



FIS PRIVATE CAPITAL SUITE

Authentication for Clients

Table of Contents

- [Microsoft Entra ID \(Azure Active Directory \(AD\)\) as IDP](#)
- [Configuring InvestranWeb to Authenticate with Client's Azure AD as IDP \(SAML 2.0 protocol\)](#)
- [Configuring Excel Add-In \(OIDC\) APIs, Tenant Information Service, RW IDP Service and Reporting Service](#)
- [Setting up Client Credential Flow for Authentication among Microservices](#)
- [OKTA as IDP](#)
- [Configuring InvestranWeb to Authenticate with Client's Okta as IDP \(SAML 2.0 protocol\)](#)
- [Configuring Excel Add-In \(OIDC\), APIs Tenant Information Service, RW IDP Service and RS](#)
- [Setting up Client Credential Flow for Authentication among Microservices](#)



Microsoft Entra ID (Azure Active Directory (AD)) as IDP

Microsoft Entra ID and Azure AD are similar in terms of functionality and license.

This section lists the following:

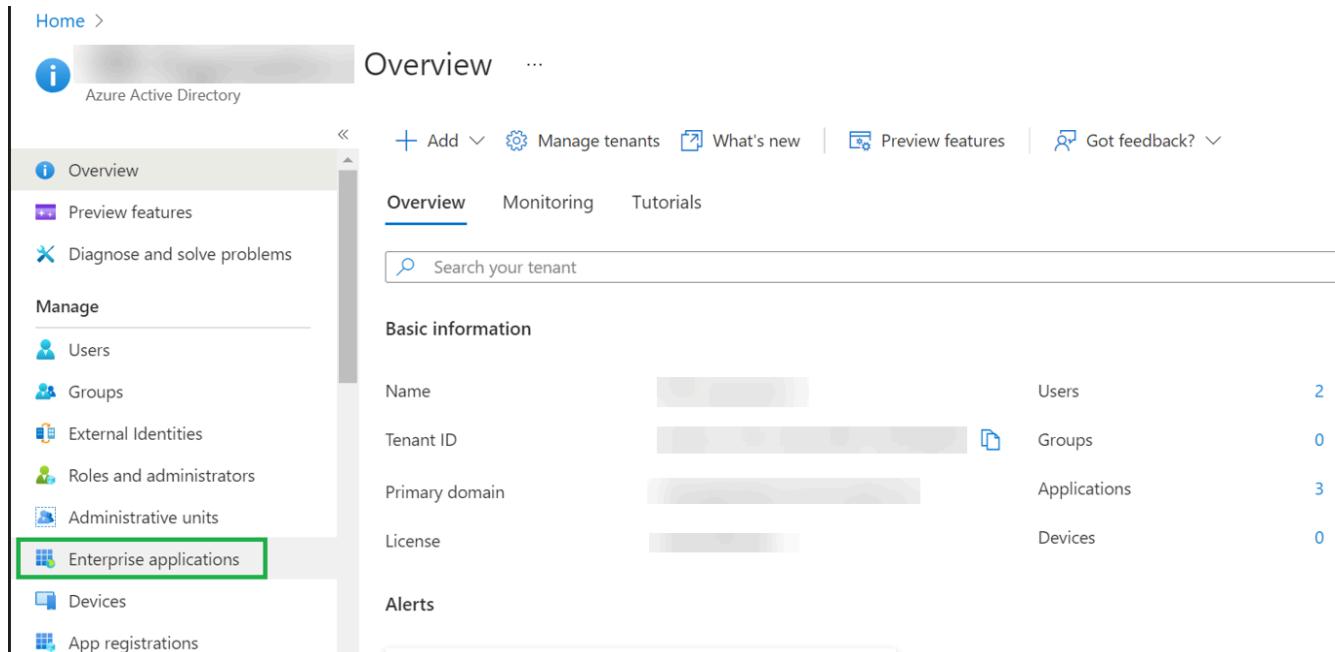
- [Configuring InvestranWeb to Authenticate with Client's Azure AD as IDP \(SAML 2.0 protocol\)](#)
- [Configuring Excel Add-In \(OIDC\) APIs, Tenant Information Service, RW IDP Service and Reporting Service](#)
- [Configure Setting up Client Credential Flow for Authentication among Microservices](#)

Configuring InvestranWeb to Authenticate with Client's Azure AD as IDP (SAML 2.0 protocol)

To setup clients to use their Azure AD tenant ID as the IDP for the main PCS Web application (InvestranWeb), do the following:

Note: Before starting the sign-in process, the user must have an active subscription for creating applications in Azure AD. In addition, they must have a Global Administrator Role in the respective Active Directory tenant in Azure.

1. Sign in to Azure Portal and click **Azure Active Directory**.
2. Ensure that the Azure Active Directory is created and along with that the custom domain is available for the account.



Home > **Azure Active Directory** Overview ...

+ Add Manage tenants What's new Preview features Got feedback? ▾

Overview Monitoring Tutorials

Search your tenant

Basic information

Name	Users
redacted	2

Tenant ID	Groups
redacted	0

Primary domain	Applications
redacted	3

License	Devices
redacted	0

Alerts

Manage

- Overview
- Preview features
- Diagnose and solve problems
- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications**
- Devices
- App registrations

3. Select **Enterprise applications > All applications (Preview)** and click **+ New application**.

Home > Enterprise applications

Enterprise applications | All applications (Preview)

Azure Active Directory

Overview

- Overview
- Diagnose and solve problems

Manage

- All applications (Preview) **(Selected)**
- Application proxy

New application Refresh Download (Preview) Preview

Want to switch back to the legacy Enterprise Apps search experience? Click here.

View, filter, and search applications in your organization that are set up to use Azure Active Directory.

Search by application name or object ID

Application type == Enterprise Applications X Applications status

- In the **Browse Azure AD Gallery** screen, click **+ Create Your Own Application**.
- In the **Create your own application** screen, select **Integrate any other application you don't find in the gallery (Non-gallery)**.
- Provide a name for your app in the text box and click **Create**. An app is created.

Home > VBK Organization > Enterprise applications >

Browse Azure AD Gallery

+ Create your own application Request new gallery app Got feedback?

You're in the new and improved app gallery experience. Click here to switch back to the legacy app gallery experience.

The Azure AD App Gallery is a catalog of thousands of apps that make it easy to deploy and configure single sign-on. Leverage prebuilt templates to connect your users more securely to their apps. Browse or create your own application.

Search application Single Sign-on : All User Account Management : All

Cloud platforms

Amazon Web Services (AWS) Google Cloud Platform

aws Google Cloud

Create your own application

Got feedback?

If you are developing your own application, using Application Proxy, or want to integrate an application that is not in the gallery, you can create your own application here.

What's the name of your app?

What are you looking to do with your application?

Configure Application Proxy for secure remote access to an on-premises application

Register an application to integrate with Azure AD (App you're developing)

Integrate any other application you don't find in the gallery (Non-gallery)

Create

- In the screen, add properties in the respective text fields and click **Setup single sign-on**.

The screenshot shows the 'Overview' page for an enterprise application named 'test'. The left sidebar includes options like Overview, Deployment Plan, Properties, Owners, Roles and administrators, Users and groups, Single sign-on, Provisioning, Application proxy, Self-service, and Custom security attributes (preview). The main area displays 'Properties' fields for Name, Application ID, and Object ID, along with a 'Getting Started' section containing two steps: '1. Assign users and groups' and '2. Set up single sign on'.

8. In the **Single sign-on** screen, select the **Disable** method.

The screenshot shows the 'Single sign-on' configuration screen for the 'test' application. The left sidebar includes options like Overview, Deployment Plan, Properties, Owners, Roles and administrators (Preview), Users and groups, Single sign-on (selected), Provisioning, Application proxy, Self-service, and Custom security attributes (preview). The main area displays four methods for selecting a single sign-on method: 'Disabled' (selected), 'SAML', 'Password-based', and 'Linked'.

9. In the **SAML-based Sign-on** screen, click **Edit** in the **Basic SAML Configuration** section
10. In the **Basic SAML Configuration** screen, enter the Identifier and Reply URL fields.

11. Save the configuration details and close the panel.
12. In the **Attributes and Claims** tab, click on the **Edit** option.

13. Click on **+ Add new claim** and provide details in the **Name** and **Source attributes** text field.

14. Once claim is saved, it will show up in the Attributes and Claims section.

The screenshot shows the Microsoft Azure portal interface for managing enterprise applications. The left sidebar lists various management options like Overview, Deployment Plan, Properties, Owners, Roles and administrators, Users and groups, Single sign-on (selected), Provisioning, Application proxy, Self-service, and Custom security attributes (preview). The main content area displays the configuration for the 'Azure_AD_SAML | SAML-based Sign-on' application. At the top, there are sections for 'Sign on URL', 'Relay State (Optional)', and 'Logout Url (Optional)'. Below this is the 'Attributes & Claims' section, which includes fields for 'givenname', 'surname', 'emailaddress', 'name', and 'LoginName[LC]'. The 'Edit' button is visible next to the claims list. The third section is 'SAML Signing Certificate', showing 'Status' as 'Active', 'Thumbprint', 'Expiration', and 'Notification Email'. The 'Edit' button is also present here. A large green box highlights the 'Download' links for 'Certificate (Base64)', 'Certificate (Raw)', and 'Federation Metadata XML'.

15. In the **SAML Signing Certificate** section, the user can download the certificate. Then, copy the login and logout URL.
Note: These certificate needs to be in xml for handshake.

This screenshot continues from the previous one, focusing on the 'SAML Signing Certificate' section. It shows the same fields: Status (Active), Thumbprint, Expiration, and Notification Email. The 'Download' links for Certificate (Base64), Certificate (Raw), and Federation Metadata XML are highlighted with a green box. Below this, the 'Set up Azure_AD_SAML' section is shown, which requires linking the application to Azure AD. It includes fields for 'Login URL', 'Azure AD Identifier', and 'Logout URL', all of which are highlighted with a green box. A link 'View step-by-step instructions' is also visible.

16. Now the Enterprise application is created and SSO is also configured.

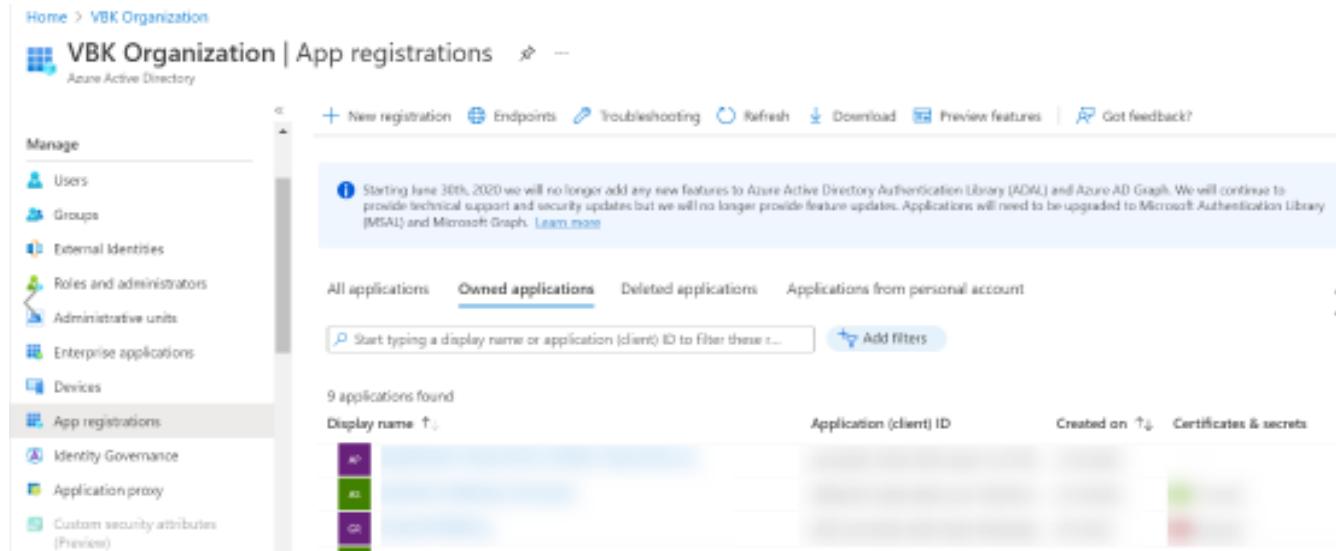
Configuring Excel Add-In (OIDC) APIs, Tenant Information Service, RW IDP Service and Reporting Service

Setting up the client's Azure AD to Authenticate the Excel Add-In

The Excel Add-In need to authenticate with Investran APIs. To achieve this, an app must be developed in the client's Azure AD tenant and the Investran API must be registered.

The following are the steps to setup client's Azure AD to authenticate the Excel Add-In:

1. Sign in to the Azure portal and in the left navigation menu open Azure Active Directory.
2. Ensure that the Azure Active Directory is created and then the custom domain is available for the account. Select the **App registrations**, and click **+ New Registration**.



The screenshot shows the Azure Active Directory App registrations page. The left sidebar has a 'Manage' section with options like Users, Groups, External Identities, Roles and administrators, Administrative units, Enterprise applications, Devices, App registrations, Identity Governance, Application proxy, and Custom security attributes (Preview). The 'App registrations' option is selected. The main area shows a message about the end of support for ADAL and MSAL. Below it, there are tabs for All applications, Owned applications (which is selected), Deleted applications, and Applications from personal account. A search bar says 'Start typing a display name or application (client) ID to filter these...'. There is a table with columns for Display name, Application (client) ID, Created on, and Certificates & secrets. The table shows 9 applications found, with three visible rows: 'AP', 'AS', and 'AR'.

3. Enter the name of the application (**rw-excel-add-in**) in the **Name** text field, select the supported account type and click **Register** to register the application.

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (VBK Organization only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

By proceeding, you agree to the Microsoft Platform Policies [↗](#)

[Register](#)

4. Note down client Id and Tenant Id which needs to be used while accessing the app.

Home > VBK Organization >

webapp ⚡ ...

Search (Ctrl+/) Delete Endpoints Preview features

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Overview Quickstart Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners

Essentials

Display name	Client credentials
webapp	Add a certificate or secret
Application (client) ID	Redirect URLs
<input type="text"/>	Add a Redirect URI
Object ID	Application ID URI
<input type="text"/>	Add an Application ID URI
Directory (tenant) ID	Managed application in local directory
<input type="text"/>	webapp

Supported account types

[My organization only](#)

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

'), object ID (''), directory ID (''), and supported account types ('My organization only'). The 'Client credentials' section shows a link to 'Add a certificate or secret'. The 'Redirect URLs' section shows a link to 'Add a Redirect URI'. The 'Application ID URI' section shows a link to 'Add an Application ID URI'. The 'Managed application in local directory' section shows a link to 'webapp'."/>

5. In the **Authentication** screen, click **+ Add a platform** and select **Single-page application**.

Configure platforms

Web applications

- Web**
Build, host, and deploy a web server application...NET, Java, Python
- Single-page application**
Configure browser client applications and progressive web applications. JavaScript

Mobile and desktop applications

- iOS / macOS**
Objective-C, Swift, Xamarin
- Android**
Java, Kotlin, Xamarin

Mobile and desktop applications

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (VBK Organization or ...)
- Accounts in any organizational directory (Any Azure AD directory - ...)

[Help me decide...](#)

Note: The Redirect uri must be RW Excel Add In uri+ /silent-refresh.html

- In the **Authentication** screen, type the redirect URL in the text field which will be redirected after authentication and select the Access and ID tokens and click **Save**.

Note: The Redirect uri must be RW Excel Add In uri+ /silent-refresh.html

WebAppUserApis | Authentication

Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

e.g. https://example.com/logout

Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. [Learn more about tokens](#).

Select the tokens you would like to be issued by the authorization endpoint:

- Access tokens (used for implicit flows)
- ID tokens (used for implicit and hybrid flows)

Save **Discard**

Creating API Application

The following are the steps to create the API Application:

1. Sign in to Azure Portal and in the left navigation menu open Azure Active Directory.
2. Ensure the Azure Active Directory is created and the custom domain is available for the account. Select the **App Registrations** and click **+ New Registration**.

The screenshot shows the Azure Active Directory portal with the 'App registrations' section selected. The 'Owned applications' tab is active, displaying a list of registered applications. A prominent message at the top states: "Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph." Below the message, there are tabs for 'All applications', 'Owned applications' (which is underlined), 'Deleted applications', and 'Applications from personal account'. A search bar and a 'Add filters' button are also present. The main table lists 9 applications found, with columns for 'Display name', 'Application (client) ID', 'Created on', and 'Certificates & secrets'. Filter options for 'Current', 'Expired', and 'Current' are shown on the right.

3. Enter the api name (**investran-web-api**) in the **Name** text field, select the supported account type and click **Register** to register the application.

The screenshot shows the 'Register an application' form. The 'Name' field is filled with 'investran-web-api'. The 'Supported account types' section contains four options: 'Accounts in this organizational directory only (VBK Organization only - Single tenant)' (selected with a blue outline), 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)', and 'Personal Microsoft accounts only'. Below these is a 'Help me choose...' link. At the bottom, a note says 'By proceeding, you agree to the Microsoft Platform Policies' with a link. The 'Register' button is highlighted in blue.

- Note down client id and tenant id which needs to be used while accessing the api.

Essentials

- Display name: webapp
- Application (client) ID: [REDACTED]
- Object ID: [REDACTED]
- Directory (tenant) ID: [REDACTED]
- Supported account types: My organization only

Client credentials
[Add a certificate or secret](#)

Redirect URIs
[Add a Redirect URI](#)

Application ID URI
[Add an Application ID URI](#)

Managed application in local directory
[webapp](#)

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

- In the **Expose an API** screen, click **+ Add a scope**.

- Enter the scope name and switch the toggle to **Admins and users**. In addition, specify the details in mandatory text field and switch the toggle to **Enable** state.

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protected by the API. An application that requires access to parts of this API can request that a user or admin consent to one or more of these.

Adding a scope here creates only delegated permissions. If you are looking to create application-only scopes, use 'App roles' and define app roles assignable to application type. [Go to App roles](#).

+ Add a scope

Scopes	Who can consent	Admin consent display ...	User consent display na...	State
[REDACTED]	Admins and users	[REDACTED]	[REDACTED]	[REDACTED]

Authorized client applications

Authorizing a client application indicates that this API trusts the application and users should not be asked to consent when the client calls this API.

+ Add a client application

Client Id	Scopes
[REDACTED]	No client applications have been authorized

Note: Ensure to take a copy of the Application ID URI which is available in the **Expose an API** screen and this can be changed.

API Access to Client Application

The following are the steps:

- Click on **App registration** and select the client app.

2. In the left pane, click **API permission** and click **+Add a new permission**.

3. Select **My APIs** and select APIs.

Home > VBK Organization > WebAppUserApis

WebAppUserApis | API permission

Search (Ctrl+ /) Refresh Add a permission API / Permissions name

Overview Quickstart Integration assistant

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions**
- Expose an API
- App roles
- Owners
- Roles and administrators

To view and manage per

Request API permissions

All APIs signed-in user.

Select permissions expand all

investran

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

Permission	Admin consent required
Permissions (1)	
investran-web-api	No

Add permissions Discard

4. Select **investran-web-api** permission and click **Add Permissions**.

User Access to Client Application

1. In the left pane, click **Owners** and click **+Add Owners**.
2. In the **Owners** screen, select a user that should have access to api.

WebAppUserApis | Owners

Search (Ctrl+ /) Add owners Remove owners Got feedback?

In addition to users with permission to manage any applications, the user

Name	Email
[Placeholder]	[Placeholder]

Selected items

No items selected

Select

Adding Tenant to Environment Configuration Service

The Environment Configuration Service provides authentication information to other services.

So, after completing the steps mentioned in **client's Azure AD** and **API Application**, you must add information to the Environment Configuration Service.

In order to do this, you must create a `.json` file named `client.domain.json` with the following content:

```

{
  "Domain": "client.domain.com",
  "Idp": {
    "IssuerUri": "https://login.microsoftonline.com/organizations/v2.0/",
    "ClientId": "<application id>", // for the app registration created in the previous step
    "ResponseType": "code",
    "Scope": "openid profile offline_access api://<api-client-id>/<scope-name>",
    // ideally it will called investra-web-api
    "ShowDebugInformation": true,
    "DummyClientSecret": "",
    "StrictDiscoveryDocumentValidation": false,
    "SkipIssuerCheck": true,
    "Theme": ""
  },
  "DataEnvironments": [
    {
      "Name": "DataEnvironmentName1",
      "Server": "SQLAlias",
      "Database": "ClientDatabaseName"
    },
    {
      "Name": "DataEnvironmentName2",
      "Server": "SQLAlias",
      "Database": "ClientDatabaseName2"
    }
  ]
}

```

```
}, {  
    "Name": "DataEnvironmentName3",  
    "Server": "SQLAlias",  
    "Database": "ClientDatabaseName3"  
}  
,  
  
"Authority": {  
    "IssuerUrl": "https://sts.windows.net/<tenant-id>/",  
    "DiscoveryUrl": "https://sts.windows.net/<tenant-id>/.well-known/openid-configuration",  
    "AllowImpersonation": "Yes", // only if client credential flow is supported in azure. Otherwise it will set to No  
    "AuthorityType": "azure"  
}  
}
```



Setting up Client Credential Flow for Authentication among Microservices

In order to authenticate with other services, certain services utilize the Client Credential Flow.

Note: It is mandatory for a user to have an active Azure AD subscription.

For the purpose of registering application in Azure AD, the client must get in touch the FIS team.

Register App for Client Credential Flow

There are some services that use Client Credential Flow to authenticate with other services. To achieve this, an app must be registered in Azure.

Next, if you wish to make the Environment Configuration Service available for all servers then you need to add the authority to it.

Note: It is mandatory for a user to have an active Azure account.

For the purpose of setting up the service application, the client must get in touch with the FIS team.

To archive this, you must create a .json file named `service.accounts.json` with the following content:

```
[  
  {  
    "IssuerUrl": "https://sts.windows.net/<tenantId>/" ,  
    "DiscoveryUrl": "https://sts.windows.net/<tenantId>/.well-known/openid-configuration" ,  
    "AllowImpersonation": "Yes" ,  
    "AuthorityType": "azure"  
  }  
]
```

This file must be located in the folder that will be mounted inside the Environment Configuration Containers that is under the path

/app/Data/Authorities.

Configure all Applications that use Client Credential Flow

API's and Tenant Information Service

All API's and Tenant Information Service use Client Credential Flow in order to authenticate against Environment Configuration Service, so in the `appSettings.json` file you must include the following configuration:

```
"EnvironmentConfiguration": {  
    "EnvironmentConfigurationServiceUri": "<environment configuration uri>",  
    "TokenEndpoint": "https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/token",  
    "ClientId": "<client id>",  
    "ClientSecret": "<client secret>",  
    "Scope": "api://<api application id>/.default",  
    "ClientCredentialStyle": "1"  
},
```

- **RS**

Following are the different types of reporting services:

- **RS Correspondence Generation Service**

Edit the RS Correspondence Generation Service `appsettings.json` file and set the following values:

```
{  
    "Kestrel": {  
        "Endpoints": {  
            "Http": {  
                "Url": "http://*:2200/"  
            }  
        }  
    },  
    "DatabaseSettings": {  
        "ConnectionStringFormat": "Server={0};Initial  
Catalog={1};Application Name={2}@Investran;Integrated Security=SSPI",  
        "SQLPort": "[SQL_PORT]"  
    },
```

```

    "FileSystemSettings": {
        "FilesRepositoryPath": "[REPOSITORY_PATH]" ,
        "CorrespondenceGenerationBaseFolder": "[GENERATION_FOLDER]" ,
        "MappedFilesRepositoryPath": "" ,
        "SourceOSDirectorySeparator": "\\",
        "TargetOSDirectorySeparator": "\\"
    },
    "CertificateSerial": "[CERTIFICATE_SERIAL_NUMBER]" ,
    "ReportWizardExecutionEndpointUri": "[REPORT_WIZARD_ENDPOINT_URL]" ,
    "ReportWizardExecutionPollingIterationsCount": 120,
    "ReportWizardExecutionPollingIntervalSeconds": 15 ,
    "Authentication": {
        "Uri": "[IDP_ENDPOINT_URL]" ,
        "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE] ,
        "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
    },
    "SqlServerHealthCheck": "[SQL_SERVER_FOR_HEALTH_CHECK]" ,
    "DatabaseHealthCheck": "[DATABASE_FOR_HEALTH_CHECK]" ,
    "ConnectionStringTemplate": "Server={Server};Initial Catalog={Database};Application Name={SqlUserId}@Investran;Integrated Security=SSPI"
}

```

- **RS Export Service**

Edit the RS Export Service `appsettings.json` file and set the following values:

```

{
    "Logging": {
        "LogLevel": {
            "Default": "Debug",
            "Microsoft": "Debug",
            "Microsoft.Hosting.Lifetime": "Debug"
        }
    },
    "DbConnectionConfig": {
        "User": "",
        "Password": ""
    },
    "EnvironmentURL": {
        "ReportingExchangeApiBaseUrl": "[REPORTING_SERVICE_WEB_SERVICE]"
    },

```

```

        "ReportWizardApiBaseUrl": "[REPORT_WIZARD_WEBSERVICE]" ,
        "InvestranEndPoint": "[INVESTTRAN_URL]" ,
        "AuthenticationUri": "[INVESTTRAN_AUTHENTICATION_WEB_SERVICE]" ,
        "ReportingExchangeWebApiCertificateSerialNumber": "[CERTIFICATE_
SERIAL_NUMBER]" ,
        "LogStackTrace": "true" ,
        "EnableTls12": "false" ,
        "SQLPort": "[SQL_PORT]" ,
        "ConnectionStringFormat": "Server={0};Initial
Catalog={1};Application Name={2}@Investran;User Id={Username} ;
Password={Password};Encrypt=false"
    },
    "Authentication": {
        "Uri": "[IDP_ENDPOINT_URL]" ,
        "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE] ,
        "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
    },
    "LoadBalancingNodes": "[LOAD_BALANCE_NODE]"
}

```

- **RS Import Service**

Edit the RS Import Service `appsettings.json` file and set the following values:

```

{
    "Logging": {
        "LogLevel": {
            "Default": "Debug",
            "Microsoft": "Debug",
            "Microsoft.Hosting.Lifetime": "Debug"
        }
    },
    "DbConnectionConfig": {
        "User": "",
        "Password": ""
    },
    "EnvironmentURL": {
        "ReportingExchangeApiBaseUrl": "[REPORTING_SERVICE_WEB_
SERVICE]" ,
        "ReportWizardApiBaseUrl": "[REPORT_WIZARD_WEBSERVICE]" ,
        "InvestranEndPoint": "[INVESTTRAN_URL]" ,
        "AuthenticationUri": "[INVESTTRAN_AUTHENTICATION_WEB_SERVICE]" ,

```

```

        "ReportingExchangeWebApiCertificateSerialNumber": "[CERTIFICATE_
SERIAL_NUMBER]" ,
        "LogStackTrace": "true",
        "EnableTls12": "false",
        "SQLPort": "[SQL_PORT]" ,
        "ConnectionStringFormat": "Server={0};Initial
Catalog={1};Application Name={2}@Investran;User Id={Username} ;
Password={Password};Encrypt=false"
    },
    "Authentication": {
        "Uri": "[IDP_ENDPOINT_URL]" ,
        "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE] ,
        "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
    },
    "LoadBalancingNodes": "[LOAD_BALANCE_NODE]"
}

```

- **Business Events**

Edit the Business Events Service `Sungard.Investran.Suite.BusinessEvents.WindowsService.exe.config` file and set the following values:

```

<appSettings configBuilders="Secrets">
    <add key="APIAbsolutePath" value="[APIAbsolutePath]" />
    <add key="ClientSettingsProvider.ServiceUri" value="" />
    <add key="ARServiceExecutablePath" value="[ARServiceExecutablePath]" />
    <add key="ARServiceInactivityTimeout" value="21600" />
    <add key="retryPolicyDelay" value="1000" />
    <add key="retryPolicyRetries" value="5" />
    <add key="AuthenticationUri" value="[IDP_ENDPOINT_URL]" />
    <add key="ClientCredentialStyle" value="[CLIENT_CREDENTIAL_STYLE]" />
    <add key="ClientScope" value="[CLIENT_CREDENTIAL_SCOPE]" />
    <add key="ReportWizardExecutionEndpointUri" value="[REPORT_WIZARD_
ENDPOINT_URL]" />
    <add key="ReportWizardExecutionPollingIterationsCount" value="120" />
    <add key="ReportWizardExecutionPollingIntervalMilliseconds"
value="1000" />
        <add key="ConnectionStringTemplate"
value="Server={Server};Database={Database};Trusted_
Connection=True;Encrypt=false;" />
</appSettings>

```

- **Batch Save Service**

Edit the Batch Save Service appSettings.json file and set the following values:

```
{  
    ....  
    "AuthenticationTokenRequestClientId": "<ClientId>",  
    "AuthenticationTokenRequestClientSecret": "<ClientSecret>",  
    "AuthenticationTokenRequestUri": "\"https://login.microsoftonline.com/<tenant-  
id>/oauth2/v2.0/token\"",  
    "AuthenticationTokenRequestScope": "api://<api application id>/.default",  
    "AuthenticationTokenRequestClientCredentialStyle": "1"  
    ....  
}
```



OKTA as IDP

OKTA is a secure Identity cloud linking to all your apps, logins, and devices.

This section lists the following:

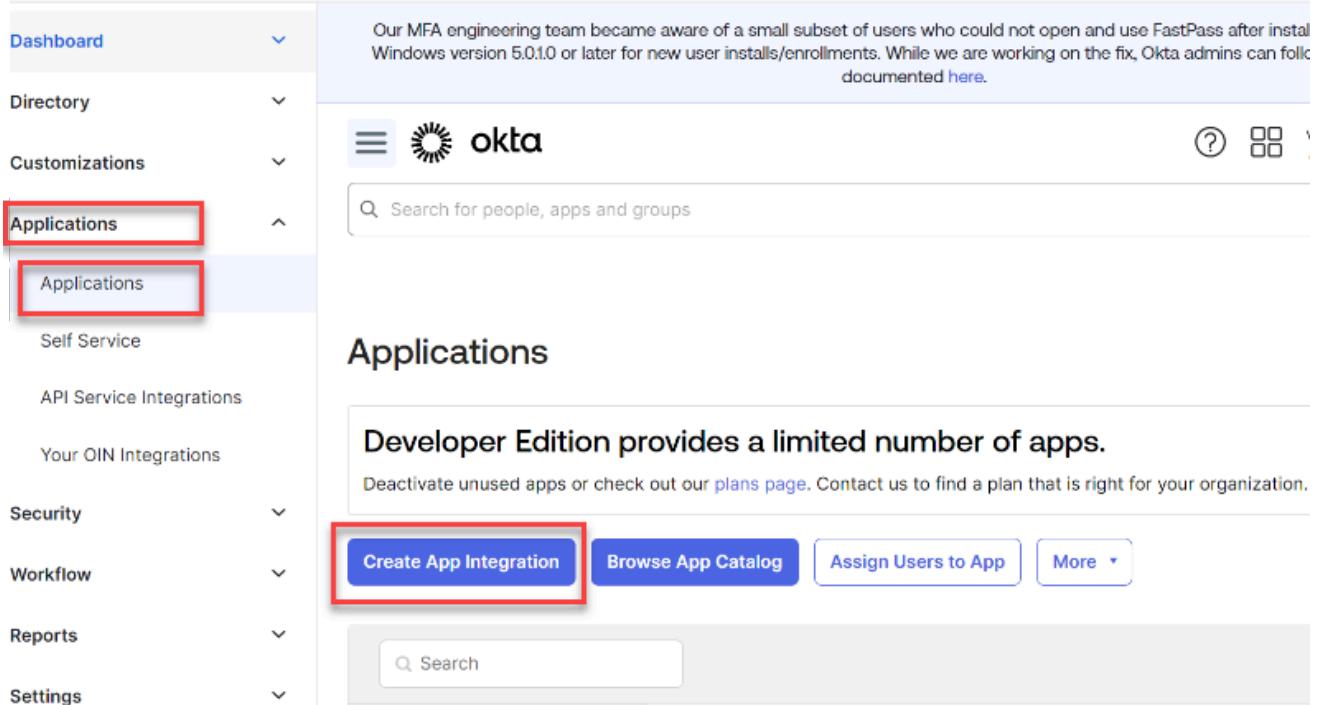
- [Configuring InvestranWeb to Authenticate with Client's Okta as IDP \(SAML 2.0 protocol\)](#)
- [Configuring Excel Add-In \(OIDC\), APIs Tenant Information Service, RW IDP Service and RS](#)
- [Setting up Client Credential Flow for Authentication among Microservices](#)

Configuring InvestranWeb to Authenticate with Client's Okta as IDP (SAML 2.0 protocol)

To setup clients to use their Okta as the IDP for the main PCS Web application (InvestranWeb), do the following:

Note: It is mandatory for a user to have an active subscription to create applications in OKTA.

1. Sign in to Okta and navigate to **Applications > Applications > Create App Integration**.



The screenshot shows the Okta Applications interface. On the left, there is a navigation sidebar with various menu items: Dashboard, Directory, Customizations, Applications (which is selected and highlighted with a red box), Applications (another item under Applications also highlighted with a red box), Self Service, API Service Integrations, Your OIN Integrations, Security, Workflow, Reports, and Settings. The main content area has a header with the Okta logo and a search bar. Below the header, the title 'Applications' is displayed. A message states 'Developer Edition provides a limited number of apps.' and includes a link to deactivate unused apps or contact for a plan. At the bottom of the main content area, there are four buttons: 'Create App Integration' (highlighted with a red box), 'Browse App Catalog', 'Assign Users to App', and 'More'. A search bar is also present at the bottom of the main content area.

2. In the **Create a New app integration** dialog box, select **SAML 2.0** and click **Next** to create SAML integration.



Create a new app integration

Sign-in method

[Learn More](#)

OIDC - OpenID Connect

Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.

SAML 2.0

XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.

SWA - Secure Web Authentication

Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.

API Services

Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

[Cancel](#)

[Next](#)

3. In the **Create SAML Integration** screen, in the **General Setting** section, provide the following details:
 - i. App name: Specify name of the app in the text field.
 - ii. App logo: Upload logo in the field.
 - iii. Single sign-on: Add sign-on URL in the text field. In addition, select the **Use this for Recipient URL and Destination URL** checkbox.
 - iv. Audience URI (SP Entity ID): Add a unique identifier for the application in the text field.

- Click **Show Advanced Settings** and map the following values as shown in the figure:



Search for people, apps and groups

[Hide Advanced Settings](#)

Response <small>?</small>	<input type="button" value="Unsigned"/>
Assertion Signature <small>?</small>	<input type="button" value="Signed"/>
Signature Algorithm <small>?</small>	<input type="button" value="RSA-SHA1"/>
Digest Algorithm <small>?</small>	<input type="button" value="SHA1"/>
Assertion Encryption <small>?</small>	<input type="button" value="Unencrypted"/>
Signature Certificate <small>?</small>	<input type="button" value="Browse files..."/>

[Enable Single Logout](#) [Allow application to initiate Single Logout](#)

5. In the **Attributes Statement**, add the **Name** and select the corresponding **Name format** and **Value**.



Search for people, apps and groups

[Hide Advanced Settings](#)

Response <small>?</small>	<input type="button" value="Unsigned"/>
Assertion Signature <small>?</small>	<input type="button" value="Signed"/>
Signature Algorithm <small>?</small>	<input type="button" value="RSA-SHA1"/>
Digest Algorithm <small>?</small>	<input type="button" value="SHA1"/>
Assertion Encryption <small>?</small>	<input type="button" value="Unencrypted"/>
Signature Certificate <small>?</small>	<input type="button" value="Browse files..."/>

[Enable Single Logout](#) [Allow application to initiate Single Logout](#)

6. Click on **Preview SAML Assertion** to review the changes and click **Next**.
7. In addition, select **This is an internal app that we have created** checkbox and click **Finish** to complete the SAML integration.

3 Help Okta Support understand how you configured this application

The optional questions below assist Okta Support in understanding your app integration.

App type This is an internal app that we have created

Previous **Finish**

Why are you asking
This form provides useful background information about your app. Thank you for appreciating it.

8. Go to **Application** and select the newly created application. The applications opens.

APPLICATIONS

Developer Edition provides a limited number of apps.
Deactivate unused apps or check out our [plans page](#). Contact us to find a plan that is right for your organization.

Create App Integration **Browse App Catalog** **Assign Users to App** **More**

Search	
STATUS	InvestranSAML
ACTIVE	1
INACTIVE	0
	Okta Admin Console
	Okta Browser Plugin

9. Sign on to the application and select **View SAML setup instructions** to view the certificate details.

Note: You can copy the single sign-on url and X509 Certificate that is used in Investran xml configuration file.

The screenshot shows the Okta application configuration interface. On the left, there's a sidebar with navigation links like Home, Applications, Groups, and Profiles. The main area has tabs for 'General' and 'Sign On'. Under 'General', there's a 'Metadata URL' section with a 'Copy' button and a 'More details' link. To the right, there's a note about prompting for manual username entry if password or profile push provisioning features are used. Below this is a 'SAML Setup' section with a note about Single Sign On not working until the app is trusted as an IdP, and a 'View SAML setup instructions' button.

10. To map to the application, navigate to **Assignments > Assign > Assign to People**. The application is created and SSO is configured and user is mapped.

The screenshot shows the 'Assign test to People' dialog box. It has a search bar at the top. Below it, there are tabs for 'General' and 'Sign On'. A large central area is labeled 'Assign' with a 'Done' button at the bottom right. On the left, there's a sidebar with 'Assign' selected, followed by 'Filters', 'People', and 'Groups'.



Configuring Excel Add-In (OIDC), APIs Tenant Information Service, RW IDP Service and RS

To setup Excel Add-In, APIs and Tenant Information Service using Okta as Identity Provider. The following are the steps:

Note: It is mandatory for a user to have an active Okta subscription.

1. Sign in to Okta subscription and navigate to Applications.
2. In the **Create a new app Integration** dialog box, select **Sign-in method as OIDC - OpenID Connect** and **Application type as Single-Page Application** and click **Next**.

Create a new app integration

X

Sign-in method

[Learn More ↗](#)

- OIDC - OpenID Connect**
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.
- SAML 2.0**
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.
- SWA - Secure Web Authentication**
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.
- API Services**
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

Application type

What kind of application are you trying to integrate with Okta?

Specifying an application type customizes your experience and provides the best configuration, SDK, and sample recommendations.

- Web Application**
Server-side applications where authentication and tokens are handled on the server (for example, Go, Java, ASP.Net, Node.js, PHP)
- Single-Page Application**
Single-page web applications that run in the browser where the client receives tokens (for example, Javascript, Angular, React, Vue)
- Native Application**
Desktop or mobile applications that run natively on a device and redirect users to a non-HTTP callback (for example, iOS, Android, React Native)

[Cancel](#) Next

3. In the **New Single-Page App Integration** dialog box, provide the following details:

- i. App integration name: Specify a name in the text field.
- ii. Logo: Upload a logo in the field.
- iii. Grant type: Select Authorization Code.
- iv. Sign-in redirect URIs: Add the sign-in URI
- v. Sign-out redirect URIs: Add the sign-out URI

New Single-Page App Integration

General Settings

App integration name

Logo (Optional) 

Grant type Client acting on behalf of a user
[Learn More](#)

Authorization Code
 Refresh Token
 Implicit (hybrid)

Sign-in redirect URIs Allow wildcard * in sign-in URI redirect.
Okta sends the authentication response and ID token for the user's sign-in request to these URIs
[Learn More](#)


[+ Add URI](#)

Sign-out redirect URIs (Optional)
After your application contacts Okta to close the user session, Okta redirects the user to one of these URIs.


[+ Add URI](#)

4. Click **Save**.
5. Navigate to **Security > API > Add Authorization Server**. In the **Add Authorization Server** dialog box, provide the following details:
 - i. Name: Specify a name in the text field.
 - ii. Audience: Specify a name in the text field.
 - iii. Description: Provide the description in the text field.

Add Authorization Server

Name

Audience

Description

Save

Cancel

6. Click **Save**.
7. Go to **Scope** tab and click **Add Scope**. In the **Add Scope**, provide the following details:
 - i. Name: Specify the name in the text field.
 - ii. Display phrase: Specify a name in the text field.
 - iii. Description: Specify the description in the text field.

Add Scope

Name	<input type="text"/>
	For example: email
Display phrase <small>(?)</small>	<input type="text"/>
	For example: Access your email 58 characters remaining
Description <small>(?)</small>	<input type="text"/>
	For example: This allows you to use your email to login to the app
User consent <small>(?)</small>	<input checked="" type="radio"/> Implicit <input type="radio"/> Optional <input type="radio"/> Required
Block services	<input type="checkbox"/> Block services from requesting this scope
Default scope	<input type="checkbox"/> Set as a default scope
Metadata	<input type="checkbox"/> Include in public metadata
Create Cancel	

8. Go to **Claims** and in the **Add Claim** dialog box, provide the following details:

- i. Name: Specify the name in the text field.
- ii. Include in token type: Select a token type from the drop down list.
- iii. Value type: Select the value type from the drop down list.
- iv. Value: Specify the expression in the text field.
- v. Disable Claim: Select this option if required.
- vi. include in: Select any of the option basis on the users requirement.

Lets consider a scenario where Claim name is upn and entries for this claim is listed in the following table:

Field Name	Entry 1	Entry 2
Include In	Any scope	Any scope
Include in token	Access Token	Id token

Value Type	Expression	Expression
Expression	appuser.userName	appuser.userName
Disable Claim	Disabled	Disabled
Include In	Any scope	Any scope

Note: Both the entries that is Entry 1 and Entry 2 are for the same claim that is one for Access Token and other for Id Token.

Add Claim

Name

Include in token type

Value type

Value [Expression Language Reference](#)

Disable claim Disable claim

Include in Any scope
 The following scopes:

Save [Cancel](#)

9. Go to **Access Policies** and click **Add Policy**. In the **Add Policy** dialog box, specify a name and text in the **Name** and **Description** text field, and select **All clients** to **Assign to**.

Add Policy

Name

Description

Assign to

- All clients
 The following clients:

Create Policy

Cancel

10. Click **Create Policy** to create a policy.
11. Remove Default rule if exists.
12. In the **Add Rule**, provide the following entries:
 - i. Rule Name: Specify a name in the text field.
 - ii. Grant Type: Select Authorization Code checkbox.
 - iii. User is: Select a required option.
 - iv. Scopes requested: Select a required option.
 - v. Use this inline hook: Select an option from the drop down list.

Add Rule

Rule Name
Excel Add In

IF Grant type is
Client acting on behalf of itself
 Client Credentials
Client acting on behalf of a user
 Authorization Code
 Implicit (hybrid)
 Resource Owner Password
 SAML 2.0 Assertion
 Device Authorization
 Token Exchange
 Client-initiated backchannel authentication flow (CIBA)

AND User is
 Any user assigned the app
 Assigned the app and a member of one of the following:

AND Scopes requested
 Any scopes
 The following scopes:

investran-web-api x openid x profile x
email x offline_access x
OIDC default scopes

THEN Use this inline hook
None (disabled) ▾

AND Access token lifetime is
1 Hours ▾

13. Click **Save**.

Setting up Client Credential Flow for Authentication among Microservices

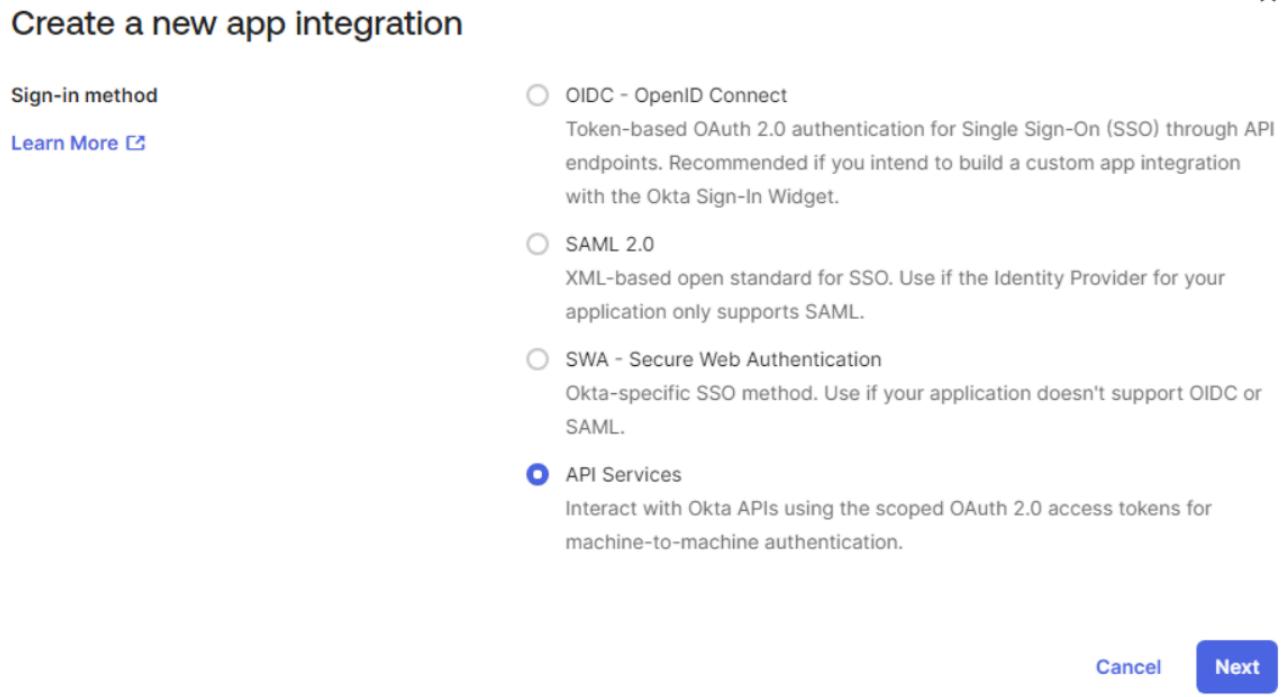
In order to authenticate with other services there are certain services that utilize the Client Credential Flow.

Register App for Client Credential Flow

Note: User must have an active Okta subscription.

The following are the steps to register in Okta:

1. Sign in to your Okta subscription and navigate to **Applications > Create a new app Integration**.
2. In the **Create a new app Integration** screen, select **API Services** and click **Next**.



Create a new app integration

Sign-in method

OIDC - OpenID Connect
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.

SAML 2.0
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.

SWA - Secure Web Authentication
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.

API Services
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

Cancel Next

3. In the **New API Services App Integration** dialog box, specify a name in the **App integration name** text field and click **Save**.

New API Services App Integration

General Settings

App integration name	My API Services App 1
Save Cancel	

4. Once the application is created, in the **General Settings** section unselect **Proof of possession**, and select **Grant type** as **Client Credentials**, and click **Save**.

General Settings Cancel

APPLICATION

App integration name	Test CCF
Application type	Service
Proof of possession	<input type="checkbox"/> Require Demonstrating Proof of Possession (DPoP) header in token requests
Grant type	<input checked="" type="checkbox"/> Client acting on behalf of itself <input checked="" type="checkbox"/> Client Credentials <input type="checkbox"/> Client acting on behalf of a user <input type="checkbox"/> Token Exchange
Save Cancel	

5. Navigate to **Security > API > Add Authorization Server**. In the **Add Authorization Server** dialog box, specify a name and text in the **Name, Audience and Description** text fields, and click **Save**.

Add Authorization Server

Name

Audience

Description

Save

Cancel

6. Go to **Scope** tab and click **Add Scope**. In the **Add Scope** dialog box, specify a name and text in the **Name**, **Display phrase**, and **Description** text fields. For example, specify name as "impersonation" in the text field.
7. In addition, select the User consent as **Implicit** and click **Create**.

Add Scope

Name

For example: email

Display phrase ?

For example: Access your email

58 characters remaining

Description ?

For example: This allows you to use your email to login to the app

User consent ?

Implicit

Optional

Required

Block services

Block services from requesting this scope

Default scope

Set as a default scope

Metadata

Include in public metadata

Create

Cancel

8. Navigate to **Access Policies > Add Policy**. In the **Add Policy** dialog box, specify a name and text in the **Name** and **Description** text fields and select **All clients** to **Assign to**.
9. Click **Create Policy** to add a new policy.

Add Policy

Name

Description

Assign to All clients
 The following clients:

Create Policy **Cancel**

10. If Default rule exists, then remove it.
11. In the **Add Rule** dialog box, specify a text in the **Rule Name** text field and select **Client Credentials** as **Grant type is**, **Any user assigned the app as User is** and **The following scopes:** as **Scopes requested**.

Add Rule

Rule Name
Client Credentials

IF Grant type is
 Client Credentials
 Client acting on behalf of itself
 Client acting on behalf of a user
 Authorization Code
 Implicit (hybrid)
 Resource Owner Password
 SAML 2.0 Assertion
 Device Authorization
 Token Exchange
 Client-initiated backchannel authentication flow (CIBA)

AND User is
 Any user assigned the app
 Assigned the app and a member of one of the following:
 Any scopes
 The following scopes:
 impersonation x investran-web-api x
 OIDC default scopes

THEN Use this inline hook
 None (disabled)

AND Access token lifetime is
 1 Hours

AND Refresh token lifetime is

Configure all applications that use client credential flow

API's and Tenant Information Service

All API's and Tenant Information Service use Client Credential Flow in order to authenticate against Environment Configuration Service, so in the `appSettings.json` file you must include the following configuration:

```

"EnvironmentConfiguration": {

    "EnvironmentConfigurationServiceUri": "<environment configuration uri>",

    "TokenEndpoint": "<default authorization server issuer uri>/oauth2/default/v1/
token",

    "ClientId": "<client id>",

    "ClientSecret": "<client secret>",

    "Scope": "investran-web-api impersonation",

    "ClientCredentialStyle": "0"

} ,

```

- **RS**

Following are the types of reporting services:

- **RS Correspondence Generation Service**

Edit the RS Correspondence Generation Service `appsettings.json` file and set the following values:

```

{
    "Kestrel": {
        "Endpoints": {
            "Http": {
                "Url": "http://*:2200/"
            }
        }
    },
    "DatabaseSettings": {
        "ConnectionStringFormat": "Server={0};Initial
Catalog={1};Application Name={2}@Investran;Integrated Security=SSPI",
        "SQLPort": "[SQL_PORT]"
    },
    "FileSystemSettings": {
        "FilesRepositoryPath": "[REPOSITORY_PATH]",
        "CorrespondenceGenerationBaseFolder": "[GENERATION_FOLDER]",
        "MappedFilesRepositoryPath": "",
        "SourceOSDirectorySeparator": "\\",
        "TargetOSDirectorySeparator": "\\"
    },
}

```

```

    "CertificateSerial": "[CERTIFICATE_SERIAL_NUMBER]",
    "ReportWizardExecutionEndpointUri": "[REPORT_WIZARD_ENDPOINT_URL]",
    "ReportWizardExecutionPollingIterationsCount": 120,
    "ReportWizardExecutionPollingIntervalSeconds": 15,
    "Authentication": {
        "Uri": "[IDP_ENDPOINT_URL]",
        "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE],
        "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
    },
    "SqlServerHealthCheck": "[SQL_SERVER_FOR_HEALTH_CHECK]",
    "DatabaseHealthCheck": "[DATABASE_FOR_HEALTH_CHECK]",
    "ConnectionStringTemplate": "Server={Server};Initial Catalog={Database};Application Name={SqlUserId}@Investran;Integrated Security=SSPI"
}

```

- **RS Export Service**

Edit the RS Export Service `appsettings.json` file and set the following values:

```

"Logging": {
    "LogLevel": {
        "Default": "Debug",
        "Microsoft": "Debug",
        "Microsoft.Hosting.Lifetime": "Debug"
    }
},
"DbConnectionConfig": {
    "User": "",
    "Password": ""
},
"EnvironmentURL": {
    "ReportingExchangeApiBaseUrl": "[REPORTING_SERVICE_WEB_SERVICE]",
    "ReportWizardApiBaseUrl": "[REPORT_WIZARD_WEBSERVICE]",
    "InvestranEndPoint": "[INVESTRAN_URL]",
    "AuthenticationUri": "[INVESTRAN_AUTHENTICATION_WEB_SERVICE]",
    "ReportingExchangeWebApiCertificateSerialNumber": "[CERTIFICATE_SERIAL_NUMBER]",
    "LogStackTrace": "true",
    "EnableTls12": "false",
    "SQLPort": "[SQL_PORT]"
}

```

```

        "ConnectionStringFormat": "Server={0};Initial
Catalog={1};Application Name={2}@Investran;User Id={Username};
Password={Password};Encrypt=false"
    },
    "Authentication": {
        "Uri": "[IDP_ENDPOINT_URL]",
        "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE],
        "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
    },
    "LoadBalancingNodes": "[LOAD_BALANCE_NODE]"
}

```

- **RS Import Service**

Edit the RS Import Service `appsettings.json` file and set the following values:

```

{
    "Logging": {
        "LogLevel": {
            "Default": "Debug",
            "Microsoft": "Debug",
            "Microsoft.Hosting.Lifetime": "Debug"
        }
    },
    "DbConnectionConfig": {
        "User": "",
        "Password": ""
    },
    "EnvironmentURL": {
        "ReportingExchangeApiBaseUrl": "[REPORTING_SERVICE_WEB_
SERVICE]",
        "ReportWizardApiBaseUrl": "[REPORT_WIZARD_WEBSERVICE]",
        "InvestranEndPoint": "[INVESTRAN_URL]",
        "AuthenticationUri": "[INVESTRAN_AUTHENTICATION_WEB_SERVICE]",
        "ReportingExchangeWebApiCertificateSerialNumber": "[CERTIFICATE_
SERIAL_NUMBER]",
        "LogStackTrace": "true",
        "EnableTls12": "false",
        "SQLPort": "[SQL_PORT]",
        "ConnectionStringFormat": "Server={0};Initial
Catalog={1};Application Name={2}@Investran;User Id={Username};
Password={Password};Encrypt=false"
    }
}

```

```

        },
        "Authentication": {
            "Uri": "[IDP_ENDPOINT_URL]",
            "ClientCredentialStyle": [CLIENT_CREDENTIAL_STYLE],
            "Scope": "[CLIENT_CREDENTIAL_SCOPE]"
        },
        "LoadBalancingNodes": "[LOAD_BALANCE_NODE]"
    }
}

```

- **Business Events**

Edit the Business Event `Sungard.Investran.Suite.BusinessEvents.WindowsService.exe.config` file and set the following values:

```

<appSettings configBuilders="Secrets">
    <add key="APIAbsolutePath" value="[APIAbsolutePath]" />
    <add key="ClientSettingsProvider.ServiceUri" value="" />
    <add key="ARServiceExecutablePath" value="[ARServiceExecutablePath]" />
    <add key="ARServiceInactivityTimeout" value="21600" />
    <add key="retryPolicyDelay" value="1000" />
    <add key="retryPolicyRetries" value="5" />
    <add key="AuthenticationUri" value="[IDP_ENDPOINT_URL]" />
    <add key="ClientCredentialStyle" value="[CLIENT_CREDENTIAL_STYLE]" />
    <add key="ClientScope" value="[CLIENT_CREDENTIAL_SCOPE]" />
    <add key="ReportWizardExecutionEndpointUri" value="[REPORT_WIZARD_ENDPOINT_URL]" />
    <add key="ReportWizardExecutionPollingIterationsCount" value="120" />
    <add key="ReportWizardExecutionPollingIntervalMilliseconds" value="1000" />
    <add key="ConnectionStringTemplate" value="Server={Server};Database={Database};Trusted_Connection=True;Encrypt=false;" />
</appSettings>

```

- **Batch Save Service**

Edit the Batch Save Service `appSettings.json` file and set the following values:

```
{  
....  
    "AuthenticationTokenRequestClientId": "<ClientId>",  
    "AuthenticationTokenRequestClientSecret": "<ClientSecret>",  
    "AuthenticationTokenRequestUri": "<default authorization server issuer  
uri>/oauth2/default/v1/token",  
    "AuthenticationTokenRequestScope": "investran-web-api impersonation",  
    "AuthenticationTokenRequestClientCredentialStyle": "0"  
....  
}
```