable by FintechOS |
| openid-auto-user-remote-roles-sync | when value is 1 the roles from AD and the default roles are always synchronized at login. Any roles manually added to a AD user are lost                           |

**Parameters:**

| Parameter         | Value                         |
|-------------------|-------------------------------|
| {portalRoot}      | root url for FintechOS portal |
| {adfs server url} | ADFS server url               |

## (Deprecated) Add keys to the web.config file

```

<add key="EBSDefaultAuthentication" value="ADFS" />

<!-- BEGIN ADFS OPENID CONFIGURATION -->

<add key="openid-client-id" value="Client identifier
configured in ADFS" />

<add key="openid-application-id" value=" this value is
not used" />

```

```

    <add key="openid-client-secret" value="ADFS Web API
shared secret"/>

    <add key="openid-callback-url" value="http://
{portalRoot}/Account/LogonCallback" />

    <add key="openid-discovery-endpoint" value="{adfs server
uri}/adfs/.well-known/openid-configuration" />

    <!-- USER MAPPING SETTINGS -->

    <add key="openid-auto-user-roles" value="Registered
User,My default role" />
    <add key="openid-auto-user-organization" value="ebs" />
    <add key="openid-auto-user-businessunit" value="root" />
    <add key="openid-auto-user-type" value="Back Office" />

    <add key="openid-auto-user-remote-roles-
add" value="0|1"/>
    <add key="openid-auto-user-remote-roles-
sync" value="0|1"/>

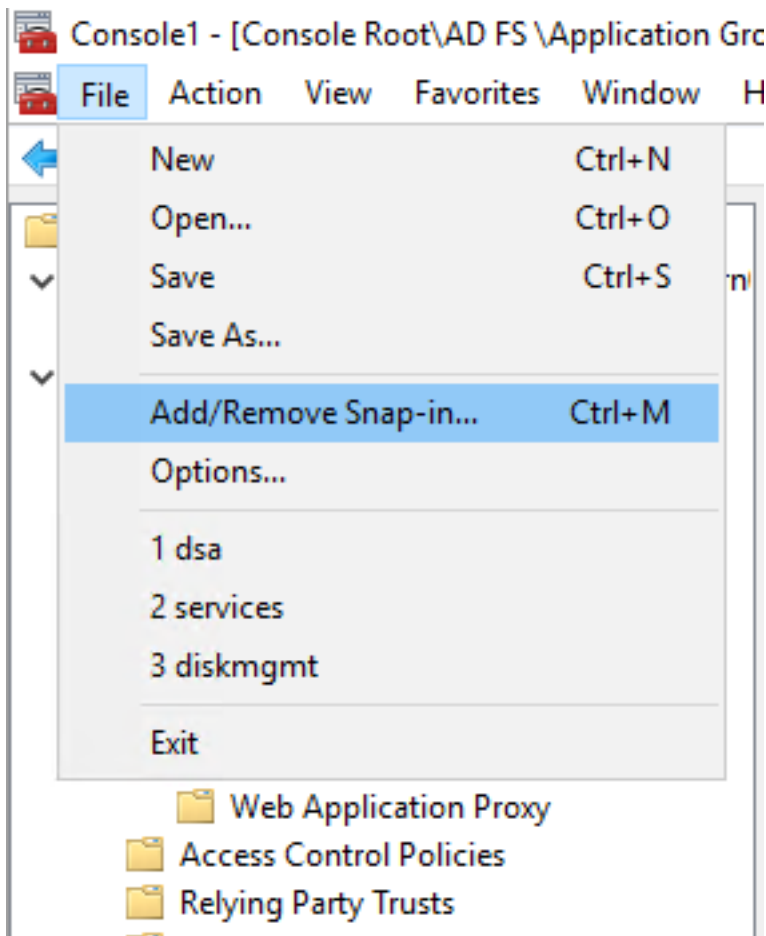
    <!-- END ADFS OPENID CONFIGURATION -->

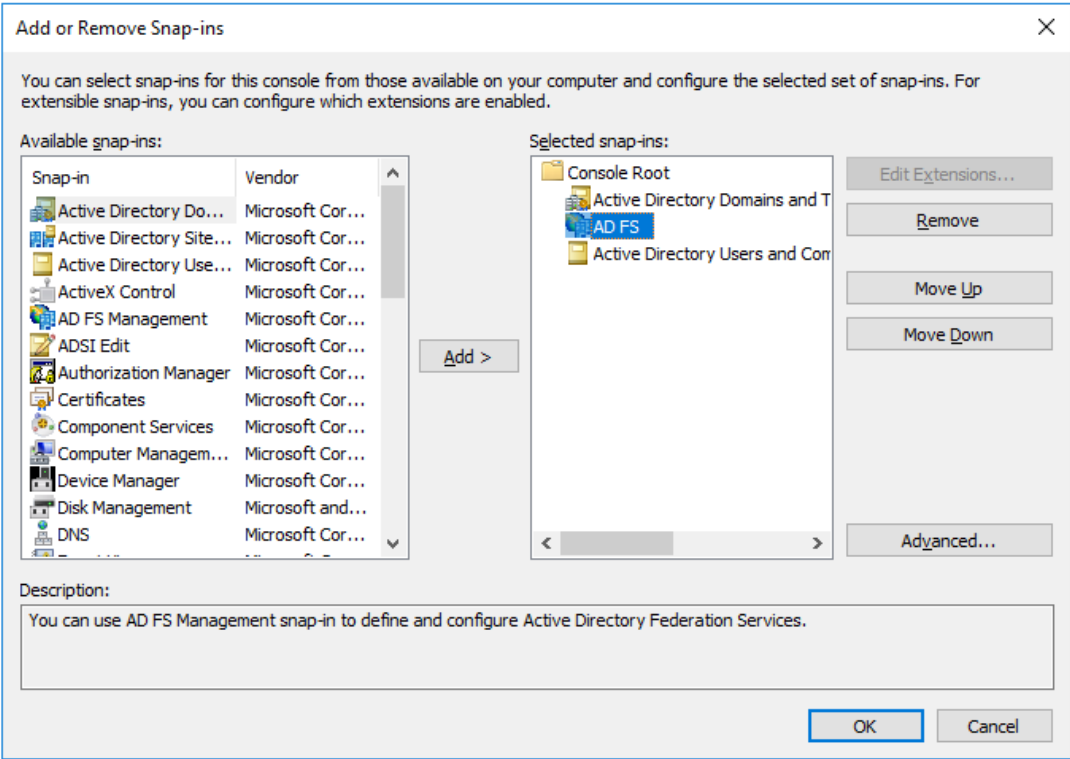
```

### ADFS configuration

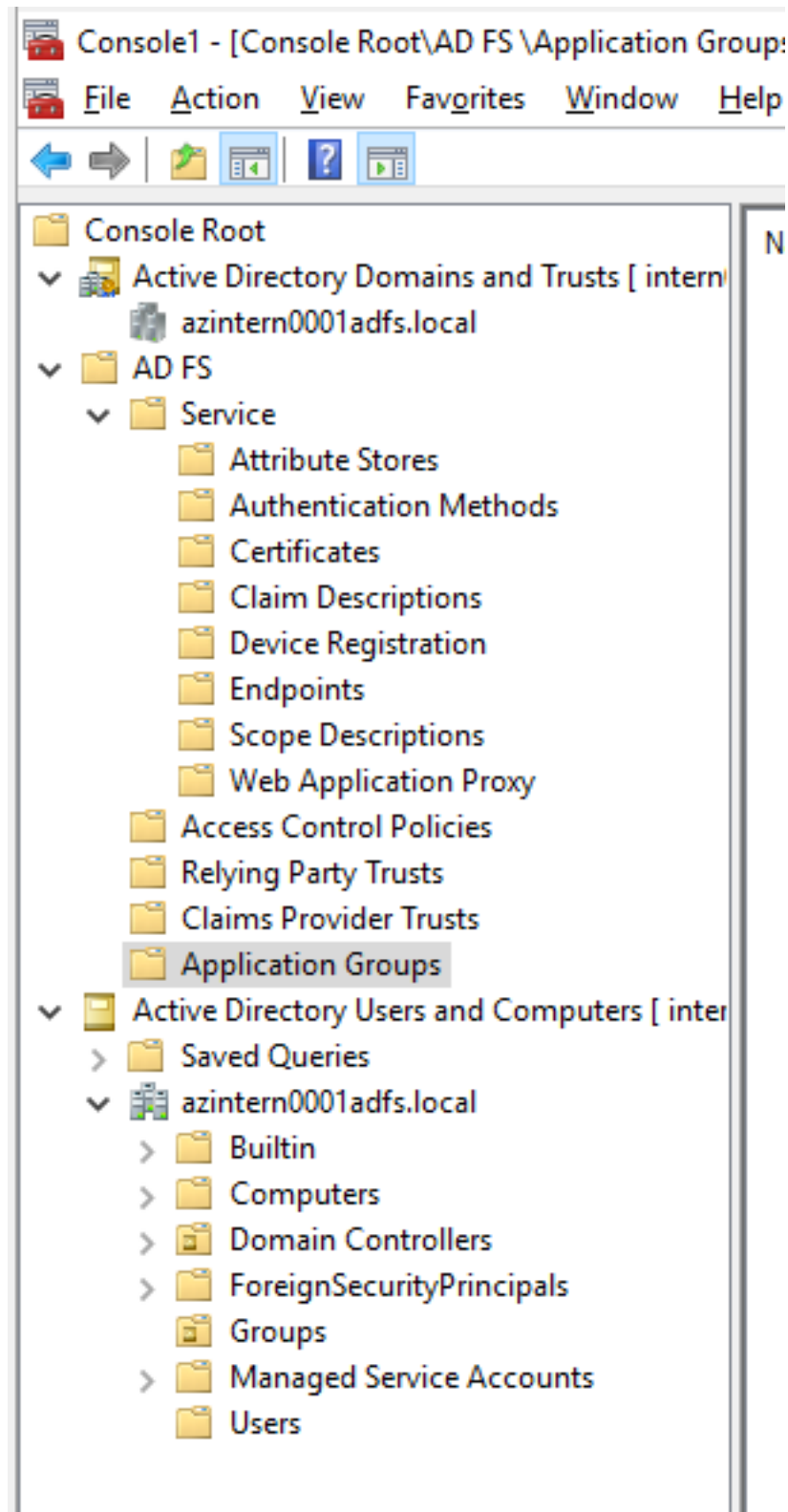
1. On a Windows Server 2016+, on the ADFS server open the Microsoft Management Console (mmc).

2. Add the ADFS snap in if not already added.

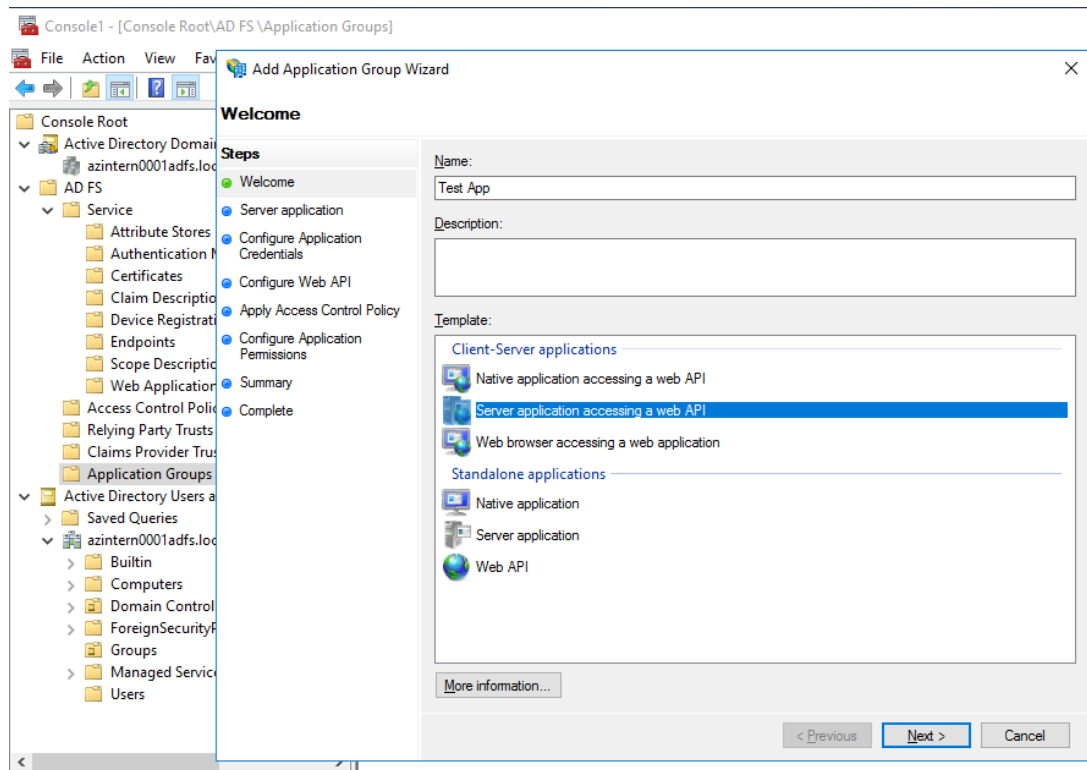




3. Open the ADFS MMC plugin and select the node Application Groups.

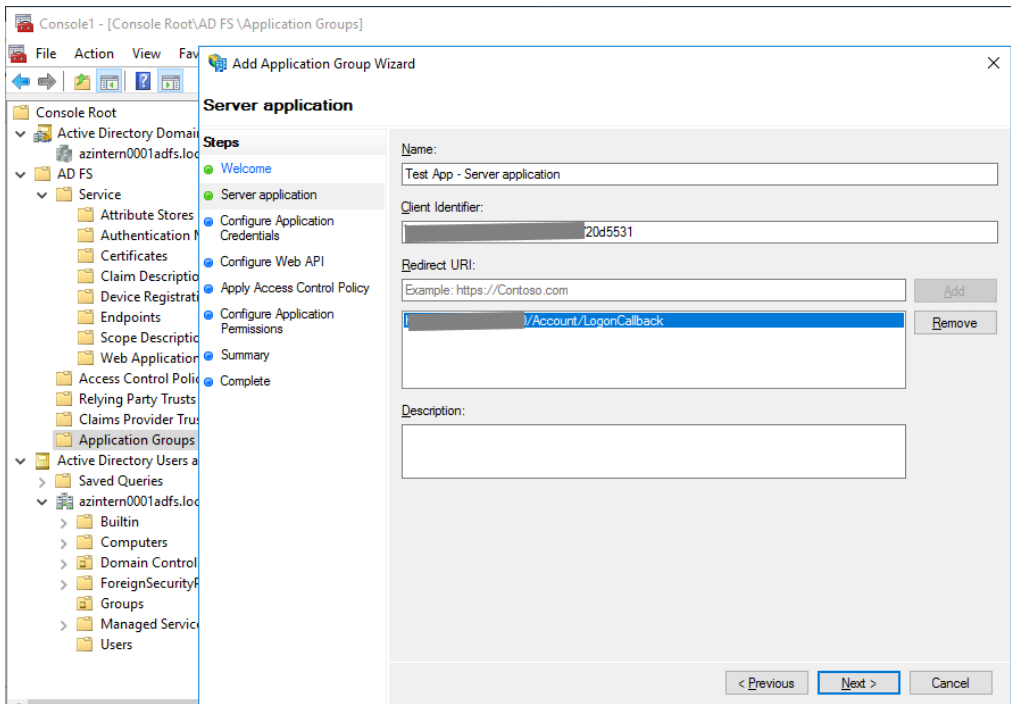


4. Right click and select Add application group. In the template list select Server application accessing a web API.

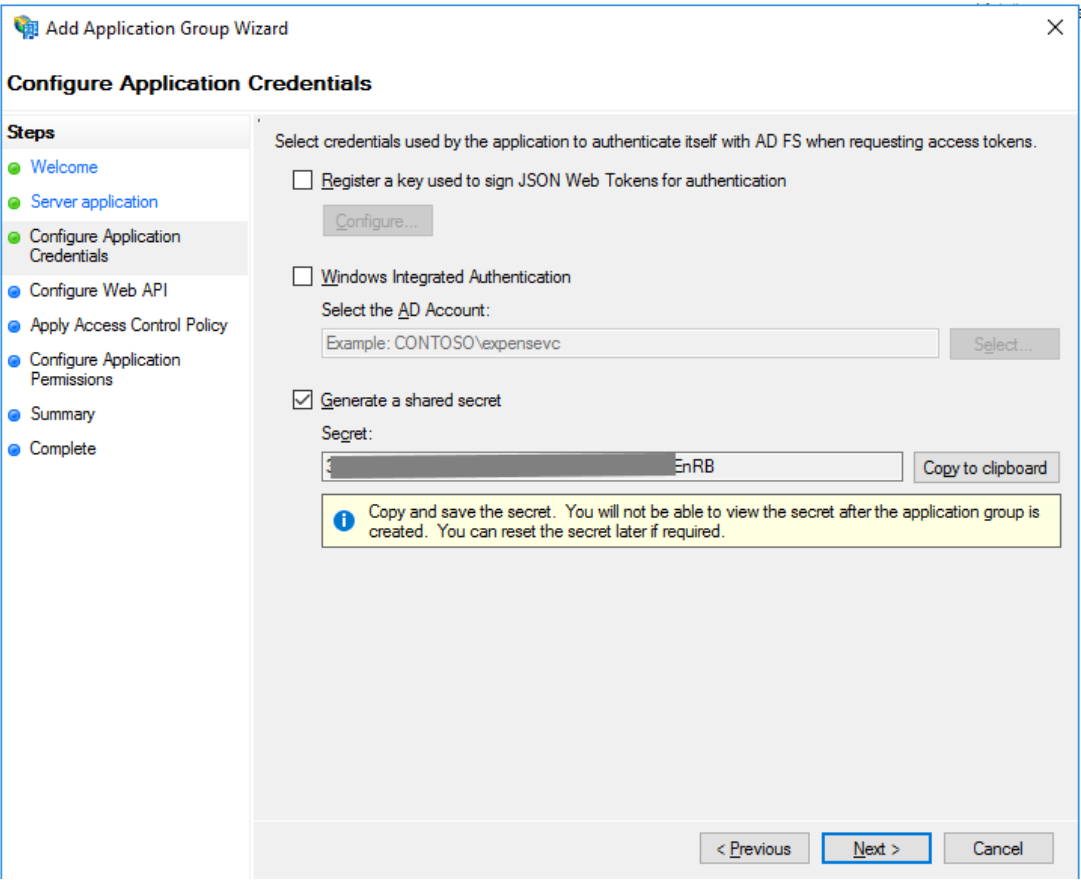


- Configure the client identifier and the redirect (callback) Url.
- Client identifier should be an global unique identifier. This value must be set in the openid-client-id configuration item in FintechOS.
- Redirect (callback) Url must be also be set in the openid-callback-url

configuration item in FintechOS.



5. Configure the shared secret. The shared secret must be set also in the openid-client-secret configuration item in FintechOS.



6. Configure the Web API identifier

**Add Application Group Wizard**

**Configure Application Permissions**

Configure permissions to enable client applications to access this Web API.

Client application (caller):

| Name                           | Description |
|--------------------------------|-------------|
| My Test 2 - Server application |             |

Add... Remove

Permitted scopes:

| Scope Name                                      | Description  |
|---|--|
| <input checked="" type="checkbox"/> allatclaims | Requests the access token claims in the identity token.        |
| <input type="checkbox"/> aza                    | Scope allows broker client to request primary refresh token.   |
| <input checked="" type="checkbox"/> email       | Request the email claim for the signed in user.                |
| <input type="checkbox"/> logon_cert             | The logon_cert scope allows an application to request logo...  |
| <input checked="" type="checkbox"/> openid      | Request use of the OpenID Connect authorization protocol.      |
| <input checked="" type="checkbox"/> profile     | Request profile related claims for the signed in user.         |
| <input type="checkbox"/> user_imperso...        | Request permission for the application to access the resour... |
| <input type="checkbox"/> von_cert               | The von_cert scope allows an application to request VPN ...    |

New scope...

< Previous Next > Cancel

**IMPORTANT!**

The Web API identifier must be THE SAME identifier as the one used for the CLIENT IDENTIFIER in the first step.

7. Configure Access Control Policy.

Add Application Group Wizard

Choose Access Control Policy

Steps

Welcome

Server application

Configure Application Credentials

Configure Web API

Apply Access Control Policy

Configure Application Permissions

Summary

Complete

Choose an access control policy:

| Name   | Description                                      |
|--|--|
| Permit everyone  | Grant access to everyone.                        |
| Permit everyone and require MFA                          | Grant access to everyone and require MFA f...    |
| Permit everyone and require MFA for specific group       | Grant access to everyone and require MFA f...    |
| Permit everyone and require MFA from extranet access     | Grant access to the intranet users and requir... |
| Permit everyone and require MFA from unauthenticated ... | Grant access to everyone and require MFA f...    |
| Permit everyone and require MFA, allow automatic devi... | Grant access to everyone and require MFA f...    |
| Permit everyone for intranet access                      | Grant access to the intranet users.              |
| Permit specific group                                    | Grant access to users of one or more specifi...  |

Policy

Permit everyone

☐ I do not want to configure the access control policy at this time. No users will be permitted access for this application.

< Previous

Next >

Cancel

8. Configure claims to be sent with the openid token.

9. Following claims must be included: allatclaims, email, openid, profile.

**Add Application Group Wizard**

**Configure Application Permissions**

Configure permissions to enable client applications to access this Web API.

Client application (caller):

| Name                           | Description |
|--------------------------------|-------------|
| My Test 2 - Server application |             |

Add... Remove

Permitted scopes:

| Scope Name                                      | Description  |
|---|--|
| <input checked="" type="checkbox"/> allatclaims | Requests the access token claims in the identity token.        |
| <input type="checkbox"/> aza                    | Scope allows broker client to request primary refresh token.   |
| <input checked="" type="checkbox"/> email       | Request the email claim for the signed in user.                |
| <input type="checkbox"/> logon_cert             | The logon_cert scope allows an application to request logo...  |
| <input checked="" type="checkbox"/> openid      | Request use of the OpenID Connect authorization protocol.      |
| <input checked="" type="checkbox"/> profile     | Request profile related claims for the signed in user.         |
| <input type="checkbox"/> user_imperso...        | Request permission for the application to access the resour... |
| <input type="checkbox"/> von_cert               | The von_cert scope allows an application to request VPN ...    |

New scope...

< Previous Next > Cancel

10. Review the configuration in the Summary step and go to Complete step.

### IMPORTANT!

In the following steps we need to expose the GROUP INFORMATION, EMAIL, GIVEN NAME and SURNAME information from AD directory to be included in the claims. This will permit the correct mapping of the users to FintechOS.

- 11. Double click the newly created Application Group.

My Test Properties

General

Name:

My Test

Description:

Applications:

| Name                         | Description |
|------------------------------|-------------|
| Server application           |             |
| My Test - Server application |             |
| Web API                      |             |
| My Test - Web API            |             |

Add application...

Edit...

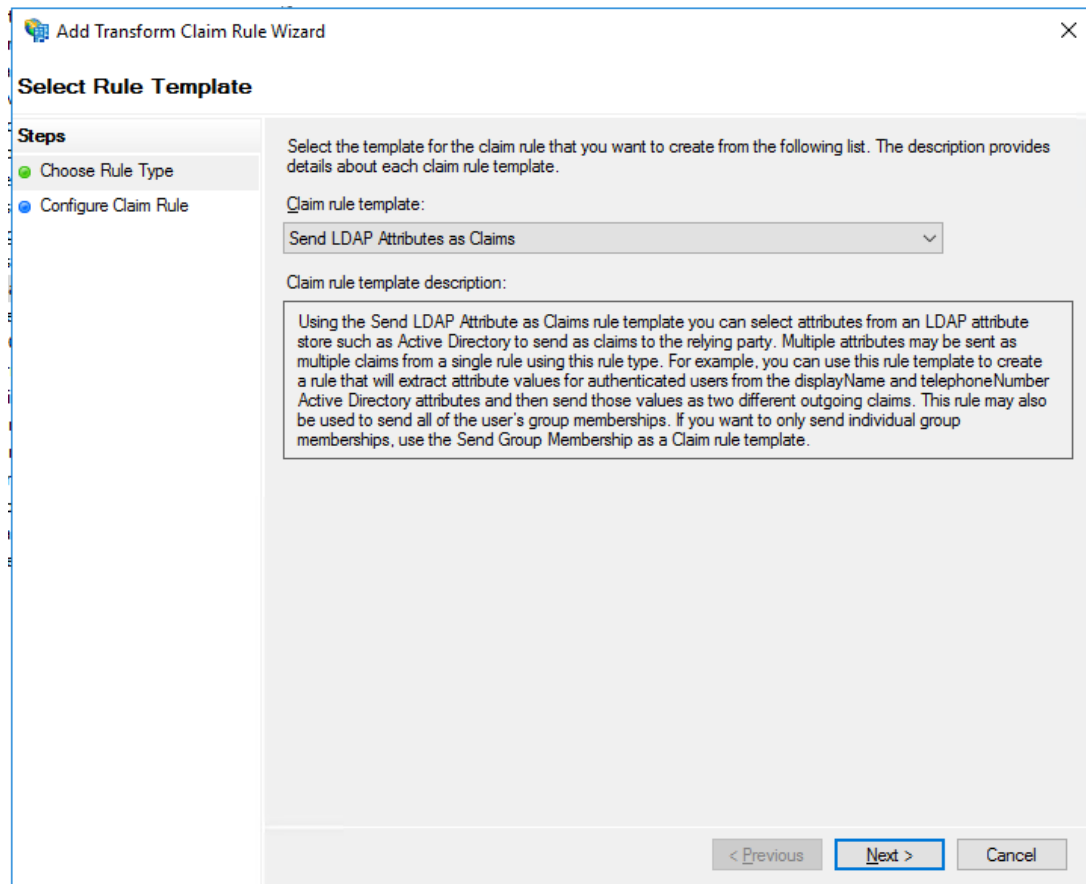
Remove

OK

Cancel

Apply

12. Select the Web API element and click the Edit... button.



13. Go to tab Issuance Transform Rules and add a new rule of type Send LDAP Attributes in Claims

14. Map the AD attributes as in the image below:

Edit Rule - Map LDAP attributes

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name:

Map LDAP attributes

Rule template: Send LDAP Attributes as Claims

Attribute store:

Active Directory

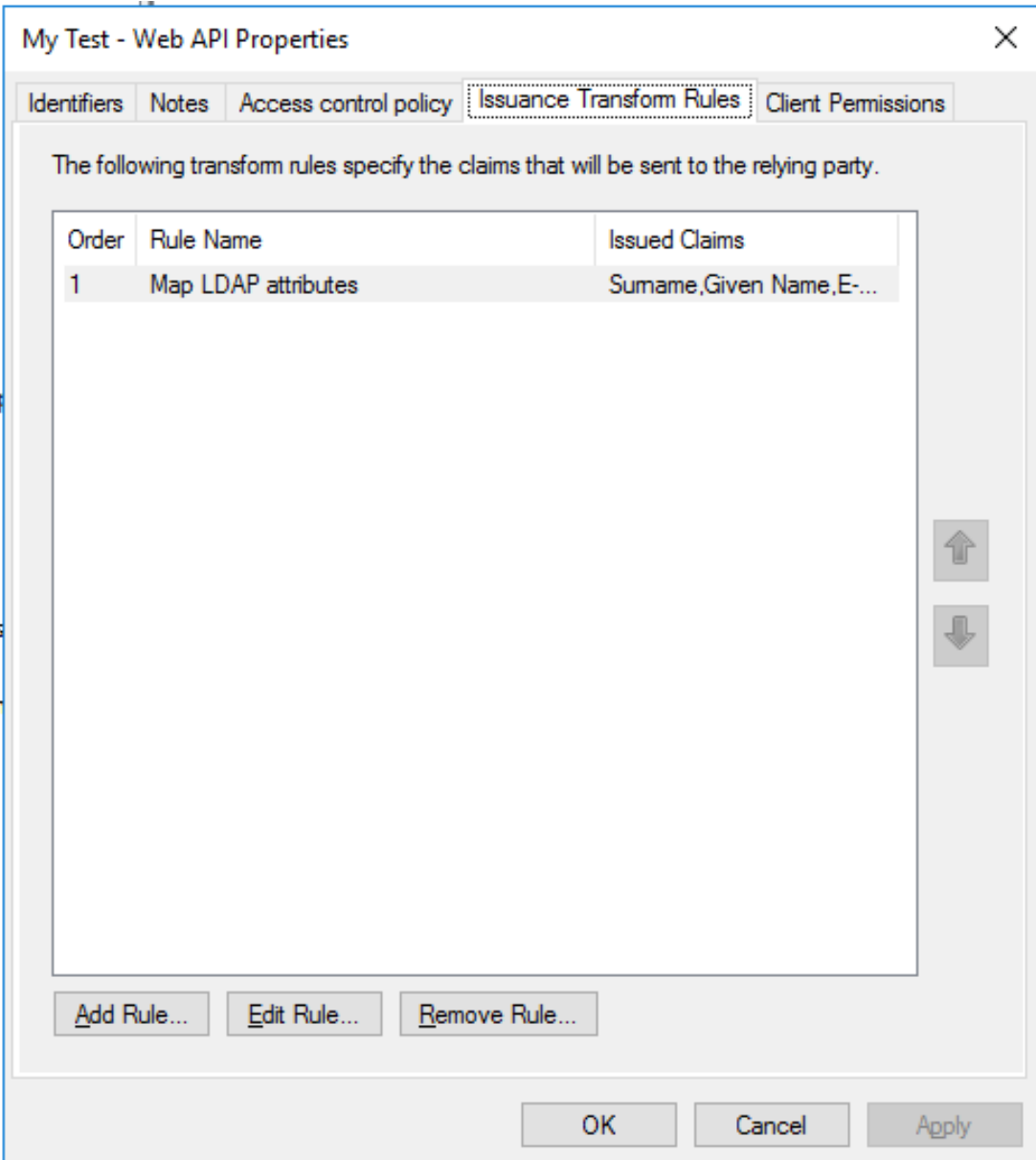
Mapping of LDAP attributes to outgoing claim types:

|   | LDAP Attribute (Select or type to add more) | Outgoing Claim Type (Select or type to add more) |
|---|---|--|
| ▶ | Surname                                     | Surname  |
|   | Given-Name                                  | Given Name                                       |
|   | E-Mail-Addresses                            | E-Mail Address                                   |
|   | Token-Groups - Unqualified Names            | Group  |
| * |   |  |

View Rule Language...

OK

Cancel



**Group mapping in FintechOS**

Once the system user has been created in the Innovation Studio, it is possible to have default roles for this user, organization, business unit and user type configured in web.config.

To configure the mappings, an XML file named `OpenIdUserConfiguration.xml` must be placed in the root of the web application. When the ADFS configuration was performed as in the section above, the ADFS token for an user authentication will include a group claim with the names of the Groups where the user is member from AD.

## Authentication with AWS Cognito

### IMPORTANT!

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services. You can configure the FintechOS Identity Provider to act as an identity broker, allowing users to log in to FintechOS applications and services using their existing AWS Cognito credentials. For more information, see ["Using AWS Cognito as External Identity Provider" on page 178](#).

This authentication method is provided only for backward compatibility.

This service provided by Amazon Web Services manages the user sign-in information for members of a platform. If your organization is using AWS Cognito for identity and access management of your users, it is possible to map the users already existing in AWS Cognito to FintechOS [Security Roles](#). Through Azure AWS OpenId provider, users to log in to FintechOS using their existing AWS Cognito credentials.

### Add keys to Vault secrets

Set AWS Cognito authentication:

| Key Path                                    | Key Name                 | Key Value  |
|---|--------------------------|------------|
| kv/<environment>/<application>/app-settings | EBSDefaultAuthentication | AWSCognito |

AWS Cognito configuration:

| Key Path                                    | Key Name             | Key Value                        |
|---|----------------------|----------------------------------|
| kv/<environment>/<application>/app-settings | openid-client-id     | AWS Cognito client id xxxxx      |
| kv/<environment>/<application>/app-settings | openid-client-secret | AWS Cognito client secret yyyyyy |

| Key Path                                    | Key Name                  | Key Value  |
|---|---------------------------|--|
| kv/<environment>/<application>/app-settings | openid-callback-url       | http://\${portalRoot}/Account/LogonCallback              |
| kv/<environment>/<application>/app-settings | openid-discovery-endpoint | https://cognito-idp.xxx/.well-known/openid-configuration |

## User mapping settings:

| Key Path                                    | Key Name                           | Key Value                        |
|---|------------------------------------|----------------------------------|
| kv/<environment>/<application>/app-settings | openid-auto-user-roles             | Guest,Developer,Registered Users |
| kv/<environment>/<application>/app-settings | openid-auto-user-organization      | ebs                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-businessunit      | root                             |
| kv/<environment>/<application>/app-settings | openid-auto-user-type              | Back Office                      |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-add  | 0 1                              |
| kv/<environment>/<application>/app-settings | openid-auto-user-remote-roles-sync | 0 1                              |

## Configuration Keys:

| Key                               | Value   |
|-----------------------------------|---|
| openid-auto-user-roles            | Platform role names, separated by colon. These roles will be added automatically when the AWS Cognito user is mapped to a platform user |
| openid-auto-user-organization     | Platform organization name. The mapped user will be added in this organization  |
| openid-auto-user-businessunit     | Platform business unit name. The mapped user will be added in this business unit  |
| openid-auto-user-remote-roles-add | not supported yet   |

| Key                                | Value             |
|------------------------------------|-------------------|
| openid-auto-user-remote-roles-sync | not supported yet |

**Parameters:**

| Parameter      | Value                         |
|----------------|-------------------------------|
| \${portalRoot} | root url for FintechOS portal |

## (Deprecated) Add keys to the web.config file

In the web.config file of your environment add the following keys.

```
<add key="EBSDefaultAuthentication" value="AWSCognito" />

<!-- BEGIN AWS COGNITO IDOPEN ID CONFIGURATION -->

    <add key="openid-client-id" value="AWS Cognito client id
xxxxx" />
    <add key="openid-client-secret" value="AWS Cognito
client secret yyyyyy" />

    <add key="openid-callback-
url" value="http://${portalRoot}/Account/LogonCallback" />

    <add key="openid-discovery-
endpoint" value="https://cognito-idp.xxx/.well-known/openid-
configuration" />

<!-- USER MAPPING SETTINGS -->

    <add key="openid-auto-user-
roles" value="Guest,Developer,Registered Users" />
    <add key="openid-auto-user-organization" value="ebs" />
    <add key="openid-auto-user-businessunit" value="root" />
    <add key="openid-auto-user-type" value="Back Office" />

    <add key="openid-auto-user-remote-roles-add" value="0"/>
    <add key="openid-auto-user-remote-roles-
sync" value="0"/>

<!-- END AWS COGNITO ID CONFIGURATION -->
```

### Group mapping for users

For each user in FintechOS, default roles can be created in the web.config file for this user, organization, business unit and user type.

1. An XML file named **OpenIdUserConfiguration.xml** must be placed in the root of the web application of FintechOS.

#### IMPORTANT!

Any changes to OpenIdUserConfiguration.xml require a manual Application Domain restart.

2. The ADFS token for an user authentication will include a group claim with the names of the Groups where the user is member from AD.

### Example:

```
<root>
  <SecurityGroup>
    <Name>GROUP1</Name>

    <DefaultBusinessUnitName>root</DefaultBusinessUnitName>
    <SecurityRoleName>Registered
Users,Developers</SecurityRoleName>
  </SecurityGroup>
  ...

  <SecurityGroup>
    <Name>GROUP2</Name>

    <DefaultBusinessUnitName>root</DefaultBusinessUnitName>
    <SecurityRoleName>GROUP2</SecurityRoleName>
  </SecurityGroup>
</root>
```

## Browser Based Multi-Factor Authentication

Multi-Factor Authentication (MFA) adds an extra layer of security on top of the basic authentication methods. It requires users to provide multiple proof of their claimed identity prior to being granted access according to their security roles and permissions.

User access can be granted based on two cumulative conditions:

- Something the user knows (login credentials): username and password.
- Something the user controls (a mobile device or email account used to receive a temporary pass code via SMS/E-mail).

When users access the app, they will be prompted to provide the login credentials associated with their HPFI account. To make sure account access is protected, after the login credentials are provided, a one-time security pass code is sent to the user's phone (the phone number set in the user account profile) or email address. Once the user enters the code received via the SMS/e-mail, access into the system is granted.

The following authenticator apps have been tested and confirmed as working:

- Microsoft Authenticator
- Google Authenticator
- FreeOTP

Follow the instructions below to set up multi-factor authentication.

### 1 Create a Browser Authentication Flow

1. Log in to the FintechOS Identity Provider admin console.
2. Select your HPFI realm.
3. Select the **Authentication** blade.

4. In the **Flows** tab, select the **Browser** based authentication flow and click the **Copy** button to create a duplicate.

**Authentication**

Flows Bindings Required Actions Password Policy OTP Policy WebAuthn Policy<sup>®</sup> WebAuthn Passwordless Policy<sup>®</sup> CIBA Policy

**Browser** New **Copy**

| Auth Type                    | Requirement  |                         |
|------------------------------|--|-------------------------|
| Cookie                       | <input type="radio"/> REQUIRED <input checked="" type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED                                   |                         |
| Kerberos                     | <input type="radio"/> REQUIRED <input type="radio"/> ALTERNATIVE <input checked="" type="radio"/> DISABLED                                   |                         |
| Identity Provider Redirector | <input type="radio"/> REQUIRED <input checked="" type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED                                   | <a href="#">Actions</a> |
| Forms ⓘ                      | <input type="radio"/> REQUIRED <input checked="" type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED <input type="radio"/> CONDITIONAL |                         |
| Username Password Form       | <input checked="" type="radio"/> REQUIRED  |                         |
| Browser - Conditional OTP ⓘ  | <input type="radio"/> REQUIRED <input type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED <input checked="" type="radio"/> CONDITIONAL |                         |
| Condition - User Configured  | <input checked="" type="radio"/> REQUIRED <input type="radio"/> DISABLED   |                         |
| OTP Form                     | <input checked="" type="radio"/> REQUIRED <input type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED                                   |                         |

5. In the window that opens, assign a name for your new browser authentication flow.

**Copy Authentication Flow** ×

**New Name**

Ftosbrowserflow

Cancel Ok

- Open the newly created authentication flow and, in the **Forms** authentication type, set the **Browser - Conditional OTP** subflow to **Required**.

**Authentication**

Flows Bindings Required Actions Password Policy OTP Policy WebAuthn Policy<sup>®</sup> WebAuthn Passwordless Policy<sup>®</sup> CIBA Policy

Ftosbrowserflow

New Copy Delete Edit Flow Add execution Add flow

| Auth Type                                 | Requirement                               | Actions |
|---|---|---------|
| Cookie                                    | REQUIRED ALTERNATIVE DISABLED             | Actions |
| Kerberos                                  | REQUIRED ALTERNATIVE DISABLED             | Actions |
| Identity Provider Redirector              | REQUIRED ALTERNATIVE DISABLED             | Actions |
| Ftosbrowserflow Forms                     | REQUIRED ALTERNATIVE DISABLED CONDITIONAL | Actions |
| Username Password Form                    | REQUIRED                                  | Actions |
| Ftosbrowserflow Browser - Conditional OTP | REQUIRED ALTERNATIVE DISABLED CONDITIONAL | Actions |
| Condition - Scope                         | REQUIRED DISABLED                         | Actions |
| SMS Authentication                        | REQUIRED ALTERNATIVE DISABLED CONDITIONAL | Actions |

**(Optional) Enable Conditional OTP only for specific roles**

- Open your authentication flow and, in the **Forms** authentication type, in the **Browser - Conditional OTP** subflow, click the **Actions** drop-down and select **Add execution**.
- In the Create Authentication Execution screen, select the **Condition - User Role** provider and click **Save**.

**Create Authenticator Execution**

Flows Bindings Required Actions Password Policy OTP Policy WebAuthn Policy<sup>®</sup> WebAuthn Passwordless Policy<sup>®</sup> CIBA Policy

Provider<sup>®</sup>


Condition - User Role


Save Cancel


3. In the authentication flow window, move the **Condition - User Role** execution above the **OTP Form** execution and, from its **Actions** drop down, select **Config**.
4. In the user authenticator configuration page, add an **Alias** of your choice and select the **Role** which will require multi-factor authentication. If you wish to apply multi-factor authentication to all user roles except the provided role, tick the **Negate output** option.
5. On your authentication flow page, in the **Forms** authentication type, set the **Browser - Conditional OTP** subflow to **Conditional**.


## 2 Associate the Authentication Flow to a Client


1. In the **Clients** blade, open the FintechOS Identity Provider client you wish to enable multi-factor authentication for, such as a Portal or Innovation Studio client.
2. In the **Settings** tab, scroll down to the **Authentication Flow Overrides** and expand it.
3. Set the **Browser Flow** option to the flow you created earlier and click **Save**.


Backchannel Logout Session Required 


Backchannel Logout Revoke Offline Sessions 

> Fine Grain OpenID Connect Configuration 


> OpenID Connect Compatibility Modes 

> Advanced Settings 

▼ Authentication Flow Overrides 

Browser Flow 

ftosbrowserflow ▼

Direct Grant Flow 

▼

**Save** **Cancel**

## Authenticator Reset

Once set up, the multi-factor authentication will be required when a user logs in for the first time. If a user loses access to the authenticator app or needs to reconfigure for various reasons, a FintechOS Identity Provider administrator must log in to the

administration console and either:

- Delete the created authenticator

or

- From the user's **Credentials** tab, set up a **Configure OTP** reset action.

The screenshot shows the FintechOS Identity Provider administration console. The left sidebar contains navigation links for 'Configure', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main content area is titled 'Host' and has tabs for 'Details', 'Attributes', 'Credentials', 'Role Mappings', 'Groups', 'Consents', 'Sessions', and 'Identity Provider Links'. The 'Credentials' tab is selected. Below the tabs is a 'Manage Credentials' section with a table showing a single credential of type 'password'. Below this is a 'Reset Password' section with fields for 'Password', 'Password Confirmation', and a 'Temporary' toggle. Below that is a 'Credential Reset' section with a 'Reset Actions' dropdown set to 'Configure OTP (CONFIGURE\_TOTP)', an 'Expires in' field set to '12 Hours', and a 'Reset Actions Email' field set to 'Send email'.

## Email/SMS/IVR Multi-Factor Authentication

Email/SMS/IVR Multi-Factor Authentication (MFA) adds an extra layer of security on top of the basic authentication methods. When users access the FintechOS Portal or Innovation Studio, they will be prompted to provide the login credentials associated with their HPFI account. To make sure account access is protected, after the login credentials are provided, users are redirected to a secondary login page where they have to enter a one-time security pass code received via email, SMS or IVR (the phone number set in the user account profile). Once a user enters the code received via the Email/SMS/IVR option, access to the system is granted.

Follow the instructions below to set up Email/SMS/IVR(Call Me) multi-factor authentication.

## 1 Set Up Your Email/SMS/IVR Service Providers

In the "Configuration Manager" on page 86, create a custom secret that holds settings for the Email/SMS/Call Me service providers available to deliver the one-time passwords, e.g.: `myEnvironment/service-pipes/myProviders`. The secret's value must have the following structure:

```
{
  "provider-configurations": [
    {
      "url": "https://api.emailprovider.com",
      "payload-template": "{\"sender\":{\"name\":\"\", \"email\":\"SENDER_
PLACEHOLDER\"}, \"to\": [ { \"email\":\"RECEIVER_
PLACEHOLDER\", \"name\":\"\"}], \"subject\":\"SUBJECT_
PLACEHOLDER\", \"body\":\"MESSAGE_PLACEHOLDER\"}\",
      "default-sender": "pwdGenie@myCompany.com",
      "communication-channel": "email",
      "headers": {
        "Accept": "application/json",
        "Api-Key": "abcdef",
        "Content-Type": "application/json"
      },
      "placeholder-mappers": {
        "sender": "SENDER_PLACEHOLDER",
        "receiver": "RECEIVER_PLACEHOLDER",
        "subject": "SUBJECT_PLACEHOLDER",
        "message": "MESSAGE_PLACEHOLDER"
      },
      "phone-number-prefix-to-add" : "",
      "phone-number-prefix-to-remove" : ""
    },
    {
      "url": "https://api.smsprovider.com",
      "payload-template": "{\"text\":\"MESSAGE_PLACEHOLDER\", \"to\":\"RECEIVER_
PLACEHOLDER\", \"from\":\"SENDER_PLACEHOLDER\"}\",
      "default-sender": "anotherPwdGenie",
      "communication-channel": "sms",
      "headers": {
        "Content-Type": "application/json"
      },
      "placeholder-mappers": {
        "sender": "SENDER_PLACEHOLDER",
```

```

"receiver": "RECEIVER_PLACEHOLDER",
"message": "MESSAGE_PLACEHOLDER"
},
"phone-number-prefix-to-add" : "",
"phone-number-prefix-to-remove" : ""
},
{
"url": "https://api.callme.com",
"payload-template": "{\"code\":\"CODE_PLACEHOLDER\",\"to\":\"RECEIVER_
PLACEHOLDER\"}",
"default-sender": "",
"communication-channel": "call-me",
"headers": {
"Content-Type": "application/json",
"Key": "123456"
},
"placeholder-mappers": {
"receiver": "RECEIVER_PLACEHOLDER",
"code": "CODE_PLACEHOLDER"
},
"phone-number-prefix-to-add" : "",
"phone-number-prefix-to-remove" : ""
}
]
}

```

| Property              | Description   |
|-----------------------|---|
| url                   | The service provider's URL that will receive the service request.   |
| payload-template      | The template for the request body. It can contain placeholders that will be later replaced with predefined information.   |
| default-sender        | If the sender is not provided as an input parameter, this field will be used as the sender of the message. Use only if the service provider supports specifying a sender. |
| communication-channel | Specifies the communication channel as either <b>Email</b> , <b>SMS</b> , or <b>Call Me</b> .   |
| headers               | List of key-value pairs representing headers that will be added to the request. The headers' keys and values can contain placeholders.                                    |

| Property                  | Description   |
|---------------------------|---|
| placeholder-mappers       | List of key-value pairs specifying what information should fill the placeholders. All the keys should belong to the following list: sender, receiver, subject, message, code.<br>E.g.: If the payload-template is the following string: "address: PLACEHOLDER_RECEIVER" and the intent is to replace the PLACEHOLDER-RECEIVER string with the email address of the client, then the value of the placeholder-mappers configuration property should be: {"receiver":"PLACEHOLDER_RECEIVER"}. |
| phoneNumberPrefixToAdd    | A string to be prepended to the client phone number before sending the request to the service provider.<br>E.g.: In the FintechOS Identity Provider, the user's phone number is stored in the following format: 40XXXXXXXXXX, but the SMS service provider needs the phone number in the following format: +40XXXXXXXXXX. To fix this, set the phoneNumberPrefixToAdd to +.   |
| phoneNumberPrefixToRemove | A string to be removed from the beginning of the phone number before sending the request to the service provider.<br>E.g.: In the FintechOS Identity Provider, the user's phone number is stored in the following format: +40XXXXXXXXXX, but the SMS service provider needs the phone number in the following format: 40XXXXXXXXXX. To fix this, set the phoneNumberPrefixToRemove to +.  |

## 2 Configure the Service Pipes to Use the Defined Service Providers

In your Azure admin console, open the Service Pipes web app and set the `app.vault.custom.secrets.path` property to the Configuration Manager secret path you created earlier, e.g.: `myEnvironment/service-pipes/myProviders`.

## 3 Create an Authentication Flow

1. Log in to the FintechOS Identity Provider admin console.
2. Select your HPFI realm.
3. Select the **Authentication** blade.
4. In the **Flows** tab, create a new authentication flow based on your preferences.

5. In the **Forms** authentication type, set the flow's last execution step to use the **SMS Authentication - Service Pipes** provider.

The screenshot displays the HPFI administration interface. At the top, there's a navigation bar with buttons: New, Copy, Delete, Edit Flow, Add execution, and Add flow. Below this, a table lists various authentication types and their requirements. The 'Forms' type is selected, and its configuration is shown below. A red box highlights the 'SMS Authentication - Service Pipes' option in the 'Forms' type configuration. Below this, a dropdown menu is open, showing a list of providers. The 'SMS Authentication - Service Pipes' provider is highlighted in blue.

| Auth Type                          | Requirement  | Actions |
|------------------------------------|--|---------|
| Cookie                             | REQUIRED <input type="radio"/> ALTERNATIVE <input checked="" type="radio"/> DISABLED <input type="radio"/>                                   | Actions |
| Kerberos                           | REQUIRED <input type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED <input checked="" type="radio"/>                                   | Actions |
| Identity Provider Redirector       | REQUIRED <input type="radio"/> ALTERNATIVE <input checked="" type="radio"/> DISABLED <input type="radio"/>                                   | Actions |
| Forms                              | REQUIRED <input type="radio"/> ALTERNATIVE <input checked="" type="radio"/> DISABLED <input type="radio"/> CONDITIONAL <input type="radio"/> | Actions |
| Username Password Form             | REQUIRED <input checked="" type="radio"/>  | Actions |
| Browser - Conditional OTP          | REQUIRED <input checked="" type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED <input type="radio"/> CONDITIONAL <input type="radio"/> | Actions |
| Condition - User                   | REQUIRED <input checked="" type="radio"/> DISABLED <input type="radio"/>   | Actions |
| SMS Authentication - Service Pipes | REQUIRED <input checked="" type="radio"/> ALTERNATIVE <input type="radio"/> DISABLED <input type="radio"/> CONDITIONAL <input type="radio"/> | Actions |

Provider

- Browser Redirect For Cookie Free Authentication
- Password Form
- Recaptcha Username Password Form
- Reset OTP
- Reset Password
- Review Profile
- Send Reset Email
- SMS Authentication
- SMS Authentication - Service Pipes**
- Update Profile Attribute
- User Session Count Limiter
- Username Form
- Username Password Challenge
- Username Password Form
- Username Password Form For Identity Provider Reauthentication
- Username Validation
- Verify Existing Account By Email
- WebAuthn Authenticator
- WebAuthn Passwordless Authenticator
- X509/Validate Username
- X509/Validate Username Form

## 4 Configure the Flow's MFA Execution Step

1. Open your authentication flow and navigate to the **SMS Authentication - Service Pipes** execution step.
2. Click the **Actions** drop-down and select **Config**.

3. Set up the following parameters:

- **Code length:** The length of the one-time password that is going to be sent to the user.
- **Time-to-live:** The duration in seconds the one-time password is valid.
- **Mobile number attribute name:** The attribute name that is going to be used to retrieve the user's phone number.
- **Subject template:** The message subject template that is going to be sent to the user when using the email communication channel. You can use the following placeholders: `${firstName}`, `${lastName}`, `${content}`.
- **Message template:** The message content template that is going to be sent to the user via SMS or email. You can use the following placeholders: `${firstName}`, `${lastName}`, `${content}`.
- **Notification Service Url:** URL to the service pipes web app which receives the request and forwards it to the chosen service provider.
- **Client Id:** The client ID used to authenticate the request to the service pipes web app.
- **Client Secret:** The client secret used to authenticate the request to the service pipes web app.
- **Simulation mode:** When set to True, the actual one-time password will not be sent to the user. Instead, the one-time password will be logged on the FintechOS Identity Provider server.
- **Allow Email channel:** If set to true, the user will be able to select Email as communication channel for receiving the OTP.
- **Allow SMS channel:** If set to true, the user will be able to select SMS as communication channel for receiving the OTP.

- **Allow Call Me channel:** If set to true, the user will be able to select Call Me as communication channel for receiving the OTP.

OTP Config

ID  
f64d885d-4dac-4a98-b98d-983a64f3d847

Alias ⓘ  
OTP Config

Code length ⓘ  
6

Time-to-live ⓘ  
300

Mobile number attribute name ⓘ  
phoneNumber

Subject template ⓘ  
Hello \${firstName}, You received a new OTP.

Message template ⓘ  
Hello \${firstName}, \${lastName} your OTP is : \${content}

Notification service Uri ⓘ

IDP Client Id ⓘ

IDPClient Secret ⓘ

Simulation mode ⓘ  
☒

Allow Email Channel ⓘ  
☒

Allow SMS Channel ⓘ  
☒

Allow Call Me Channel ⓘ  
☒

Save Cancel

4. Click **Save**.

## 5 Activate the Authentication Flow

Once the authentication flow is configured, in the FintechOS Identity Provider, navigate to **Authentication > Bindings** and replace the Browser flow with the newly

created flow. This will set the authentication flow globally. However, if you want to enable the flow only for one client, you can navigate to the client page, go to **Settings > Authentication Flow Overrides**, and change the **Browser Flow** there.

A user who logs in to FintechOS Portal or Innovation Studio, will be directed to the one-time password form and will receive the required password via Email, SMS or IVR (Call Me).

## Deprecated Multi-Factor Authentication

### IMPORTANT!

Starting with release 22.1, the FintechOS HPFI uses the FintechOS Identity Provider as the default authentication layer for the FintechOS applications and services.

Alternate identity providers and the corresponding multi-factor authentications are supported only for backward compatibility.

For information about the FintechOS Identity Provider multi-factor authentication, see ["Browser Based Multi-Factor Authentication" on page 224](#).

Multi-Factor Authentication (MFA) adds an extra layer of security on top of the basic authentication methods. It requires users to provide multiple proof of their claimed identity prior to being granted access to resources based on business need to know according to their security role and granted permissions.

User access can be granted on two-factors:

- Something the user knows (login credentials): username and password.
- Something the user has (pass code received via an SMS/E-mail or mobile soft token).

When users access the app, they will be prompted to provide the login credentials associated with their HPFI account. To make sure account access is protected, after the login credentials are provided, a one-time security pass code is sent to the user's phone (the phone number set in the user account profile) or email. Once the user enters the code received via the SMS/e-mail, access into the system is granted.

---

## SMS-based Two-Factor Authentication

SMS-based two-factor authentication is the most popular choice when it comes to multi-factor authentication as most users have their own mobile phones and have them handy when logging into apps.

### How it works?

Users will be granted access to the HPFI app following these two steps:

- Users will navigate to the HPFI web app and they will provide their account credentials (username and password). Based on the app configuration, the credentials can be either local or from external providers. To make sure account access is protected, after the login credentials are provided, a one-time security pass code is sent to the user's phone (the phone number set in the user account profile).
- In the web app, they provide the pass code received via SMS. Access into the system is granted.

### How to set up the SMS-based MFA?

Setting up the SMS-based multi-factor authentication is a two-step process:

#### **1** Enable Multi-Factor Authentication

On the server where the HPFI installation package resides, go to the **web.config** file and add the following settings:

- To the **<configSections>** tag, add the **multifactorAuthentication** section:

```
<configuration>
  <configSections>
    ...
    <section name="multifactorAuthentication" type=
"EBS.Core.Authentication.Common.MultifactorAuthentication.Co
nfig.MultifactorAuthenticationSection,
EBS.Core.Authentication.Common" />
  </configSections>
</configuration>
```

- Add the **<multifactorAuthentication>** tag:

```
<configuration>
...
  <multifactorAuthentication
xmlns
=
"http://fintechos.com/ebs/schemas/multifactorAuthentication"
enabled="true">
    <providers>
      <provider
name="SMS" enabled="true" default="true">
        <type fullName=
"EBS.Core.Authorization.Services.Security.SmsMultiFactorAuth
enticationProvider, EBS.Core.Authorization.Services" />
        <properties>
          <property
name="ChannelProvider" value="GatewaySmsOTP" />
          <property
name="CommunicationChannel" value="Sms" />
          <property
name="MaxNumberOfAuthenticationRetries" value="3" />
          <property
name="MaxNumberOfSmsSendings" value="3" /> </properties>
        </provider>
      </providers>
      <runtime>
        <providers>
          <provider name="SMS">
            <roles>
              <role name="*" /> </roles>
            </provider>
          </providers>
        </runtime>
```

```
</multiFactorAuthentication>
</configuration>
```

Where:

- `multiFactorAuthentication/@enabled` - controls if MFA is enabled or not.  
Default value: false;
- `multiFactorAuthentication/providers/provider/@enabled` - controls if a specific MFA provider is enabled or not. Default value: true;
- `multiFactorAuthentication/providers/provider/@default`
  - When MFA is enabled, there can be at most one provider marked with `default="true"`.
  - The provider marked with `default="true"` will be selected for MFA when there are multiple providers available for user's roles and the user hasn't selected any preferred communication channel or his preferred communication channel is not present into the configured providers list.
  - Default value: false.
- on `multiFactorAuthentication/providers/provider/properties`:
  - `ChannelProvider` - the channel provider used by the SMS MFA provider to send text messages. Its value must be the Name of one of the records from **EbsMetadata.FTOS\_DPA\_ChannelProvider** table;
  - `CommunicationChannel` - the communication channel used by the SMS provider to send text messages. Its value must be the Name of one of the records from **EbsMetadata.FTOS\_DPA\_CommunicationChannel** table;
  - `MaxNumberOfAuthenticationRetries` - the user has up to `MaxNumberOfAuthenticationRetries` chances to enter the correct code. If this threshold is reached the user will be redirected to the login page. Default value: **3**;

- `MaxNumberOfSmsSendings` - if needed the user may request a resending of the code for up to `MaxNumberOfSmsSendings` times. If this threshold is reached the user will be redirected to the login page. Default value: **3**;
- `multiFactorAuthentication/runtime/providers/provider`  
`[@name='SMS']/roles` will include a `<role name=""/>` child for each role that contains users that have to be authenticated through this provider. Note that a `<role name="*" />` means that all roles will be taken into account;

If the multi-factor authentication is activated, at the next profile change, users will have to provide their phone number (in the Edit System User, My Account page, the Phone Number field is mandatory).

Once you've activated the SMS-based authentication, you need to configure the Job Server for Multi-Factor Authentication.

## 2 Configure the Job Server for MFA

### IMPORTANT!

The MessageBus (OCS) plugin for the FintechOS Job Server already includes the configurations required for multi-factor authentication (see the *FintechOS Installation Guide* for details about MessageBus (OCS) installation).

- If you have the MessageBus (OCS) plugin installed, skip this step.
- If you are using the standard Job Server configuration, follow the instructions below to configure the multi-factor authentication settings.

1. On the server where the HPFI installation package resides, go to the **schedule.config** file and add the following section:

```
<triggers>
...
  <trigger>
    <name>FTOS.OCB.OTP</name>
    <startTime>02.11.2017 11:00</startTime>
    <endTime>03.11.2080 11:02</endTime>
```

```

        <repeatCount>-1</repeatCount>
        <rescheduleAfterRun>false</rescheduleAfterRun>
        <async>false</async>
        <expression>0/10 * * * * ?</expression>
        <services>

        <service>FTOS.OCB.SendMessagesServiceSmsOTP</service>

        <service>FTOS.OCB.UpdateStatusServiceSmsOTP</service>

        <service>FTOS.OCB.UpdateExpiredMessageServiceSmsOTP</service>
        </services>
    </trigger>
</triggers>

```

2. On the server where the HPFI installation package resides, go to the **services.config** file and add the following sections:

```

<serviceList>
...
    <!--OTP-->
    <service>
        <name>FTOS.OCB.SendMessagesServiceSmsOTP</name>
        <type>class</type>
        <method></method>

    <
class
>
FTOS.MessageBus.ScheduledServices.SendMessagesService</class>

<assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

<execParams>
provider=gateway;providerSetting=gatewaySmsOTP</execParams>
    </service>
    <service>
        <name>FTOS.OCB.UpdateStatusServiceSmsOTP</name>
        <type>class</type>
        <method></method>
        <class>
FTOS.MessageBus.ScheduledServices.UpdateStatusMessagesService
</class>

    <assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

```

```

<execParams>
provider=gateway;providerSetting=gatewaySmsOTP</execParams>
</service>
<service>

<name>FTOS.OCB.UpdateExpiredMessageServiceSmsOTP</name>
<type>class</type>
<method></method>
<class>
FTOS.MessageBus.ScheduledServices.UpdateExpiredMessageService
</class>

<assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

<execParams>
provider=gateway;providerSetting=gatewaySmsOTP</execParams>
</service>
</serviceList>

```

#### Configure Multi Factor Authentication to use an SMS Service provider

In the web.config file, set the ChannelProvider property of the MFA provider with value "FTOSApiSms".

## Example

```

<multiFactorAuthentication xmlns
="http://fintechos.com/ebs/schemas/multiFactorAuthenticatio
n" enabled="true">
  <providers>
    <provider name="SMS" enabled="true">
      <type fullName
="EBS.Core.Authorization.Services.Security.SmsMultiFactorAut
henticationProvider, EBS.Core.Authorization.Services" />
      <properties>
        <property
name="ChannelProvider" value="FTOSApiSms" />
        <property
name="MaxNumberOfAuthenticationRetries" value="3" />
        <property
name="MaxNumberOfSmsSendings" value="3" />
        <property
name="MessageTemplate" value="myMessageTemplate_SmsApi" />
      </properties>
    </provider>
  </providers>
</multiFactorAuthentication>

```

```

        </provider>
    </providers>
    ...
</multiFactorAuthentication>

```

## Password reset SMS for the log-in credentials

To set up password reset confirmation:

Add section

```

<configSections>
...
  <section name="passwordReset"
type="EBS.Core.Web.MVC.PasswordResetConfig, EBS.Core.Web.MVC"/>
</configSections>

```

Add configuration element

```

<configuration>
...
  <passwordReset xmlns="urn:EBS.Core.Web.MVC">
    <confirmation channelProvider="" messageTemplate=""
enabled="true"/>
  </passwordReset>
...
</configuration>

```

where:

- enabled - if true, after the completion of the password reset flow a message will be sent to user's phone number. Default value: false;
- channelProvider - the provider that will be used to send the message. Must be one of "GatewaySmsOTP" or "FtosApiSms";

- `messageTemplate` - the template that will be used to create the message. Must be a record from `FTOS_CMB_ActionTemplate` entity.

If the configuration element is missing the message will not be sent.

## Email-based Two-Factor Authentication

Email-based two-factor authentication is a popular choice when it comes to multi-factor authentication.

### How it works?

Users will be granted access to the HPFI app following these two steps:

- Users will navigate to the HPFI web app and they will provide their account credentials (username and password). Based on the app configuration, the credentials can be either local or from external providers. To make sure account access is protected, after the login credentials are provided, a one-time security pass code is sent to the user's email address.
- In the web app, they provide the pass code received via email. Access into the system is granted.

### How to set up the Email-based MFA?

Setting up the email-based multi-factor authentication is a two-step process:

#### Step 1 Enable Multi-Factor Authentication

On the server where the HPFI installation package resides, go to the **web.config** file and add the following settings:

- To the **<configSections>** tag, add the **multifactorAuthentication** section:

```
<configuration>
  <configSections>
    ...
    <section name="multifactorAuthentication" type=
"EBS.Core.Authentication.Common.MultifactorAuthentication.Co
nfig.MultifactorAuthenticationSection,
EBS.Core.Authentication.Common" /> </configSections>
  </configuration>
```

- Add the **<multifactorAuthentication>** tag:

```
<configuration>
  ...
  <multifactorAuthentication
xmlns
=
"http://fintechos.com/ebs/schemas/multifactorAuthentication"
enabled="true">
    <providers>
      <provider
name="Email" enabled="true" default="true">
        <type fullName=
"EBS.Core.Web.MVC.Security.EmailMultifactorAuthenticationPro
vider, EBS.Core.Web.MVC" />
        <properties>
          <property
name="ChannelProvider" value="GatewayEmailOTP" />
          <property
name="CommunicationChannel" value="Email" />
          <property
name="MaxNumberOfAuthenticationRetries" value="3" />
          <property
name="MaxNumberOfEmailSendings" value="3" /> </properties>
        </provider>
      </providers>
      <runtime>
        <providers>
          <provider name="Email">
            <roles>
              <role name="*" /> </roles>
            </provider>
          </providers>
        </runtime>
      </multifactorAuthentication>
```

```
</configuration>
```

Where:

- `multiFactorAuthentication/@enabled` - controls if MFA is enabled or not.  
Default value: false;
- `multiFactorAuthentication/providers/provider/@enabled` - controls if a specific MFA provider is enabled or not. Default value: true;
- `multiFactorAuthentication/providers/provider/@default`
  - When MFA is enabled, there can be at most one provider marked with `default="true"`.
  - The provider marked with `default="true"` will be selected for MFA when there are multiple providers available for user's roles and the user hasn't selected any preferred communication channel or his preferred communication channel is not present into the configured providers list.
  - Default value: false.
- on `multiFactorAuthentication/providers/provider/properties`:
  - `ChannelProvider` - the channel provider used by the Email MFA provider to send email messages. Its value must be the Name of one of the records from **EbsMetadata.FTOS\_DPA\_ChannelProvider** table;
  - `CommunicationChannel` - the communication channel used by the Email provider to send email messages. Its value must be the Name of one of the records from **EbsMetadata.FTOS\_DPA\_CommunicationChannel** table;
  - `MaxNumberOfAuthenticationRetries` - the user has up to `MaxNumberOfAuthenticationRetries` chances to enter the correct code. If this threshold is reached the user will be redirected to the login page. Default value: **3**;

- `MaxNumberOfEmailSendings` - if needed the user may request a resending of the code for up to `MaxNumberOfSmsSendings` times. If this threshold is reached the user will be redirected to the login page. Default value: **3**;
- `multiFactorAuthentication/runtime/providers/provider`  
`[@name='Email']/roles` will include a `<role name=""/>` child for each role that contains users that have to be authenticated through this provider. Note that a `<role name="*" />` means that all roles will be taken into account;

Once you've activated the Email-based authentication, you need to configure the Job Server for Multi-Factor Authentication.

#### Step 2. Configure the Job Server for MFA

### IMPORTANT!

The MessageBus (OCS) plugin for the FintechOS Job Server already includes the configurations required for multi-factor authentication (see the *FintechOS Installation Guide* for details about MessageBus (OCS) installation).

- If you have the MessageBus (OCS) plugin installed, skip this step.
- If you are using the standard Job Server configuration, follow the instructions below to configure the multi-factor authentication settings.

1. On the server where the HPFI installation package resides, go to the **schedule.config** file and add the following section:

```
<triggers>
...
<trigger>
  <name>FTOS.OCB.OTP</name>
  <startTime>02.11.2017 11:00</startTime>
  <endTime>03.11.2080 11:02</endTime>
  <repeatCount>-1</repeatCount>
  <rescheduleAfterRun>false</rescheduleAfterRun>
  <async>false</async>
  <expression>0/10 * * * * ?</expression>
```

2.

```

    <services>

    <service>FTOS.OCB.SendMessagesServiceEmailOTP</service>

    <service>FTOS.OCB.UpdateStatusServiceEmailOTP</service>

    <service>
    FTOS.OCB.UpdateExpiredMessageServiceEmailOTP</service>
    </services>
  </trigger>
</triggers>

```

3. On the server where the HPFI installation package resides, go to the **services.config** file and add the following sections:

```

<serviceList>
  ...
  <!--OTP-->
  <service>
    <name>FTOS.OCB.SendMessagesServiceEmailOTP</name>
    <type>class</type>
    <method></method>

  <
  class
  >
  FTOS.MessageBus.ScheduledServices.SendMessagesService</class>

  <assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

  <execParams>
  provider=gateway;providerSetting=GatewayEmailOTP</execParams>
  </service>
  <service>
    <name>FTOS.OCB.UpdateStatusServiceEmailOTP</name>
    <type>class</type>
    <method></method>
    <class>
  FTOS.MessageBus.ScheduledServices.UpdateStatusMessagesService
  </class>

  <assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

```

- 4.

```

<execParams>
provider=gateway;providerSetting=GatewayEmailOTP</execParams>
  </service>
</service>

<name>FTOS.OCB.UpdateExpiredMessageServiceSmsOTP</name>
  <type>class</type>
  <method></method>
  <class>
FTOS.MessageBus.ScheduledServices.UpdateExpiredMessageService
</class>

<assembly>FTOS.MessageBus.ScheduledServices</assembly>-->

<execParams>
provider=gateway;providerSetting=GatewayEmailOTP</execParams>
  </service>
</serviceList>

```

## Register TLS Client Certificates

Client certificates allow you to access web services that require client authentication via the TLS/SSL protocol. Once a certificate is registered, you can refer it in your server side scripts and include it in API calls.

To register a TLS Client Certificate add the following secret in Vault:

| Key Path                                    | Key Name                                  |
|---|---|
| kv/<environment>/<application>/app-settings | automation-client-certificate-clientCert1 |

| Key Name                                  | Key Value  |
|---|--|
| automation-client-certificate-clientCert1 | "{ 'storeName': 'My', 'storeLocation': 'LocalMachine', 'thumbPrint': 'd77621fa50114404a6e5820c6d066b019c13fdd8', 'description': 'Client certificate for Api1' }" |

You must provide a programmatic **name**, preceded by the `automation-client-certificate-` prefix. For instance, in the example above, the name of the client certificate is going to be `clientCert1`.

The **value** is provided in JSON format and must be XML escaped. For simpler scenarios you can use single quotes instead of double quotes. The JSON value has the following structure:

```
{
  "storeName": "My",
  "storeLocation": "LocalMachine",
  "thumbPrint": "d77621fa50114404a6e5820c6d066b019c13fdd8",
  "description": "Client certificate for Api1",
  "checkValidity": true
}
```

| Property  | Description   |
|-----------|---|
| storeName | <p>You can populate the storeName property with one of the following values:</p> <ul style="list-style-type: none"><li>• AddressBook - X.509 certificate store for other users.</li><li>• AuthRoot - X.509 certificate store for third-party certificate authorities.</li><li>• CertificateAuthority - X.509 certificate store for intermediate certificate authorities.</li><li>• Disallowed - X.509 certificate store for revoked certificates.</li><li>• My - X.509 certificate store for personal certificates.</li><li>• Root - X.509 certificate store for trusted root certificate authorities.</li><li>• TrustedPeople - X.509 certificate store for directly trusted people and resources.</li><li>• TrustedPublisher - X.509 certificate store for directly trusted publishers.</li></ul> |

| Property      | Description   |
|---------------|---|
| storeLocation | <p>You can populate the storeLocation property with one of the following values:</p> <ul style="list-style-type: none"> <li>• <b>CurrentUser</b> - X.509 certificate store used by the current user.</li> <li>• <b>LocalMachine</b> - X.509 certificate store assigned to the local machine.</li> </ul>         |
| thumbPrint    | This is the thumbprint of the client certificate.   |
| description   | A user-friendly description of the certificate. This information will be displayed in the code editor's intelligent code completion suggestions.  |
| checkValidity | <ul style="list-style-type: none"> <li>• <b>true</b> - Even if the thumbprint is found, the API returns the certificate only if the root issuer in the certificate build chain is part of the trusted root certification authorities.</li> <li>• <b>false</b> - For development or testing purposes.</li> </ul> |

## (Deprecated) Add configuration in web.config files:

```

<app-settings>
  ...
  <add key="automation-client-certificate-
clientCert1" value="{ 'storeName': 'My', 'storeLocation':
'LocalMachine', 'thumbPrint':
'd77621fa50114404a6e5820c6d066b019c13fdd8',
'description':'Client certificate for Api1' }"/>
  or
  <add key="automation-client-certificate-
clientCert1" value="{ &apos;storeName&apos;;: &apos;My&apos;;,
&apos;storeLocation&apos;;: &apos;LocalMachine&apos;;,
&apos;thumbPrint&apos;;:
&apos;d77621fa50114404a6e5820c6d066b019c13fdd8&apos;;, &apos;
description&apos;;:&apos;Client certificate for Api1 a&apos;
}"/>
  ...
</app-settings>

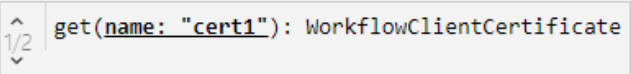
```

## Usage in server-side scripts

The automation API supports referencing client certificates and passing them in the `httpGet/httpPost` functions. For more information, see the [Server SDK Reference Guide](#) documentation.

```
var cert = server.clientCertificates.get('clientCert1');
var getResult = httpGet('https://server.com/route1', {}, {
  clientCertificate: cert
});
var postResult = httpPost('https://server.com/route2', myPostData, {
  clientCertificate: cert
});
```

In the code editor, the `server.clientCertificates.get` function provides automatic code completion suggestions for the registered client certificates.



```
var cert = server.clientCertificates.get(
```

## Configure JSON Web Token (JWT) Providers

An access token is a security key issued by an authorization server to provide access to Web APIs and other protected resources.

To access a resource that uses JWT token authentication, you need to register the connection settings for that resource's authentication provider in Vault secrets. Once the provider is set up, you can refer it in your server-side scripts to retrieve an access token for the supported resources.

To register a JWT authentication provider, add the following secrets:

| Key Path                                    | Key Name                     |
|---|------------------------------|
| kv/<environment>/<application>/app-settings | feature-jwt-token-provider-a |
| kv/<environment>/<application>/app-settings | feature-jwt-token-provider-b |

For each key, you must provide a programmatic name, preceded by the `feature-jwt-token-provider-` prefix. The programmatic name must also match the `name` property provided in the key's value. For instance, in the example above, the name of the providers are going to be `a` and `b` respectively.

The value is provided in JSON format and must be XML escaped. For simpler scenarios you can use single quotes instead of double quotes, as exemplified above.

The following properties are generic and apply to all authentication providers:

| Property                           | Description   |
|------------------------------------|---|
| <code>name</code><br>(required)    | The provider's name. The value has to be unique in the provider keys registry collection.   |
| <code>type</code><br>(required)    | The type of provider. Currently, only the Azure Active Directory token provider is supported, so the only valid value is <code>azure-ad-provider</code> . More authentication providers may become available in the future. |
| <code>timeout</code><br>(optional) | The service timeout in milliseconds, indicating for how long the application should wait for the token to be generated. Default value: 10000 (10 seconds).  |

In addition to the generic properties, you must also provide settings that are particular to each authentication provider, in accordance with their specifications. Currently, only Azure Active Directory is supported, with additional providers to be potentially added in the future. For Azure Active Directory, the following properties apply:

| Property                                       | Description   |
|--|---|
| <code>scopeForAccessToken</code><br>(required) | Azure AD scope for which the access is requested.   |
| <code>instance</code> (required)               | Azure AD instance name.   |
| <code>tenantId</code> (required)               | Azure AD tenant ID.   |
| <code>clientId</code> (required)               | Azure AD client ID.   |
| <code>clientSecret</code>                      | The application client secret used to retrieve the access token. The property is mutually exclusive with <code>clientCertificate</code> , but one of them must be set. The client secret must exist in the targeted Azure AD instance.  |
| <code>clientCertificate</code>                 | The certificate used to retrieve the access token. The property is mutually exclusive with <code>clientSecret</code> , but one of them must be set. The certificate should be registered in the targeted Azure AD instance.<br>For more information about client certificates, see <a href="#">"Register TLS Client Certificates" on page 248</a> . |

## (Deprecated) Add keys in the web.config file

```

<app-settings>
  ...
  <!-- Token provider configuration using client secret -->
  <add key="feature-jwt-token-provider-a" value="{
    'name': 'a',
    'type': 'azure-ad-provider',
    'scopeForAccessToken': 'api://xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxx/.default',
    'instance': 'https://login.microsoftonline.com/',
    'tenantId': 'yyyyyyyy-yyy-yyy-yyy-yyyyyyyyyyyyyy',
    'clientId': 'zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz',
    'timeout': 10000,
    'clientSecret': '~xyzxyz-xxxxxxxxxx-yyyyyyyyyy-zzz~x'
  }"/>

  <!-- Token provider configuration using client
certificate -->
  <add key="feature-jwt-token-provider-b" value="{
    'name': 'b',
    'type': 'azure-ad-provider',
    'scopeForAccessToken': 'api://xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxx/.default',
    'instance': 'https://login.microsoftonline.com/',
    'tenantId': 'yyyyyyyy-yyy-yyy-yyy-yyyyyyyyyyyyyy',
    'clientId': 'zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz',
    'timeout': 10000,
    'clientCertificate': {
      'storeName': 'My',
      'storeLocation': 'CurrentUser',
      'thumbPrint':
'xyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyzxyz',
      'description': 'Client certificate for wso2-devel',
      'checkValidity': false
    }
  }"/>

</app-settings>

```

## Usage in server-side scripts

To retrieve a JWT access token in a server-side script, use the `getJwtTokenByProviderName` function in the code editor. For example:

```
var myToken = getJwtTokenByProviderName('a')
```

For more information, see the [Server SDK Reference Guide](#).

## Authorization

In HPFI, access to specific resources (authorization) is done via security role-based access which enables you to

- Protect information from being mishandled by users.
- Ensure that users have access to information based on business need to know.

This section covers platforms' critical aspects of segregation of duties and data ownership.

## Security Roles

Users with elevated privileges (admin users) can control data access by setting up the organizational structure to protect sensitive data and configuring various organization layers to allow communication, collaboration or reporting.

To set up the organizational structure, they need to create the business units, security roles, and assign users the appropriate security roles to map the job-related responsibilities with the required level of access privileges within the platform.

You can grant even more granular access privileges in HPFI, by associating security roles to digital journeys, digital journey steps, business workflows, dashboards, endpoints and DB tasks. The data is automatically filtered based on the privileges and level of access defined within the security role via the security items.

The lowest level of access privileges you can grant to users in HPFI is on attribute level. You can choose if a specific attribute (field) is to be mandatory, recommended or optional, by selecting the desired option from the Required Level drop-down:

- None – The field is optional. No error message will be displayed if the field is empty.
- Recommended – A blue dot will be displayed on the upper-left corner of the field in the user interface to indicate that it might be useful to fill in the field.
- Required - A red dot will be displayed on the upper-left corner of the field in the user interface to indicate that it is a mandatory field. The end user will not be able to add a new record if the field will be left blank.

#### NOTE

- You can only add required attributes to entities which have no records (empty entities), so if you try adding a required attribute to an entity for which you already have required attributes stored within the database, you'll receive an error message.
- You can add required attributes without creating constraints in the database, from entity form/digital journey configuration page, Advanced tab > After Events tab, by providing a code in the JavaScript field and the capabilities of field options.

For information on how create security roles and how to provide granular access to entities, digital journeys and dashboards, see the [Innovation Studio User Guide](#).

## Data Ownership

In HPFI, data ownership is given by the security roles, which allows you to manage complex scenarios of access privileges and the level of access.

Admin users are the ones who can define the organizational structure, create users and assign the security roles according to the business need-to-know, inline with their job responsibilities.

The information presented in the user menu and the actions a user is able to perform are aligned with the security roles assigned.

For information on how create the organizational structure, add users and assign security roles, see the [Innovation Studio User Guide](#), section Security.

## Password Security

By default, FintechOS can log into the Innovation Studio by using FintechOS credentials: username and password. After successfully logging in, users can access the FintechOS resources based on the privileges granted by the security role assigned.

HPFI has various options in place to ensure password security:

- prevent users to log in using a wrong password
- set the password to expire
- allow users to recover their password
- set password complexity
- forbid users setting their password matching previous passwords
- forbid users logging in with expired passwords
- lock users who have been inactive for a specific number of days

In order to comply with any password policies that might be enforced within your organization, you can customize the HPFI password complexity either from the **web.config** file (see section [Global Password Complexity Settings](#)) or by using server scripting (see section [Customize Password Complexity Rules using Server Scripting](#)).

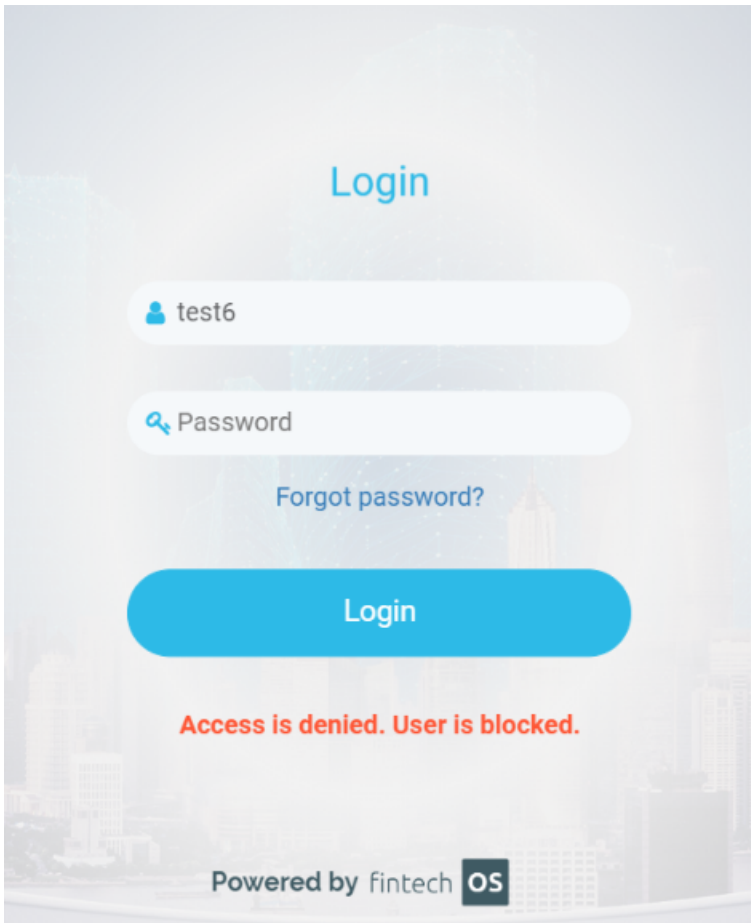
When users will choose to reset their password, an email is sent to the email address associated with their HPFI account. HPFI offers a default email template that is used for password reset. It's easy to customize the [default email template](#), or by using [server scripting](#).

If the Forgot Password feature has been activated, users will be able to reset their password from the login page by providing either their emails address or their username.

In addition to the forgot password security, you can also [forbid access for users who have been idle for a specific period of time](#).

## Locked account

If users enter a wrong password multiple times, reaching the maximum number of retries (that is, 5), their account will be locked.



To unlock their account they should contact their HPFI admin to unlock their account. After the account is unlocked, they will be able to log in using the last password (if they remember it) or recover the password if they forgot it.

## Password expired

If the password is expired, a message displays on the login page notifying the user that the password. It also provides the user with the option to reset the password.

**NOTE** This feature is available only for EBS Authentication Provider.

## Activate Forgot Password Feature

In HPFI, the Forgot Password feature allows users to reset their password and enables HPFI developers to set the password complexity and to customize the password reset email template.

**NOTE** The forgot password feature is disabled by default and the validity of the password reset token is by default 15 minutes.

To activate the forgot password feature, on the server where the HPFI installation package resides, set the following secret:

| Key Path                                    | Key Name               | Key Value |
|---|------------------------|-----------|
| kv/<environment>/<application>/app-settings | feature.reset-password | 1         |

## (Deprecated) Set key in the **web.config** files:

```
<app-settings>
  <add key="feature.reset-password" value="1" />
  ...
</app-settings>
```

In order to send email instructions to users who have requested password reset from the login page, also make sure to include the following:

| Key Path                                    | Key Name          |
|---|-------------------|
| kv/<environment>/<application>/app-settings | SMTP:Port         |
| kv/<environment>/<application>/app-settings | SMTP:Host         |
| kv/<environment>/<application>/app-settings | SMTP:EnableSSL    |
| kv/<environment>/<application>/app-settings | SMTP:User         |
| kv/<environment>/<application>/app-settings | SMTP:Password     |
| kv/<environment>/<application>/app-settings | DefaultFrom Email |

## (Deprecated) Using web.config files:

```

<add key="SMTP:Port" value="" />
<add key="SMTP:Host" value="" />
<add key="SMTP:EnableSSL" value="" />
<add key="SMTP:User" value="" />
<add key="SMTP:Password" value="" />
<add key="DefaultFromEmail" value="" />

```

The default value of the password reset token is 15 minutes. In order to set the validity to another time frame, configure the following key:

For token expiration after 5 minutes:

| Key Path                                    | Key Name                | Key Value |
|---|-------------------------|-----------|
| kv/<environment>/<application>/app-settings | PasswordResetExpiration | 00:05:00  |

## (Deprecated) Using web.config keys:

```

<app-settings>
  <add key="PasswordResetExpiration" value="00:05:00"/>
  ...
</app-settings>

```

## Configure Password Change

HPFI provides you with various options to configure password change:

- set the period of time (in hours) to pass until users are able to change their password.
- set the period of time (in days) allowed before a password must be changed.
- configure password change based on the password history.

## Setting password minimum age

The minimum password age setting determines the period of time (in hours) that a password can be used before the users can change their password.

To set the password minimum age, on the server where the HPFI installation package resides, add the following:

In **Vault** secrets:

| Key Path                                    | Key Name                              | Key Value |
|---|---------------------------------------|-----------|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-password-min-age | 24        |

Where **value** is the number of hours until users can change their password.

If **value** is empty or a negative value or the key is missing from **web.config** the **minimum password age** is set to 0 hours allowing immediate password changes, which is not recommended.

When using the minimum password age, we recommend you to configure the password history as well. This way you prevent users to changing their password with the same password.

## (Deprecated) Add password minimum age key in web.config files:

```
<add key="core-setting-ebsauth-password-min-age" value="24"/>
```

## Setting password expiry

The maximum password age setting determines the period of time (in days) that a password can be used before the system requires the user to change it.

| Key Path                                    | Key Name                              | Key Value |
|---|---------------------------------------|-----------|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-password-max-age | 30        |

Where **value** is the number of days allowed before a password expires and should be changed. The maximum number of days is limited to 999. If value is empty, 0 or a negative value or the key is missing, the password expiration feature is disabled, that is, the password never expires, which is not recommended.

If the user tries to authenticate with an expired password the login page will provide the user with the option to reset the password only if the [reset password feature is enabled](#).

## (Deprecated) Add password maximum age key in web.config files:

```
<add key="core-setting-epsauth-password-max-age" value="30"/>
```

## Configuring password change based on password history

HPFI provides you with the password history features which allows you to set whether a new password is checked against passwords stored in the user's password history. This prevents the user from re-using a recently used password.

To configure the password change to take into consideration user's password history, on the server where the HPFI installation package resides, go to the **Vault** and add the following secret:

| Key Path                                    | Key Name                                    | Key Value |
|---|---|-----------|
| kv/<environment>/<application>/app-settings | core-setting-epsauth-password-history-depth | 5         |

Where **value** is the number of historical passwords that will be checked when a user tries changing the password. If the user tries to set one of the old passwords then the system will forbid user to use that password. If **value** is empty, 0 or a negative value or the key is missing from the **web.config** file, the password history feature is not enabled (i.e. the user can change the password with the same password).

## (Deprecated) Add password change based on password history key in web.config files:

```
<add key="core-setting-ebsauth-password-history-
depth" value="5"/>
```

## Setting password about to expire notifications

You might want to remind users that they should change their passwords within x days before their password expired. FintechOS allows you to set such a notification to be shown on a web page and also customize the notification message.

To set the password expiry notification, on the server where the HPFI installation package resides, go to **Vault** and add the following secret:

| Key Path                                    | Key Name  | Key Value |
|---|---|-----------|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-password-about-to-expire-days-until-expiration | 30        |

If the number of days until the password will expire is less than the **value** specified, a page with the remaining days will be shown.

The notification message is localizable, so in order to be properly interpreted by the system, make sure that the text is a json array.

**(Deprecated.)** Add password expiry notification in the web.config file:

```
<add key="core-setting-ebsauth-password-about-to-expire-
days-until-expiration" value="30"/>
```

To customize the notification message ,add the following secret in **Vault**:

| Key Path                                    | Key Name   | Key Value  |
|---|--|--|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-password-about-to-expire-meessage | [{'en-GB': 'Password will expire in {10} days.'}, {'ro-RO': 'Parola va expira in {10} zile.'}] |

When the language is set to Romanian the message will be : "Parola va expira in {10} zile.", where {10} is the number of days until the password will expire.

The Server SDK function `usersAboutToExpirePasswords(int passwordExpireDaysMax)` enables you to get the list of users for which the password will expire in '`passwordExpireDaysMax`' days or less.

## (Deprecated) Add customize the notification message key in web.config file:

```
<add key="core-setting-ebsauth-password-about-to-expire-  
message" value="[{'en-GB': 'Password will expire in {10}  
days.'}, {'ro-RO': 'Parola va expira in {10} zile.'}]" />
```

## Skipping the password expiry rule for specific security roles

**NOTE** To ensure higher security, we recommend you to use this feature only in rare specific cases, e.g., for admin accounts.

To set password never expire for users who have specific security roles, add the following secret:

| Key Path                                    | Key Name  | Key Value    |
|---|---|--------------|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-password-expired-expected-role | securityRole |

The users with the security role specified in the value will never have to reset the password due to the password expiry rule.

## (Deprecated) Add key in web.config to set password never to expire

```
<add key="core-setting-ebsauth-password-expired-expected-  
role" value="securityRole" />
```

## Reset Password Global Email Template

The default email template for password reset is named: “ResetPasswordEmail” and it is included in the installation script.

To see the content of the default email template, from the Admin menu, click Email Templates. The Email Templates List page appears. Double-click on the “ResetPasswordEmail” record. The Edit Email Template page will be displayed.

EDIT EMAIL TEMPLATE

EMAIL TEMPLATE

Template name

ResetPasswordEmail

Subject

Reset your FintechOS password

Body

File Edit Insert View Format Table Tools

Undo

Redo

Formatters

**B**

*I*

UI Designer

Hello,

No need to worry, you can reset your password by clicking the link below:

[Reset password](#)

Your username is: {userName}

If you didn't request a password reset, feel free to delete this email.

Thanks,

FintechOS Team.

**NOTE** You can change the content of the default email template based on your preferences, but make sure to include in the template the following tokens: **username** and **generatedToken**, otherwise, the email sent to users will contain incomplete information.

You can also customize the email template by using server scripting. For information on how to do it, see [Customize Reset Password Email Template using Server Scripting](#)

## Customize Reset Password Email Template

You can customize the reset password email template using server scripting (automation scripts) by following two steps:

## Step 1. Add a specific secret in Vault

Add the following secret in Vault file in order to provide the name of the automation script name which customizes the email template.

| Key Path                                    | Key Name                               | Key Value    |
|---|--|--------------|
| kv/<environment>/<application>/app-settings | ResetPasswordEmailTemplateWorkflowName | WorkflowName |

If you do not provide the name of the automation script for email template customization, the system will search for an on-demand automation server script named “FTOS\_ResetPasswordEmail”. For backwards compatibility, the system also searches for ‘ResetPasswordEmail’.

**NOTE** The FTOS\_ResetPasswordEmail” on-demand automation server script does not exist by default; you have to create it.

## (Deprecated) Add key in web.config file:

In the **web.config** files, on the server where the HPFI installation package resides:

```
<app-settings>
    <add
      key
      =
      "ResetPasswordEmailTemplateWorkflowName"
      value="WorkflowName"/>
      ...
</app-settings>
```

## Step 2. Create FTOS\_ResetPasswordEmail on-demand automation script

The automation script offers customization based on associated user and roles.

The new password reset email template must be returned as the “emailTemplate” key of the Values property:

```
var user = context.Values["user"];
```

```

for(var i=0;i<user.Roles.length;i++)
{
    let role = user.Roles[i];
    if (role.Name == "special")
    {
        context.Values["emailTemplate"] =
"SpecialEmailTemplate";
        break;
    }
}

```

The user value has the following format:

```

{
  "UserName" : "user1",
  "BusinessUnitId" : "guid",
  "DisplayName" : "user display name",
  "Email" : "user email",
  "ExternalId" : "guid",
  "OrganizationId" : "guid"
  "Roles" :
  [
    {
      "SecurityRoleId" : "guid",
      "Name" : "role name 1"
    },
    {
      "SecurityRoleId" : "guid",
      "Name" : "role name 2"
    }
  ]
}

```

For information on how to create an on-demand server automation scripts, see the [Innovation Studio User Guide](#), section *Creating On-demand Server Automation Scripts*.

## Global Password Complexity Settings

For the default Membership provider, the complexity of the password is controlled by the following settings in the **web.config** file:

- minimum required password length
- minimum required non alpha numeric characters
- password strength regular expression

web.config settings for password complexity:

```
<membership defaultProvider="SqlProvider"
  userIsOnlineTimeWindow = "20">
  <providers>
    <add name="CustomMembership"
      type="EBS.Core.Authentication.Providers.CustomMembership"
      connectionStringName="EbsSqlServer"
      ...
      minRequiredNonalphanumericCharacters="1"
      minRequiredPasswordLength="7"
      passwordStrengthRegularExpression="(?=.*[A-Z].*[A-Z])(?=.*
[#@$*!& ;])(?=.*[0-9].*[0-9])(?=.*[a-z].*[a-z].*[a-z])"
    />
  </providers>
</membership>
```

You can also customize the password complexity by using server scripting. For more information, see [Customize Password Complexity Rules using Server Scripting](#).

## Customize Password Complexity Rules

You can customize the password complexity using server scripting (automation scripts) by following two steps:

### Step 1. Add a specific secret in Vault:

Add the following secret in Vault in order to provide the name of the automation script name which configures the password complexity.

| Key Path                                    | Key Name                       | Key Value    |
|---|--------------------------------|--------------|
| kv/<environment>/<application>/app-settings | ResetPasswordRulesWorkflowName | WorkflowName |

If you do not provide the name of the automation script for password complexity customization, the system will search for an on-demand automation server script named “FTOS\_ResetPasswordRules”.

**NOTE** The “FTOS\_ResetPasswordRules” on-demand automation server script does not exist by default; you have to create it.

## (Deprecated) Add key in web.config files:

On the server where the HPFI installation package resides, add in the **web.config** file:

```
<app-settings>
  <add
    key="ResetPasswordRulesWorkflowName" value="WorkflowName"/>
    ...
</app-settings>
```

### Step 2. Create FTOS\_ResetPasswordRules on-demand automation script

The server automation script offers customization based on password content and associated user and roles.

For information on how create an on-demand server automation script. For information on how to create an on-demand server automation scripts, see the [Innovation Studio User Guide](#), section *Creating On-demand Server Automation Scripts*.

Do not permit passwords containing letter ‘z’

```
var password = context.Values["password"];
if (password.match(/z/))
    throw new Exception("Password contains letter z");
```

Context contains two keys in the **Values** property:

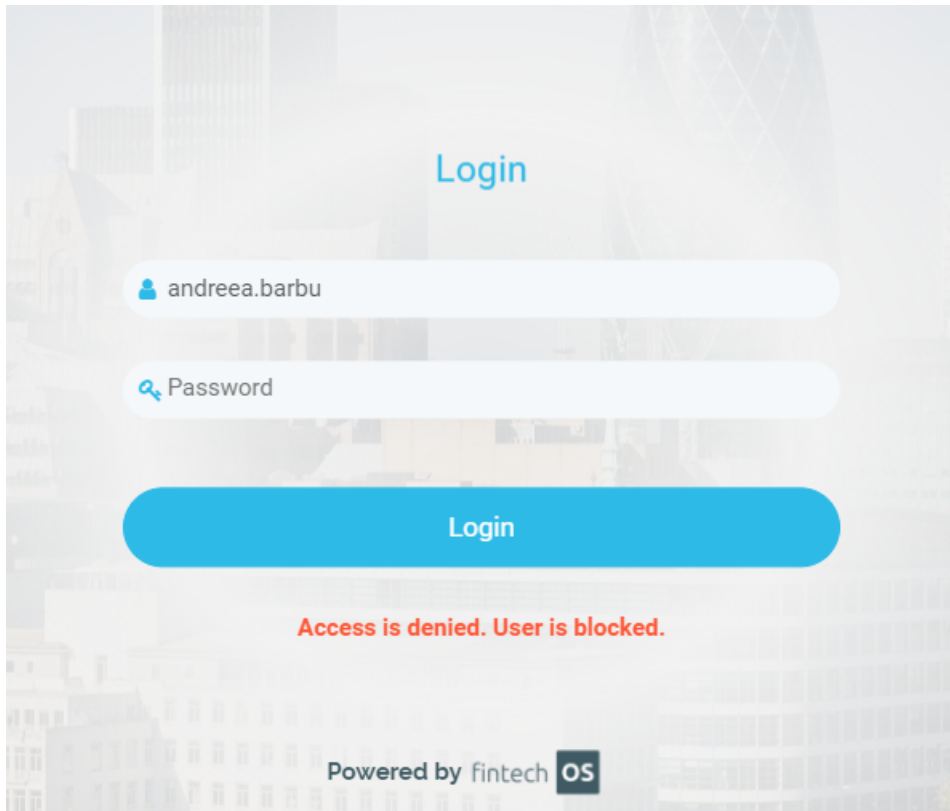
- password
- user, which contains a JSON similar to:

```
{
  "UserName" : "user1",
```

```
"BusinessUnitId" : "guid",
"DisplayName" : "user display name",
"Email" : "user email",
"ExternalId" : "guid",
"OrganizationId" : "guid"
"Roles" :
[
  {
    "SecurityRoleId" : "guid",
    "Name" : "role name 1"
  },
  {
    "SecurityRoleId" : "guid",
    "Name" : "role name 2"
  },
]
}
```

## Temporary Blocked User

A temporary blocked user is the account that has opened the FintechOS Portal, inserted the wrong password for that said account for a maximum of five times or for a maximum that was set previously and cannot access the Portal anymore.



When the temporary block time interval has passed and the user wishes to reset the password, click **Forgot password** and follow the steps to receive the email with reset password which implies opening the e-mail and follow the reset password link. The user is unblocked so that reset password flow can be followed.

## How to setup the number of retries Portal

In order to create the setup, add the following configuration in Vault:

To configure the the maximum amount of retries (the default value is zero):

| Key Path                                    | Key Name                                      | Key Value |
|---|---|-----------|
| kv/<environment>/<application>/app-settings | core-setting-ebsauth-account-lockout-duration | 5         |
| kv/<environment>/<application>/app-settings | feature.reset-password                        | 1         |

## When using the EbsAuth provider

For those who are using the EbsAuth provider, to web.config insert an up to date key (“core-setting-ebsauth-account-lockout-duration”) to set the amount of minutes the user will not be able to access the account after having tapped in the false password. The key inserted earlier can be zero/ can be a negative value/ empty, then only the administrator has the power to unblock the account for the user.

### IMPORTANT!

The userId present in TemporarilyLockedAccount is deleted when an admin unlocks an account.

The system entity “**TemporarilyLockedAccount**” tracks the modifications happening when an user's account is blocked after having inserted the wrong password.

The feature temporary blocked user has the key set to a positive value, the user wishes to open the FintechOS Portal and **EbsAuth provider** states that the account has been locked after failed attempt to log in, there are two situations:

Firstly, the user was previously temporarily locked out with the current date/ time bigger than the Lockeduntil value, the user will be automatically unlocked, the data from the system entity “**TemporarilyLockedAccount**” is eased and the user will be able to use the Portal. However, when the current date/ time is smaller than the Lockeduntil value, the system will not automatically unlock, but the block will be effective.

Secondly, when the user is not blocked, the LockedUntil value is equal to aspnet\_Membership.LastLockoutDate plus value of “account-lockout-duration”. Then, when the current date/ time is bigger than the LockedUntil value, the user's account is automatically unblocked and the user will be able to use the Portal. Nevertheless, when the current date/ time is equal or smaller than the LockedUntil value, there is an entry in the TemporarilyLockedAccount and the user cannot log in the Portal.

## Send Notifications for Locked Accounts or Password Resets

To set up the notifications users receive when their account is locked (after reaching the maximum number of failed login attempts) or when they need to reset their passwords, on the server where the FintechOS platform resides, go to the **web.config** file and add the following settings:

```
<configuration>
  <configSections>
    ...
    <section name="ebsAuthProvider" type
="EBS.Core.Authentication.Common.Configuration.EBSAuthProviderConfig, EBS.Core.Authentication.Common"/>
    ...
  </configSections>
  ...
  <ebsAuthProvider>
    <notifications>
      <communicationChannels>
        <channel
name
="myChannel"

channelProvider
="channelProvider" communicationChannel="communicationChannel"/>
        </communicationChannels>
        <notificationTypes>
          <type
enabled
="true"

name
="UserLockedOutOnLastLogin"
messageTemplate="messageTemplate" from="no-reply@myCompany.com">
            <supportedChannels>
              <supportedChannel name="myChannel"/>
            </supportedChannels>
          </type>
```

```

        <type
enabled
="true"

name
="UserResetPasswordEmail"
messageTemplate="messageTemplate" from="no-reply@myCompany.com">
    <supportedChannels>
        <supportedChannel name="myChannel"/>
    </supportedChannels>
    </type>
</notificationTypes>
</notifications>
</ebsAuthProvider>
...
</configuration>

```

The notifications are set by configuring the `ebsAuthProvider` section with the communication channels and templates used to send the locked account and password reset messages.

## communicationChannels

Defines the communication channels available for sending notifications. For each `channel`, you can configure the following settings:

| Setting                           | Description   |
|-----------------------------------|---|
| <code>name</code>                 | Name used to identify the channel used to send notifications.   |
| <code>channelProvider</code>      | Provider used by the communication channel, such as GatewayEmailOTP or FTOSApiSms. Its value must be the Name of one of the records from <b>FTOS_DPA_ChannelProvider</b> entity.  |
| <code>communicationChannel</code> | <p>The type of channel by which the notification will be sent. Its value must be the Name of one of the records from <b>FTOS_DPA_CommunicationChannel</b> entity.</p> <div> <p><b>IMPORTANT!</b></p> <p>Currently, only email and SMS communication channels are supported. More channel types may be added in the future.</p> </div> |

## Custom email providers

If you wish to use an automation script to send your notifications via a custom email processor, configure the communication channel based on the following model:

```
<communicationChannels>
...
  <channel name="Email_With_
AutomationScript"

    channelProvider="CustomEmailProvider" communicationChannel="Email">
      <customProperties>
        <property
name="AutomationScriptName" value="myAutomationScript"/>
      </customProperties>
    </channel>
...
</communicationChannels>
```

Where `myAutomationScript` is the name of the automation script that will process the notification message. The automation script's `context.Data` object will include a data structure called `emailInfo`, which you can use for your custom processing:

```
...
"Data": {
  "emailInfo": {
    "from": "sender@a.com",
    "to": "recipient@b.com",
    "cc": null,
    "bcc": null,
    "body": "email body",
    "subject": "email subject"
  }
}
...
```

## notificationTypes

Defines the types of notification that will be sent automatically to the users. For each notification `type`, you can configure the following settings:

| Setting | Description   |
|---------|---|
| enabled | true/false. Activates or deactivates the notification type. |

| Setting                        | Description   |
|--------------------------------|---|
| <code>name</code>              | <ul style="list-style-type: none"> <li>UserLockedOutOnLastLogin - Notify the user after reaching the maximum number of failed login attempts.</li> <li>UserResetPasswordEmail - Send the user a message with the password reset link.</li> </ul>  |
| <code>messageTemplate</code>   | <p>Content template used for the notification message. For information on how to work with personalized content templates, see the <a href="#">Hyper-Personalization Automation User Guide</a>.</p> <p>Depending on the type of notification, you can insert the following tokens in the content template:</p> <ul style="list-style-type: none"> <li>UserLockedOutOnLastLogin - <code>{{user_display_name}}</code> and <code>{{application_name}}</code>. For example:<br/><i>The user {{user_display_name}} was blocked for {{application_name}}. Too many login attempts.</i></li> <li>UserResetPasswordEmail - <code>{{user_display_name}}</code> and <code>{{password_reset_link}}</code>. For example:<br/><i>Hello {{user_display_name}}. Use the following link to reset your password: {{password_reset_link}}.</i></li> </ul> |
| <code>from</code>              | Default email sender address or telephone number from which the notification was sent.  |
| <code>supportedChannels</code> | <p>Communication channels available for sending the notifications (based on the entries defined in the <a href="#">"communicationChannels" on page 274</a> section).</p> <p>If the user has a preferred communication channel configured, the notification uses the first matching supported channel. If there is no such supported channel, the first supported channel that is enabled is used instead.</p>   |

## Random Character Password Authentication

This method of authentication does not require the full password, only random characters are typed in by the user. This is done in order to mitigate potential "person in the middle" type of cyber attacks. The number of asked random characters is 3. For

example, if the password is "MyPassword" the user might be asked to provide the chars on positions 1,3,6 ('M','P','s'). The positions (indexes) asked are different on each attempt. The number of failed log-ins that block the user is 5. To support this, a new column was added to `EbsMetadata.SystemUser`, called `PartialPass`. This is populated by a JSON with the details necessary to validate the random character login.

Add the following setting in Vault, under `<app-settings>` to enable the feature:

- `ebsauth-partial-password` to true (default this is false),
- within the keys that have the following structure:

| Key Path   | Key Name   | Key Value         |
|--|--|-------------------|
| <code>kv/&lt;environment&gt;/&lt;application&gt;/app-settings</code> | <code>core-setting-ebsauth-partial-password</code> | <code>true</code> |

## (Deprecated) Add key in the `web.config` file:

```
<app-settings>
...
<add key="core-setting-ebsauth-partial-password"
value="true"/>
...
</app-settings>
```

## Architecture

### 1 Capture the Username

In order to determine the password identity (such as the password length), the username is captured firstly. Once the identity is set, random characters can be extracted from the password.

### 2 Generate the random characters

When the user is asked to input random characters, they are from the entire range of the password, and not just the minimum required length.

## Unauthorize Inactive Users

The company's security policies might require that users who have been idle for a specific number of days are forbidden access to the company's resources.

HPFI provides you with two SDK functions which enable you to identify any inactive HPFI users and disable their access as an extra security measure for protecting your HPFI resources against unauthorized access:

- `inactiveUsers(int daysOfInactivity)` - get the list of users who have not been active in HPFI in the last number of days specified by the **daysOfInactivity** parameter.
- `unauthorizeUser(string userName)` - makes the user who has the username specified by the **userName** parameter not authorized.

## Session Expiration Time

Each session is timed to a specific interval during which if the user presents no inactivity, the Studio/Portal will expire. To set the session timing, the key `core-setting-tokenExpiresIn` is used and it functions with a specific time syntax. The availability time frame when working inside the Studio and Portal is set using the following syntax `d/day/days m/min/minutes h/hour/hours s/sec/seconds`. For example, it is possible to set:

- `3 d 5 h`
- `3 days 5 hours 3 minutes 20 seconds`
- `3 d/days 5 h 3 m/min 20 s/sec`

The default value is 20 minutes.

The necessarily changes are made in the **web.config**. If core-setting-tokenExpiresIn is not found in the config, the legacy appSetting TokenExpiresIn is loaded.

## Example:

How to set the time for when the Portal/Studio should log out the user in Vault:

| Key Path                                    | Key Name                    | Key Value    |
|---|-----------------------------|--------------|
| kv/<environment>/<application>/app-settings | core-setting-tokenExpiresIn | 2d 12h 3m 5s |
| kv/<environment>/<application>/app-settings | core-setting-tokenExpiresIn | 600          |
| kv/<environment>/<application>/app-settings | TokenExpiresIn              | 1200         |
| kv/<environment>/<application>/app-settings | TokenExpiresIn              | 1h30m        |

How to set the time for when the Portal/Studio should log out the user in web.config :

```
<add key="core-setting-tokenExpiresIn" value="2d 12h 3m 5s"/>
<add key="core-setting-tokenExpiresIn" value="600"/> <!--
seconds-->
<add key="TokenExpiresIn" value="1200" />
<add key="TokenExpiresIn" value="1h30min" />
```

## OTP Login Session

The OTP login session expiry time can be configure in the **web.config** file. It is done as follows:

```
<multiFactorAuthentication
xmlns="http://fintechos.com/ebs/schemas/multiFactorAuthentication"
enabled="true" otpTimeout="120">
```

The `otpTimeout` attribute is configured in seconds. The default value is 300 seconds. If a negative value is inserted, then it defaults to 300 seconds.

## File Upload Malware Scanning

To configure anti-malware scanning for file uploads, add the following secrets in **Vault** :

| Key Path                                    | Key Name                          | Key Value        |
|---|-----------------------------------|------------------|
| kv/<environment>/<application>/app-settings | feature-upload-malware-detection  | 1                |
| kv/<environment>/<application>/app-settings | feature-upload-malware-use-remote | 1                |
| kv/<environment>/<application>/app-settings | feature-upload-malware-endpoint   | API endpoint     |
| kv/<environment>/<application>/app-settings | feature-upload-malware-apikey     | subscription-key |
| kv/<environment>/<application>/app-settings | feature-upload-malware-timeout    | 30               |

| Key                               | Description  |
|-----------------------------------|--|
| feature-upload-malware-detection  | Set to 1 to enable anti-malware scanning or 0 to disable it.   |
| feature-upload-malware-use-remote | <p>Set to 1 to use a remote scan engine (see below) or 0 to use the local anti-malware engine. The local anti-malware engine is available only for on-premise deployments. For cloud deployments, you can use only the remote scan engine. Default: 0.</p> <div> <p><b>NOTE</b></p> <p>Currently, the Kaspersky Scan Engine is the only remote scan engine supported. Additional scan engines may be added in the future.</p> </div> |

| Key                             | Description   |
|---------------------------------|---|
| feature-upload-malware-endpoint | (Remote scan engine only) API endpoint of the remote scan engine.   |
| feature-upload-malware-apikey   | (Remote scan engine only) Subscription key for the remote scan engine.  |
| feature-upload-malware-timeout  | (Remote scan engine only) Duration in minutes to wait for a response from the remote scan engine before rejecting the file.<br>Default: 30. |

## (Deprecated) Add keys in **web.config** files:

```
<add key="feature-upload-malware-detection" value="1"/>

<add key="feature-upload-malware-use-remote" value="1"/>

<add key="feature-upload-malware-endpoint" value="API
endpoint"/>
<add key="feature-upload-malware-
apikey" value="subscription-key"/>

<add key="feature-upload-malware-timeout" value="30"/>
```

## Log Management

A log is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries. Each entry contains information related to a specific event that has occurred within a system or network.

FintechOS logs contain records related to computer security (which are generated by sources such as antivirus software, firewalls, and intrusion detection and prevention systems), as well as logs generated by operating systems (on servers, workstations, and networking equipment) and applications.

## Security Logs

Irrespective of the device or application, it is imperative that log data has accurate time stamps.

In FintechOS, logs are generated at the following levels:

### 1. **Applications**

Applications log their activity in correlation with the business processes they support, particularly those operations that modify permissions or access rights.

These logs generally include:

- The business operation that was requested.
- Whether the request was accepted or denied.
- The time and date the operation was performed (Start and end times may be appropriate for long operations).
- Who initiated the operation.
- System and network resources used.
- Any information needed for business process controls.
- Client hardware and software characteristics.

### 2. **Systems**

Many components of the IT infrastructure generate logs. Examples of these components include:

- Operating Systems.
- Web servers.
- Database servers.
- Print servers.
- File servers.

- Authentication servers.
- DHCP servers.
- DNS servers.
- E-mail server logs.

### 3. **Network devices**

Many components of the network infrastructure generate logs. Examples of these components include:

- Routers.
- Switches.
- Wireless access points.
- Network-based firewalls, intrusion detection and prevention systems, next-generation firewalls.

These logs typically describe flows of information through the network, but not the individual packets contained in that flow. Information logged for a flow should include:

- Network (IP) addresses or telephone numbers of the end points.
- Service identifiers (port numbers) for each of the end points.
- Whether the flow was accepted or denied.
- Date, time, and duration of the flow.
- Number of packets and bytes used by the flow.

### 4. **Microsoft Azure**

Three types of platform logs are generated which record the following types of actions:

- Azure Active Directory Reports – detailed changes made in Azure AD and login activity.

- Activity logs – record operations performed on an Azure resource – such as creating a VM.
- Resource logs – capture operations performed within an Azure resource, such as querying a database or writing to a storage bucket.

Both Activity Logs and Resource Logs use the Log Analytics workspace to store and access these logs. Also, Log Analytics data is used/analyzed by SIEM to generate alerts/incidents based on defined analytics rules.

## Operating system logs and application logs

FintechOS operating system logs and application logs typically hold a variety of information, including computer security-related data.

### 1. Operating System logs

Operating systems (OS) for servers, workstations, and networking devices (e.g., routers, switches) usually log a variety of information related to security. FintechOS is logging the most common types of security-related OS data, as follows:

- System Events - System events are operational actions performed by OS components, such as shutting down the system or starting a service. Each event is timestamped and other supporting information may include event, status, and error codes; service name; and user or system account associated with an event.
- Audit Records - Audit records hold security event information such as successful and failed authentication attempts, file accesses, security policy changes, account changes (e.g. account creation and deletion, account privilege assignment), and use of privileges.

### 2. Applications/platform logs

Logging level can take one of the following values: Verbose, Debug, Information, Warning, Error, Fatal. A minimum level of logging for each logging sink can be set.

FintechOS is actively using two logging sinks which can be enabled together or separately: file logging and Application Insights logging. We also have SEQ logging already added to the platform and can add more Serilog sinks in future FintechOS releases (based on request).

Type of logs that we expect to meet:

- Platform
  - Exceptions in the execution of the FTOS platform logic.
  - Warning/Info messages added by the platform developers to have a better view on what happened in a certain workflow.
  - Errors where the exceptions are handled, and information is provided about the cause on why a specific logic failed.
  - Digital developers can decide to add their own logging with the desired log level in the automation scripts written for client specific implementations.
  - Items related to observability on the application performance/statistics are not handled at the platform level and should be extracted using, for example, AppInsights out-of-the-box capabilities.
  - Trace.log files configured by default with periodical switching to new files of logs. Trace log files are grabbed by external sources. (SIEM, log server).

- Database

There are four types of logging events which can be stored in the DB:

- ["HPFI API Logging" on page 289](#)
  - Optional setting – based on customer requirements
  - Can be filtered by source type (i.e. OpenApi, Data Service) and method name
  - Logs are stored in EbsLogs.ApiLog

- Attention needs to be paid to filtering and cleaning the logging table, as the high number of requests can increase the DB size considerably
- Starting with 22.1.0 this can also be sent directly to the same logging sink as platform logging (no filtering on security, it is all or nothing)
- CRUD Operations Logging
  - Optional setting – based on customer requirements
  - It can only be turned on and off without any other granularity settings – logging the CRUD events on all entities
  - Logs stored in EbsLogs.UniversalLog
  - Starting with 22.1.0 this can also be sent directly to the same logging sink as platform logging (no filtering on security, it is all or nothing)
- Authentication Events logging
  - Logs stored in EbsMetadata.SystemAudit
  - Optional setting starting with 22.1.0, always enabled on older versions
  - Starting with 22.1.0 this can also be sent directly to the same logging sink as platform logging (no filtering on security, it is all or nothing)
- ["Data Audit" on the next page](#)

Optional setting which by default is turned off

### NOTE

FintechOS is logging all CRUD and API operations executed in the platform, by default, in a separate database schema named EbsLogs. Database administrators can restrict read access for this schema and

grant insert rights only for the SQL login used by the FintechOS platform. Granular rights can be set up.

### 3. DCS/DCI (Digital Cloud Services and Digital Cloud Infrastructure)

Logs generated by integration with FTOS technology partners hold data needed for solution debugging, sometimes files sent to the partners and, in some cases, initial requests and responses. Logs are stored on virtual machines where solutions reside. Restricted access to these virtual machines is enforced.

## Data Audit

The forth pillar of HPFI security, logging, provides you with comprehensive audit trail of what happened at any given time and who performed the action.

The logging configuration is specified within the **web.config** file. The platform uses the log.NET component for logging and it generates a **trace\_roll.log** file and multiple **trace\_roll.dd-mm-yyyy.n.log** files. The log files are saved in the web directory.

**NOTE** FintechOS API logs and FintechOS LOGS are kept in different audit tables.

## Entity Audit

HPFI has an extensive audit functionality that can be enabled for any entity, allowing change tracking at entity level.

Using the Innovation Studio, users can activate the auditing feature for a specific entity, by selecting the **Is Audited** checkbox. When auditing is enabled, the platform creates and maintains a system entity named **{entityName}\_ADT** where all changes to the initial entity are recorded including: the type of changes on the entity, when the changes have been made and by whom.

When the user navigates to the list view of an entity with audit enabled the **History** button will be available on the toolbar.

Clicking the **History** button will open the History List view which lists all the changes associated to the current entity instance (the associated ADT entity).

When navigating to the detail view for an audited entity, the **History** button will open a list with all the changes associated to the current entity instance.

To programmatically navigate to the audit logs use the commands below:

To get all audit logs for the specified entity (where, the ID is the entity ID):

```
'entity/{entityName}/history/viewAll/{id}'
```

To get audit logs for the specified operation:

```
'entity/{entityName}/history/{operation}'
```

To get audit logs for two specific operations:

```
'entity/{entityName}/history/{opOne}/{opTwo}'  
'entity/{entityName}/businessTransactions/{id}'
```

The data audit is independent of entity records (when the **Audit enabled** checkbox is selected on entity). An unique identifier (UID) is automatically added by the system to records. When users delete records, based on the UID, the action is logged into the audit trail.

The History List view which lists all the changes associated to the current entity instance (the associated ADT entity) has a new column, Unique Identifier (UID).

If the user deletes an income of a customer. the action is logged into **{entityName}\_ADT**. The user can consult anytime the History List page on that customer entity and see that the income has been deleted, when and by whom.

## HPFI Logging

HPFI logs all CRUD operations executed in the platform, by default, in a separate database schema named EbsLogs.UniversalLog.

Database administrators can restrict read access for this schema and grant insert rights only for the SQL login used by the HPFI platform.

### How to Configure the Logging of CRUD Operations

To configure this feature, go in **Vault** and set the feature-universal-logging setting, as desired. By default, it is set to **0**, that is, the feature is enabled.

In **Vault** secrets:

| Key Path                                    | Key Name                  | Key Value      |
|---|---------------------------|----------------|
| kv/<environment>/<application>/app-settings | feature-universal-logging | 0 1 true false |

### (Deprecated) Add key in the **web.config** files:

```
<configuration>
  <app-settings>
    ...
    <add key="feature-universal-
logging" value="0|1|true|false"/>
  </app-settings>
</configuration>
```

## HPFI API Logging

HPFI logs the calls over the HPFI API (REST AND WCF) and DataService CRUD operations.

The logs are saved by default in a separate database schema named EbsLogs. Database administrators can restrict read access for this schema and grant insert only rights for the SQL login used by the HPFI platform.

Source names:

- OpenApi (REST endpoint)
- ApiService (WCF endpoint)
- DataService (MVC endpoint)

## EbsLogs.ApiLog Schema

| Field         | Type             | Description  |
|---------------|------------------|--|
| Id            | bigint           | identity, primary key                                |
| LogId         | uniqueidentifier | alternate unique key                                 |
| Tenant        | nvarchar(150)    | tenant name, default value: ebs_default              |
| UserName      | nvarchar(200)    | authenticated user name                              |
| Source        | nvarchar(150)    | controller name : OpenApi, ApiService or DataService |
| Method        | nvarchar(150)    | action name  |
| Request       | nvarchar(max)    | request parameter as JSON                            |
| Response      | nvarchar(max)    | response as JSON                                     |
| Message       | nvarchar(max)    | response message                                     |
| Exception     | nvarchar(max)    | response error                                       |
| Success       | bit              | success/error  |
| CreatedAtUtc  | datetime         | call moment UTC                                      |
| Duration      | bigint           | call duration milliseconds                           |
| CorrelationId | nvarchar(100)    | correlation id                                       |
| RequestId     | nvarchar(100)    | request id   |
| ApiInfo       | nvarchar(max)    | call authentication information                      |

## How to Configure the HPFI API Logging

To configure this feature, go to the **web.config** file and use a custom configuration section, as provided below:

```
<configuration>
  <configSections>
    <section name="ftosApiLogging"
      type
      ="EBS.Core.Utills.ApiLoggingConfiguration.ApiLoggingConfigSection,
      EBS.Core.Utills"/>
  </configSections>
```

```

<ftosApiLogging enabled="true|false">
  <sources>
    <source
name="OpenApi|ApiService|DataService" exclude="true|false">
      <methods>
        <method name="*">
          <input exclude="true|false">
            </input>
          </method>

        <method name="GetById" exclude="true|false">
          <input exclude="true/false">
            <properties>
              <property
name="A" exclude="true|false"/>
            </properties>
          </input>
          <output exclude="true|false">
            <properties>
              <property
name="B" exclude="true|false"/>
            </properties>
          </output>
        </method>
      </methods>
    </source>
  </sources>
</ftosApiLogging>
</configuration>

```

The configuration allows filtering the out from logging elements at different levels of granularity: source, method (action), input (request), output (result), input property, output property.

The user can configure all other methods of a source by specifying "\*" for method name. Any explicitly defined method will override all settings from "\*".

**NOTE** When a property is excluded, it will not be serialized in the log.

# Restrict Access to Innovation Studio Based on the StudioUser Role

Apart from the security roles that limit the access to specific features within Innovation Studio, an additional layer of security is provided by preventing users from logging in to Innovation Studio based on the StudioUser role.

For this, use the following configuration key in the ["Configuration Manager" on page 86](#)

**kv/{environmentName}/{studioName}/app-settings/feature-restrict-access-in-studio-to-studio-user-role**

| Setting | Description   |
|---------|---|
| 0       | All users can log to Innovation Studio regardless of the assigned security roles. This is the default behavior. |
| 1       | Only users with the StudioUser role can log in to Innovation Studio   |

This key is optional and, if not added, the behavior is equivalent to it being set to 0.

Keep in mind that the StudioUser role must be created and assigned to users manually.

Edit Security Role

Name: StudioUser

SECURITY ITEMS

| Entity  | Security Scope | Operations |
|---------|----------------|------------|
| No data |                |            |

+ Insert X Remove

5 10 20 1