FIS PRIVATE CAPITAL SUITE

Authentication for Clients

Table of Contents

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:

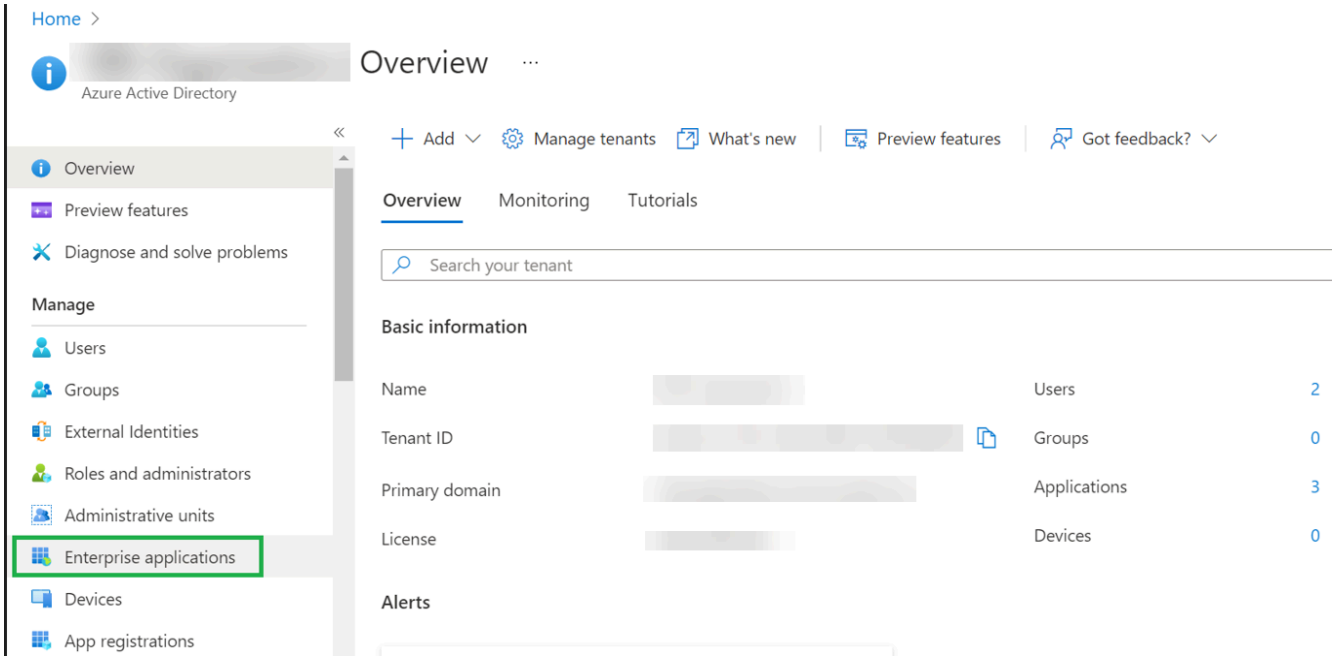
- Configuring InvestranWeb to Authenticate with Client's Azure AD as IDP (SAML 2.0 protocol)
- Configuring Excel Add-In (OIDC) APIs, Tenant Information Service, RW IDP Service and Reporting Service
- Configure Setting up Client Credential Flow for Authentication among Microservices

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

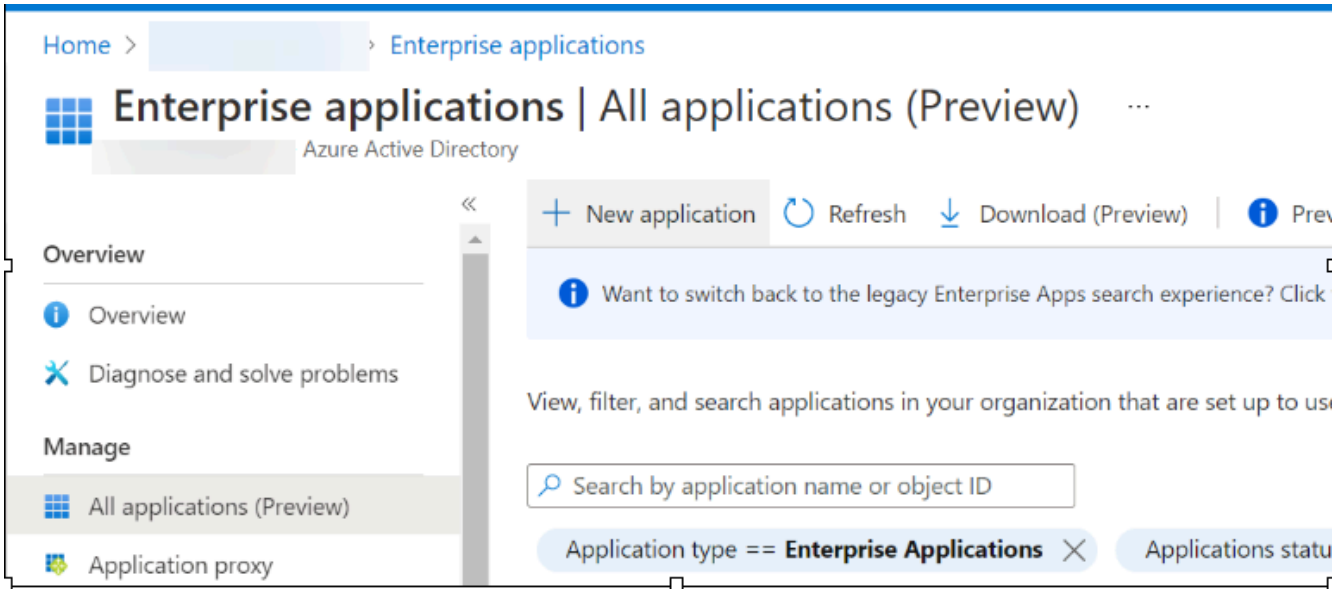
To setup clients to use their Azure AD tenant ID as the IDP for the main PCS Web application (InvestranWeb), do the following:

> 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.

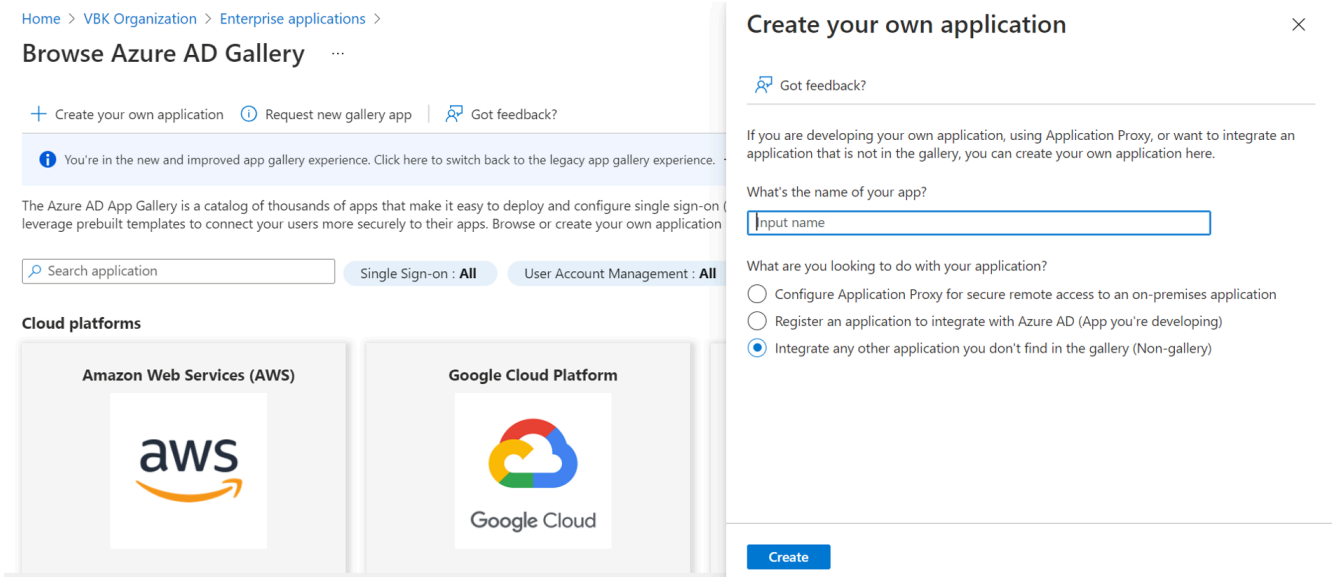
1. Sign in to Azure Portal and click **Azure Active Directory**.
2. Ensure that the Azure Active Directory is created and along with that the custom domain is available for the account.



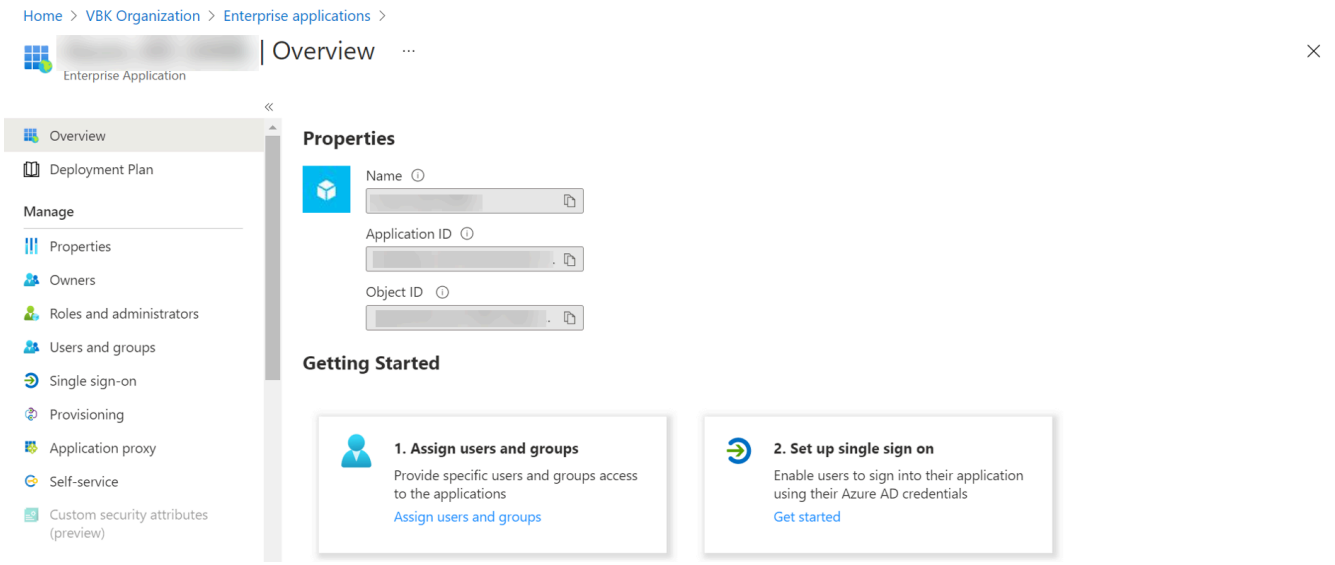
3. Select **Enterprise applications** > **All applications (Preview)** and click **+ New application**.



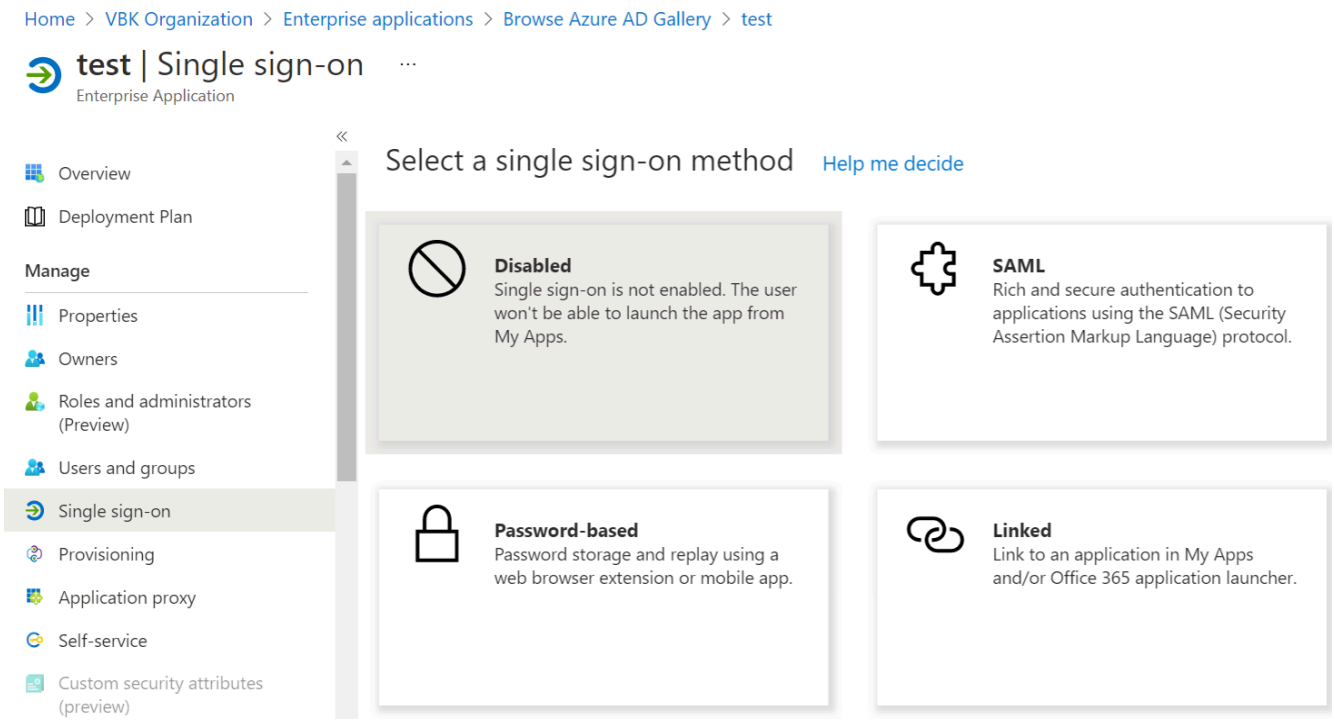
4. In the **Browse Azure AD Gallery** screen, click **+ Create Your Own Application**.
5. In the **Create your own application** screen, select **Integrate any other application you don't find in the gallery (Non-gallery)**.
6. Provide a name for your app in the text box and click **Create**. An app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



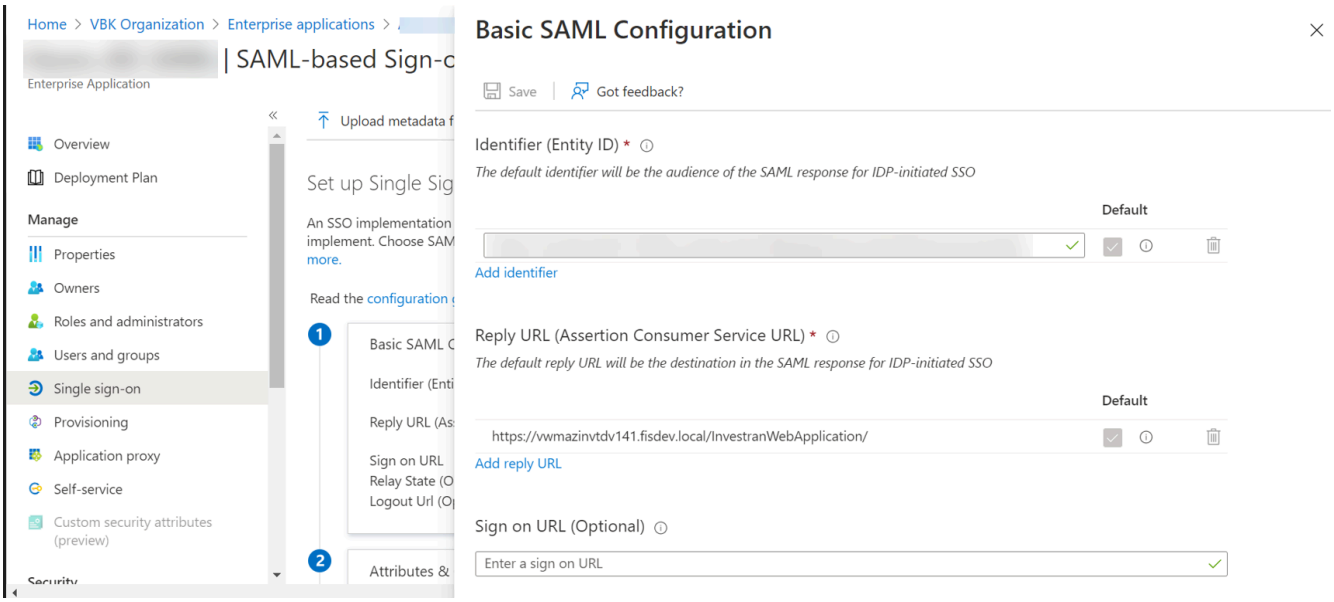
7. In the screen, add properties in the respective text fields and click **Setup single sign-on**.



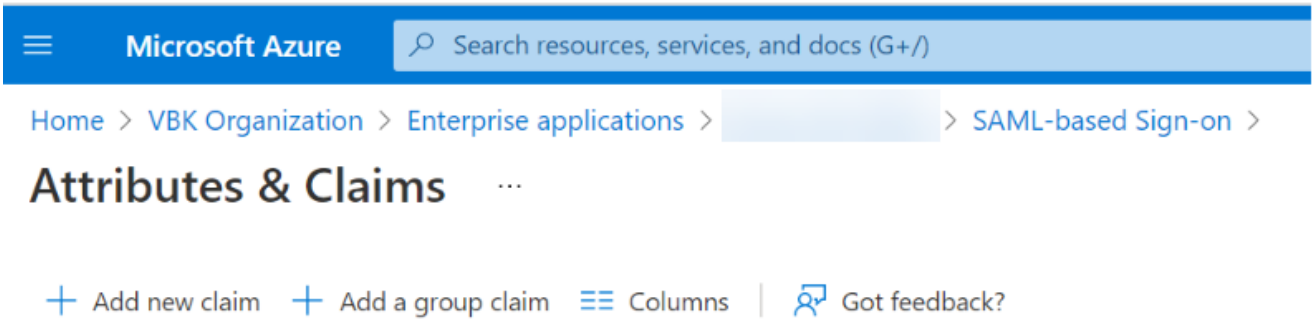
8. In the **Single sign-on** screen, select the **Disable** method.



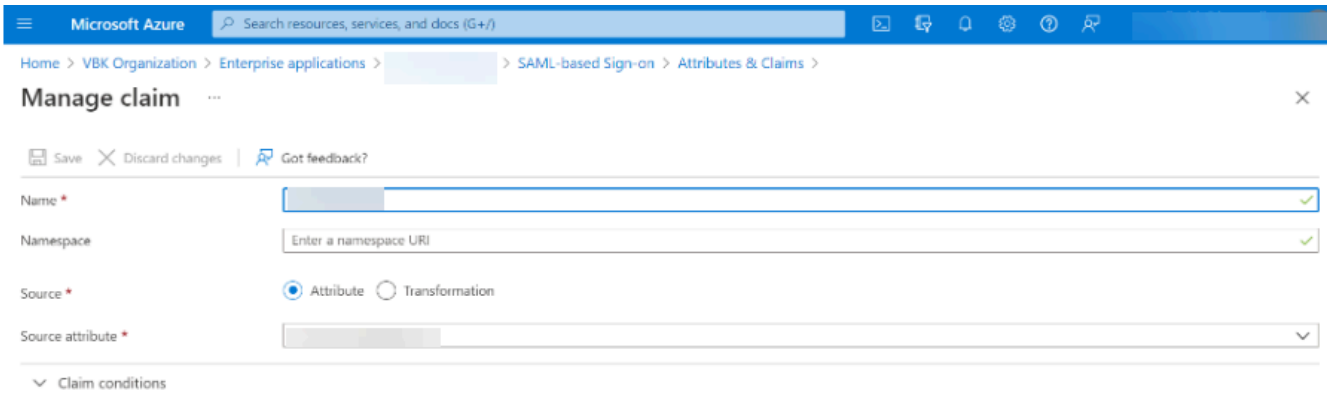
9. In the **SAML-based Sign-on** screen, click **Edit** in the **Basic SAML Configuration** section
10. In the **Basic SAML Configuration** screen, enter the Identifier and Reply URL fields.



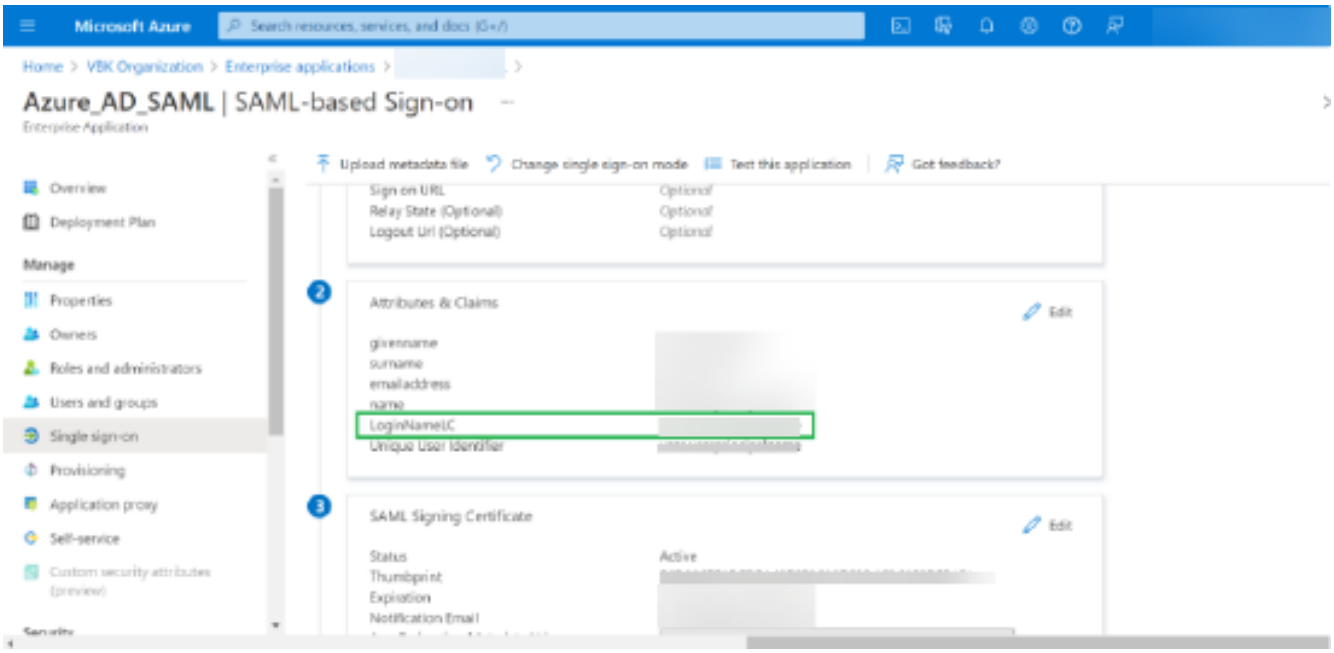
11. Save the configuration details and close the panel.
12. In the **Attributes and Claims** tab, click on the **Edit** option.



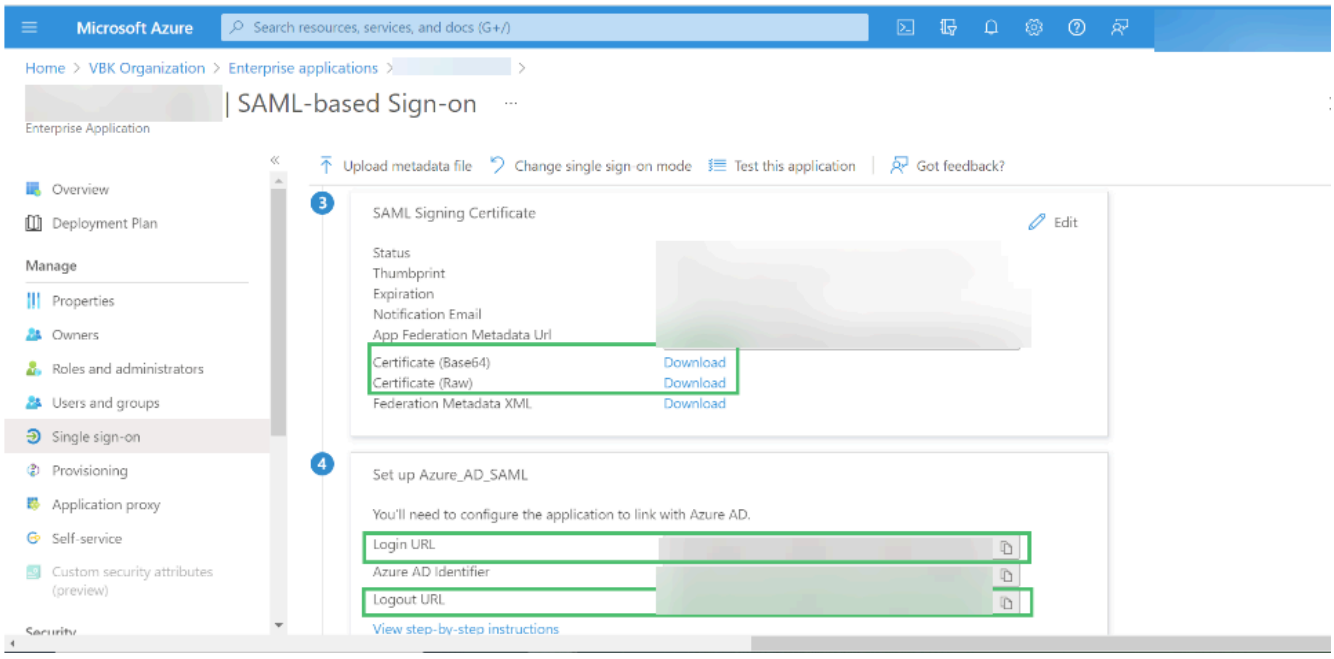
13. Click on **+ Add new claim** and provide details in the **Name** and **Source attributes** text field.



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



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.



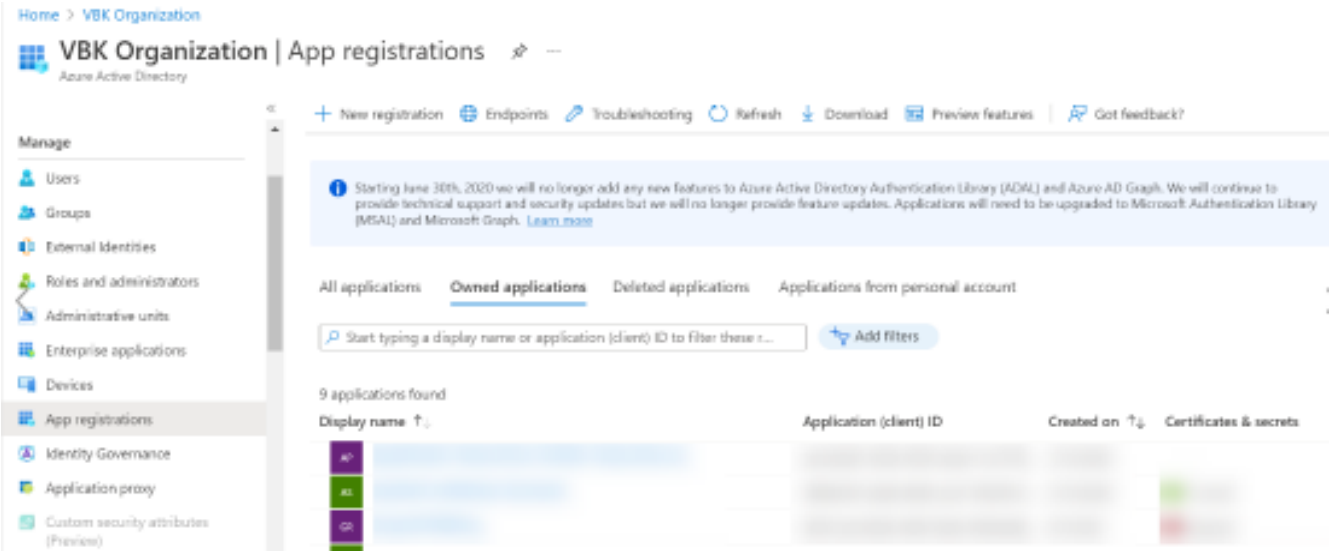
16. Now the Enterprise application is created and SSO is also configured.

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

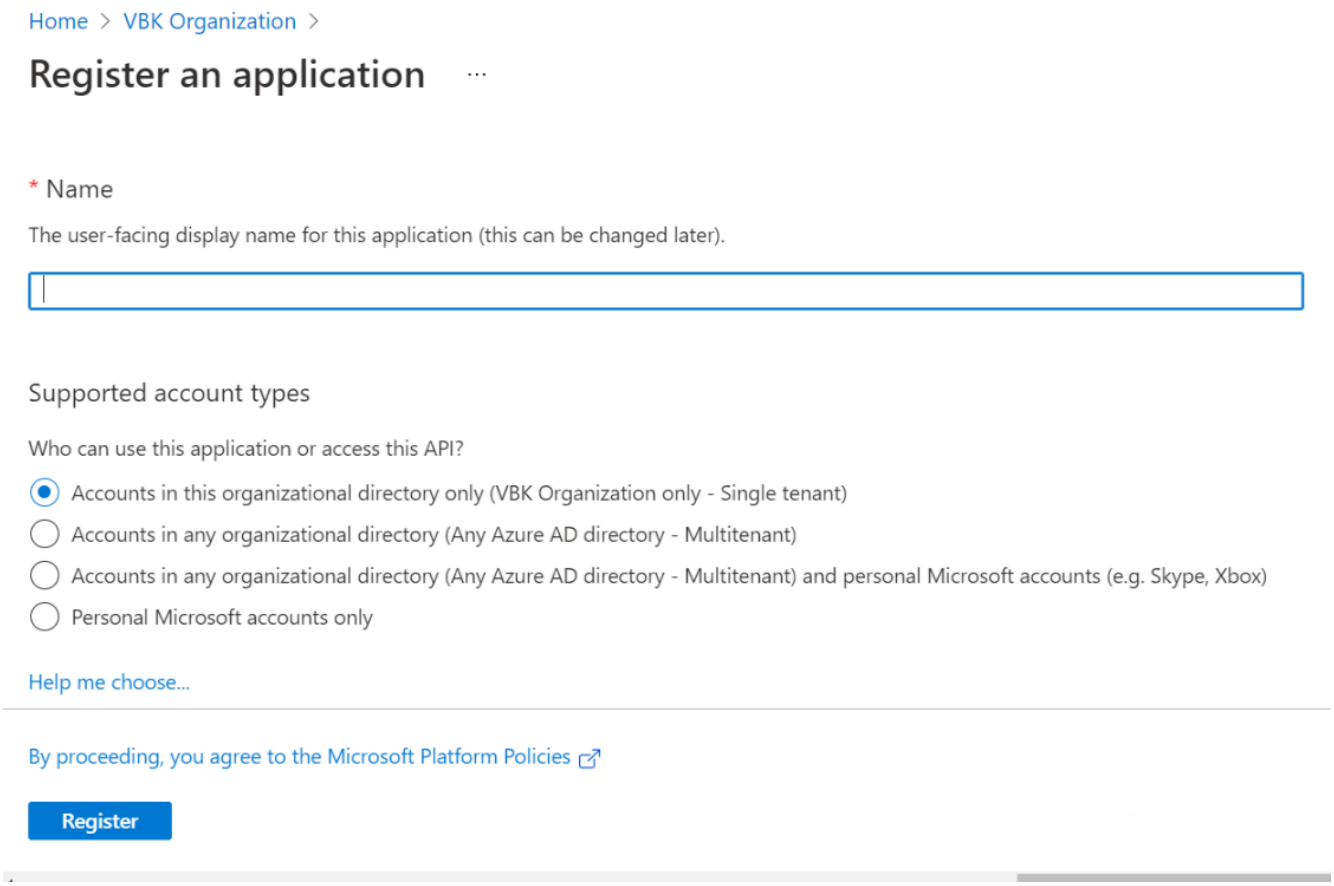
Setting up the client's Azure AD to Authenticate the Excel Add-In

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:

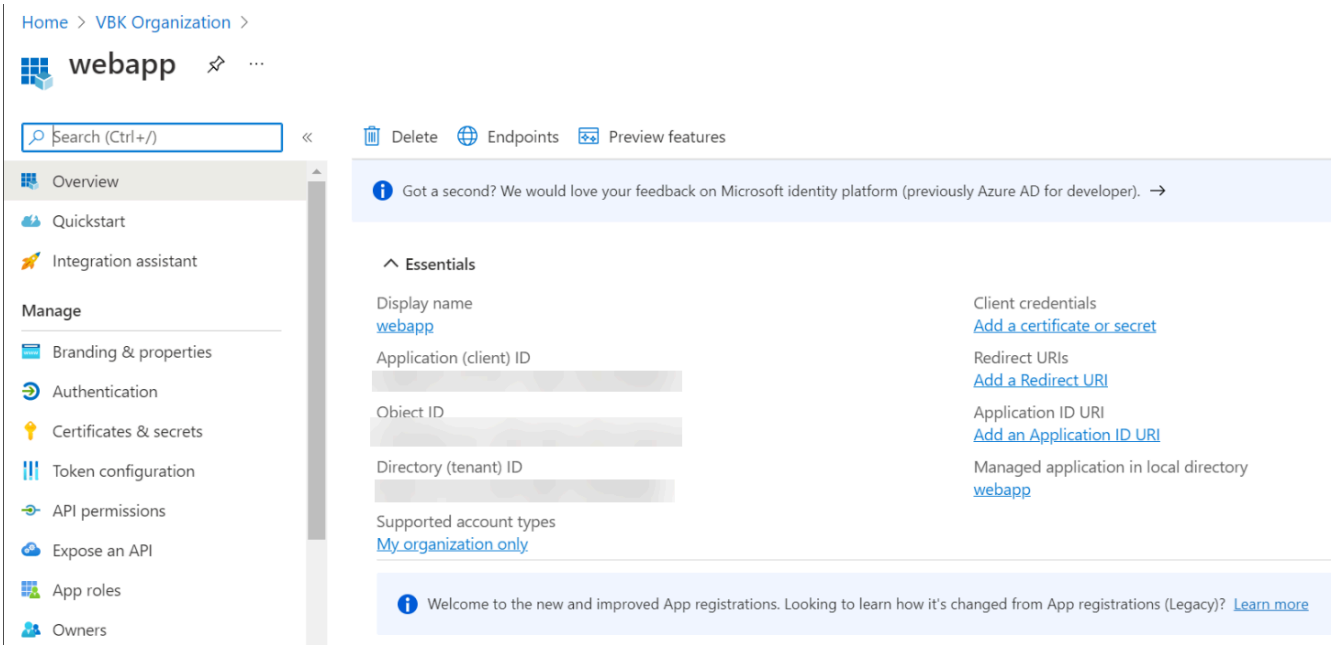
1. Sign in to the Azure portal and in the left navigation menu open Azure Active Directory.
2. Ensure that the Azure Active Directory is created and then the custom domain is available for the account. Select the **App registrations**, and click **+ New Registration.**



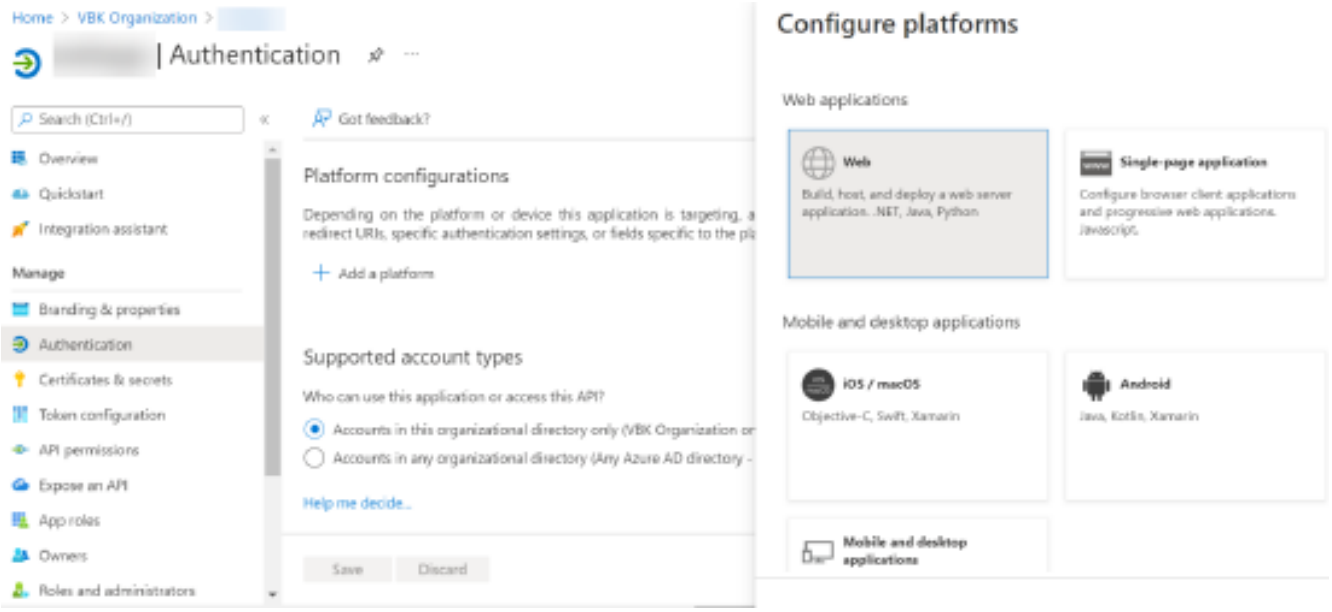
3. Enter the name of the application (**rw-excel-add-in)** in the **Name** text field, select the supported account type and click **Register** to register the application.



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

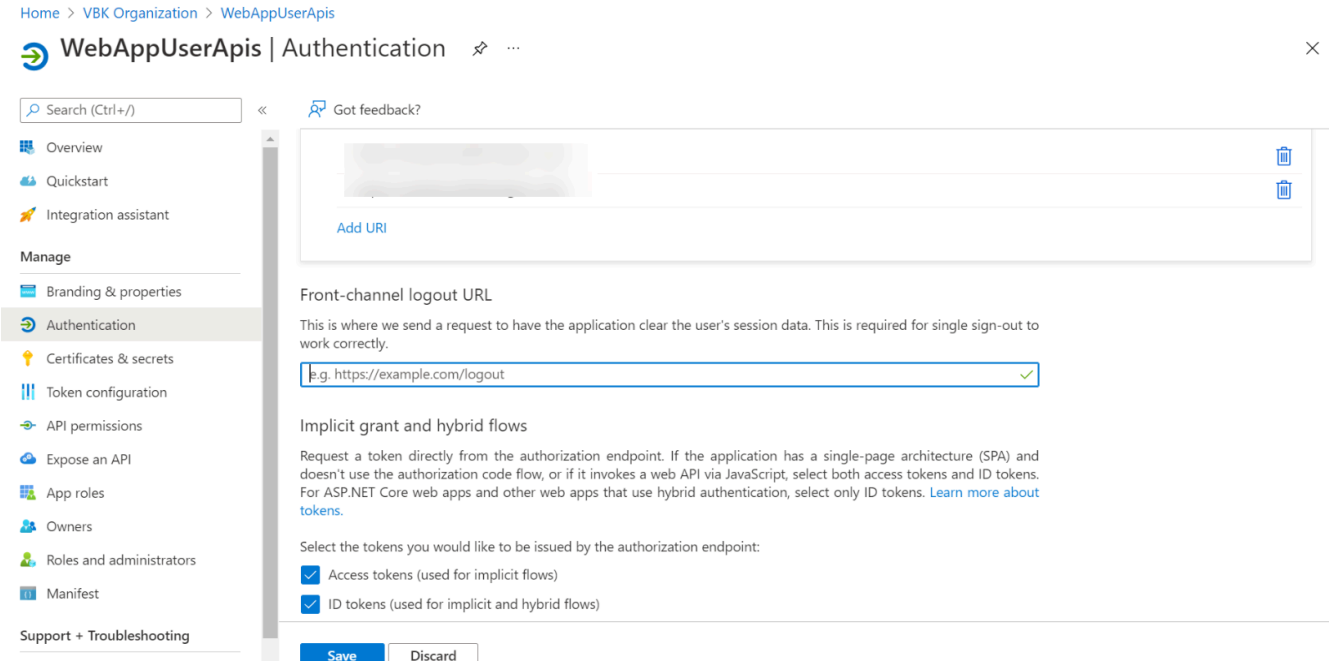


5. In the **Authentication** screen, click **+ Add a platform** and select **Single-page application**.



6. 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**.

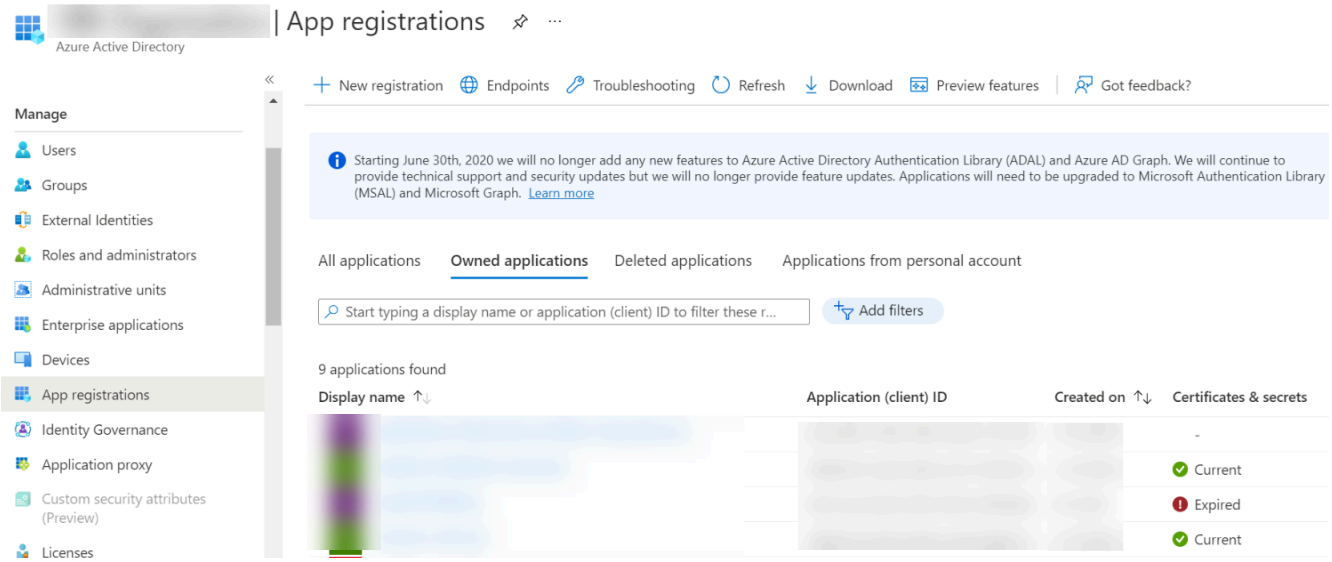
> Note: The Redirect uri must be RW Excel Add In uri+ /silent-refresh.html



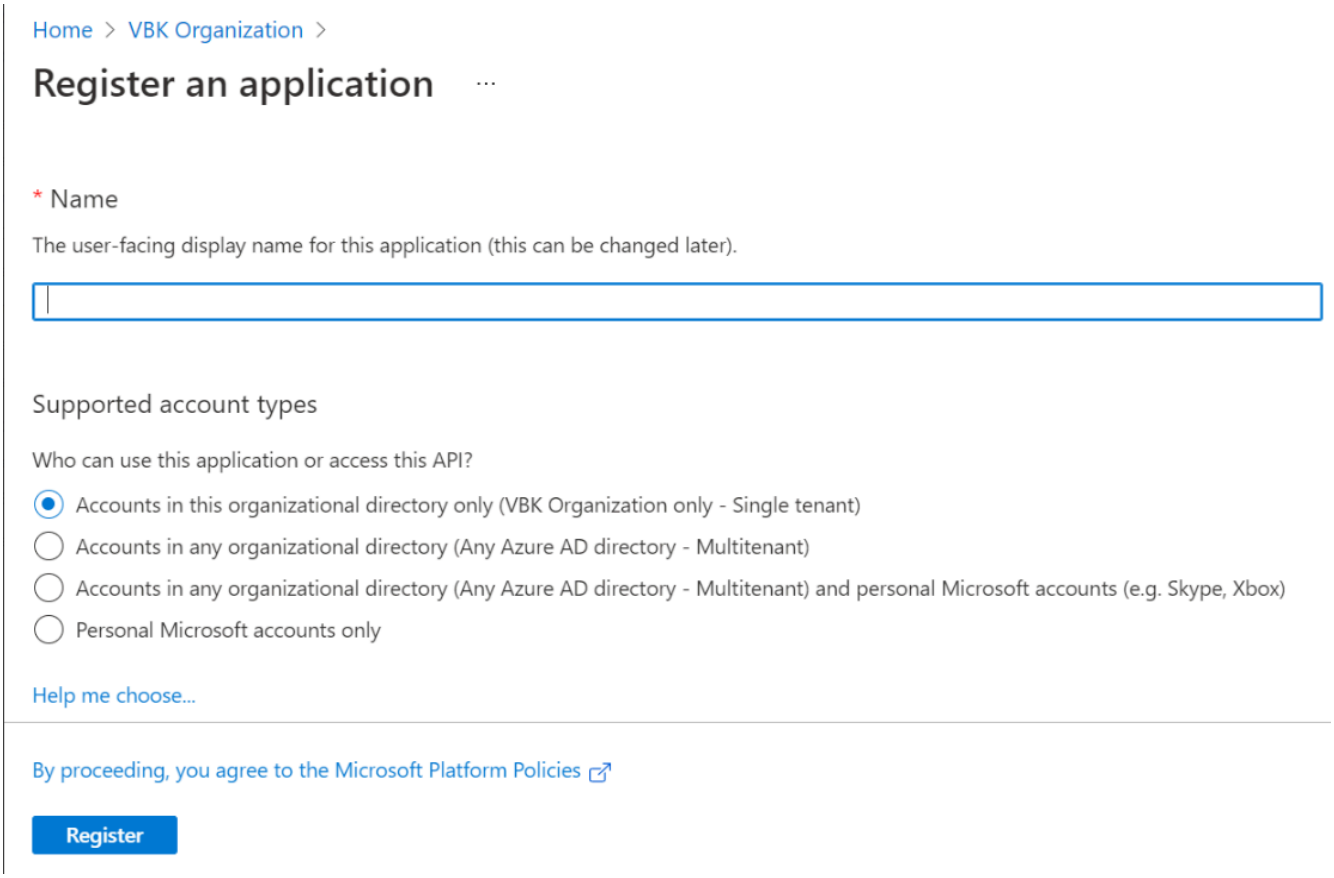
Creating API Application

The following are the steps to create the API Application:

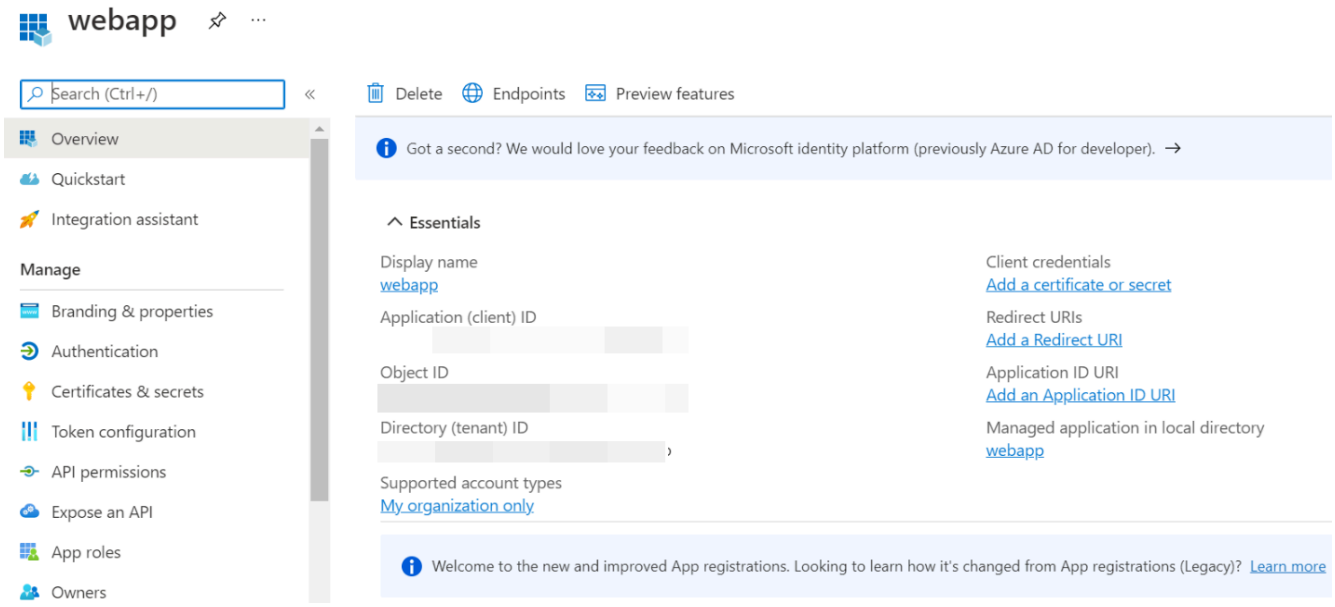
1. Sign in to Azure Portal and in the left navigation menu open Azure Active Directory.
2. Ensure the Azure Active Directory is created and the custom domain is available for the account. Select the **App Registrations** and click **+ New Registration**.



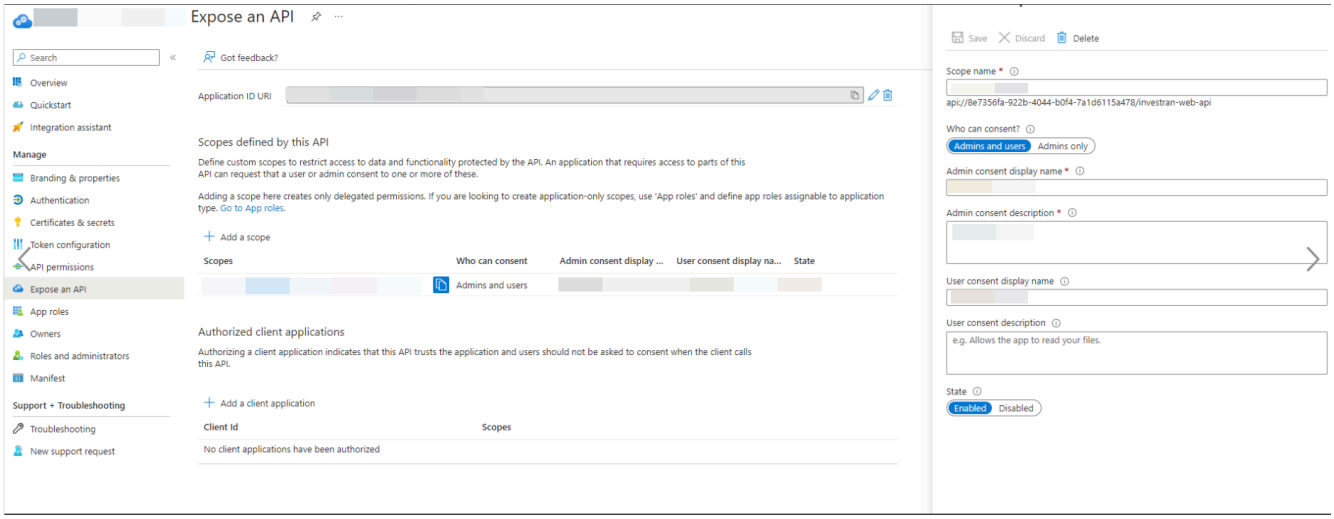
3. Enter the api name (**investran-web-api**) in the **Name** text field, select the supported account type and click **Register** to register the application.



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



5. In the **Expose an API** screen, click **+ Add a scope**.
6. 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.

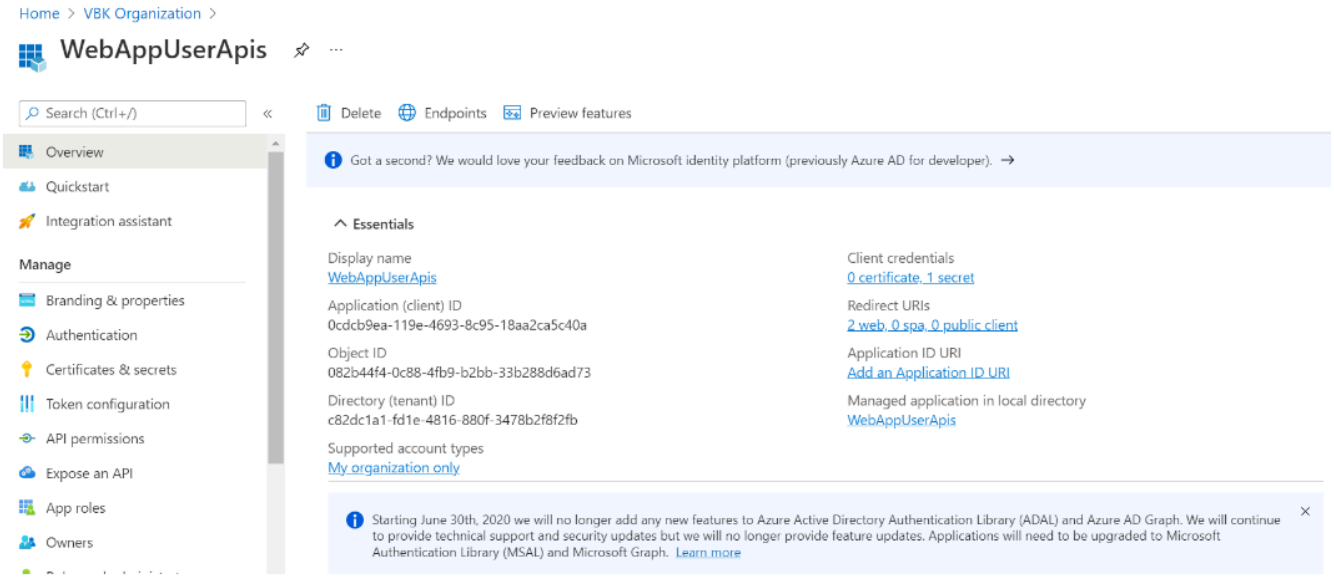


Note: Ensure to take a copy of the Application ID URI which is available in the **Expose an API** screen and this can be changed.

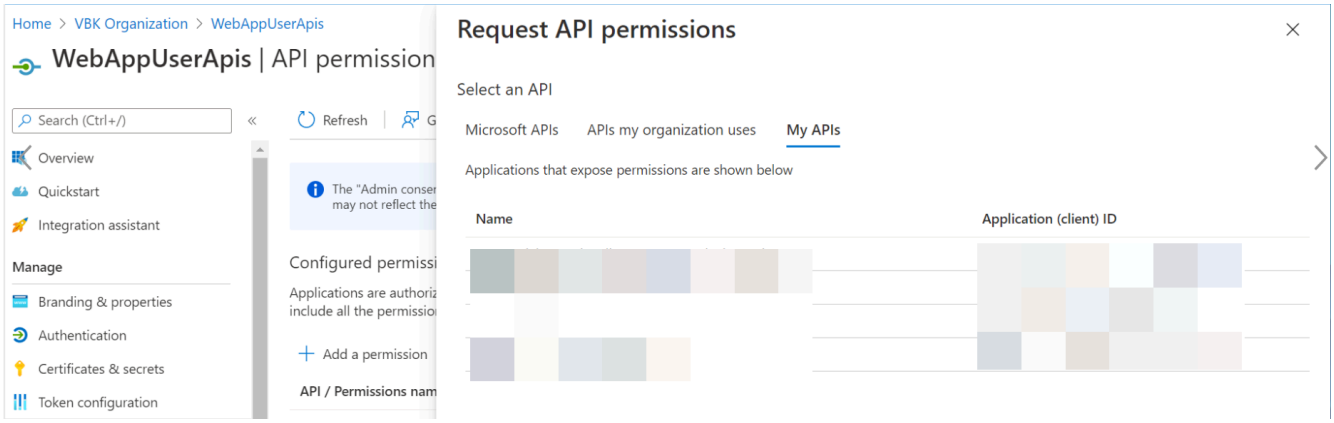
API Access to Client Application

The following are the steps:

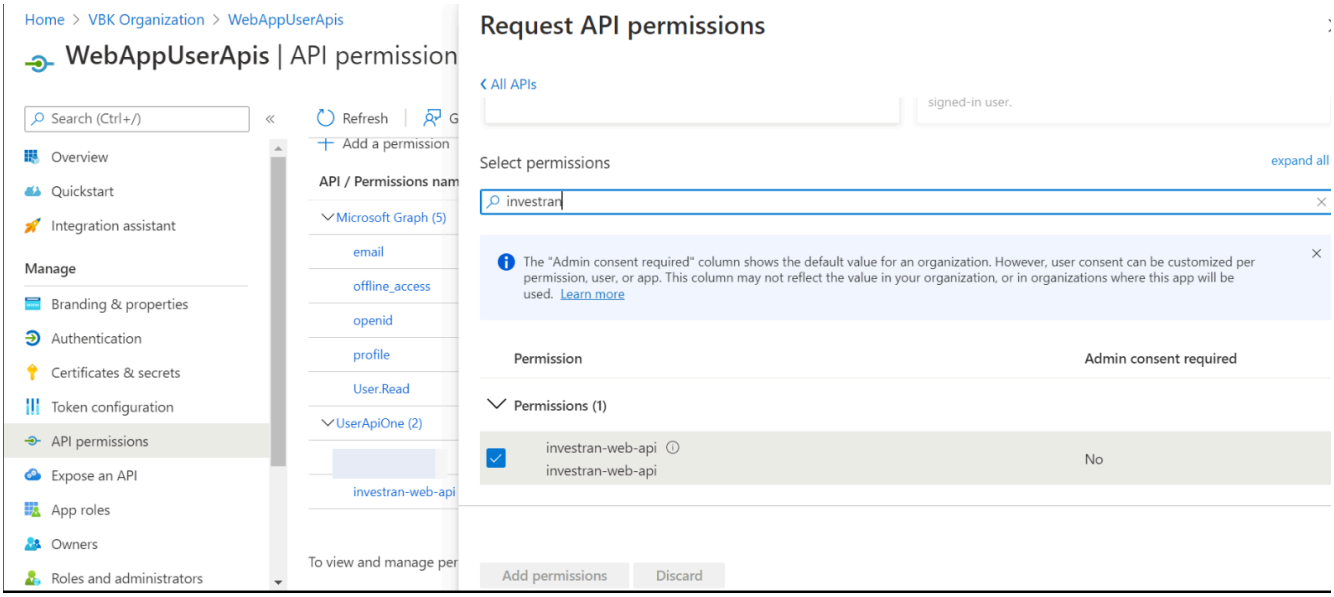
1. Click on **App registration** and select the client app.



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



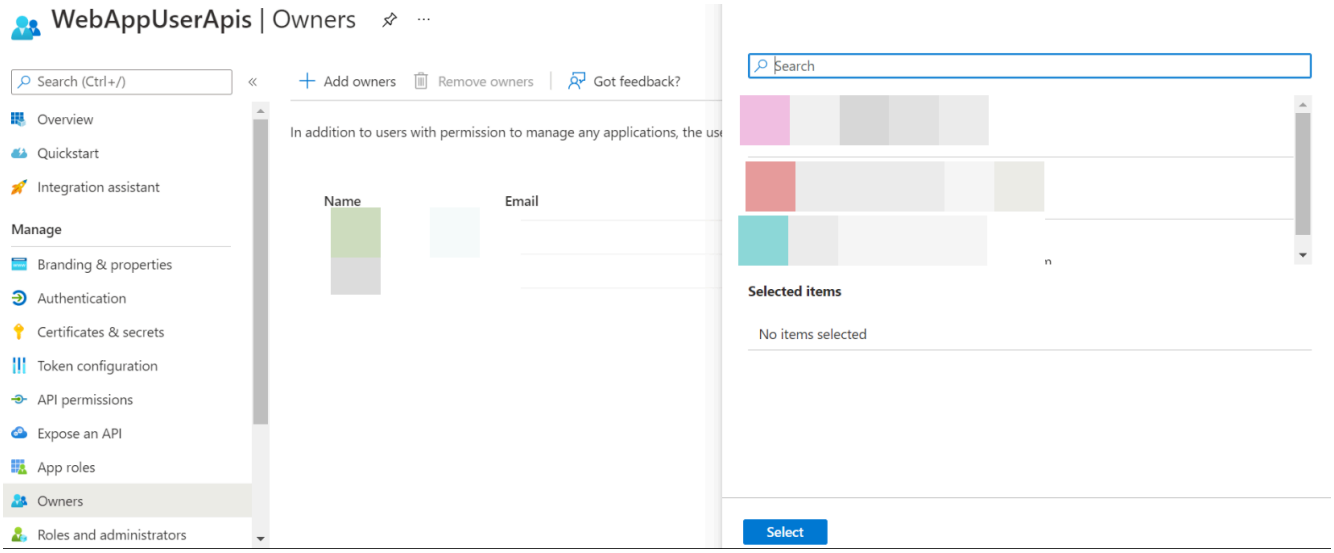
3. Select **My APIs** and select APIs.



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.



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/<tenent-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": "[CERTICATE_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 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/<tenent-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:

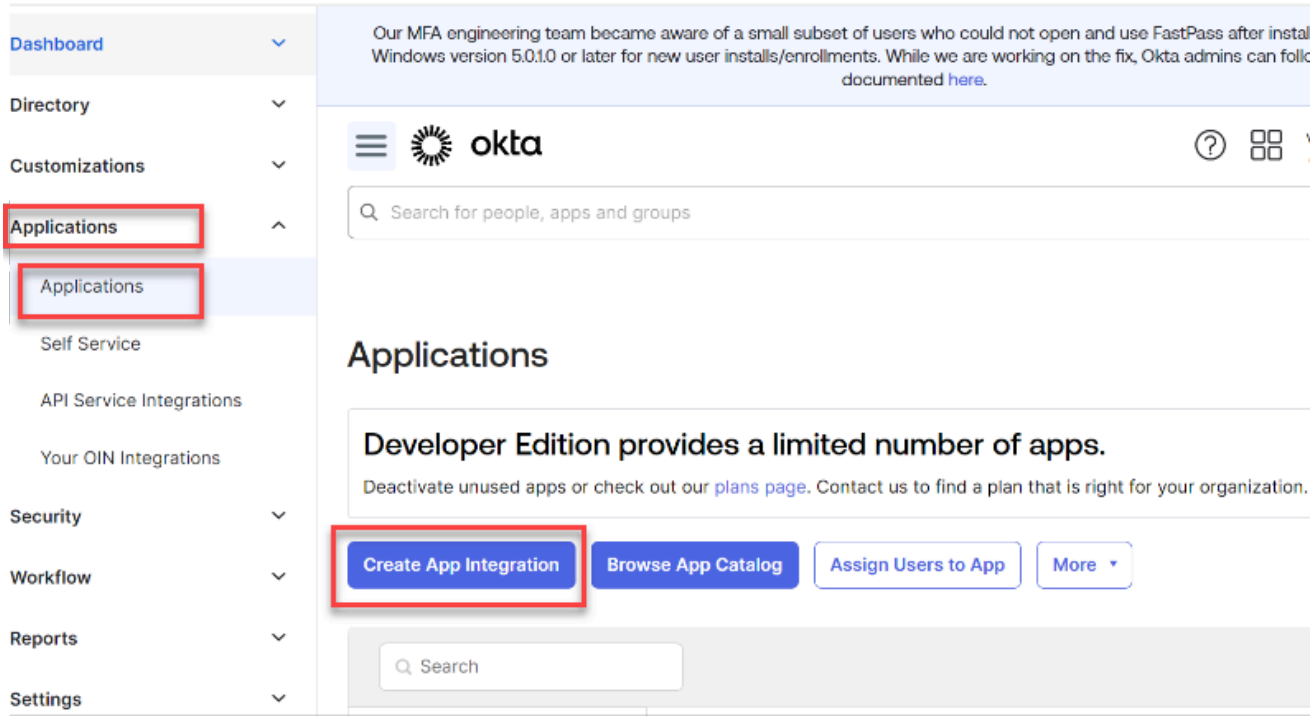
- Configuring InvestranWeb to Authenticate with Client's Okta as IDP (SAML 2.0 protocol)
- Configuring Excel Add-In (OIDC), APIs Tenant Information Service, RW IDP Service and RS
- Setting up Client Credential Flow for Authentication among Microservices

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

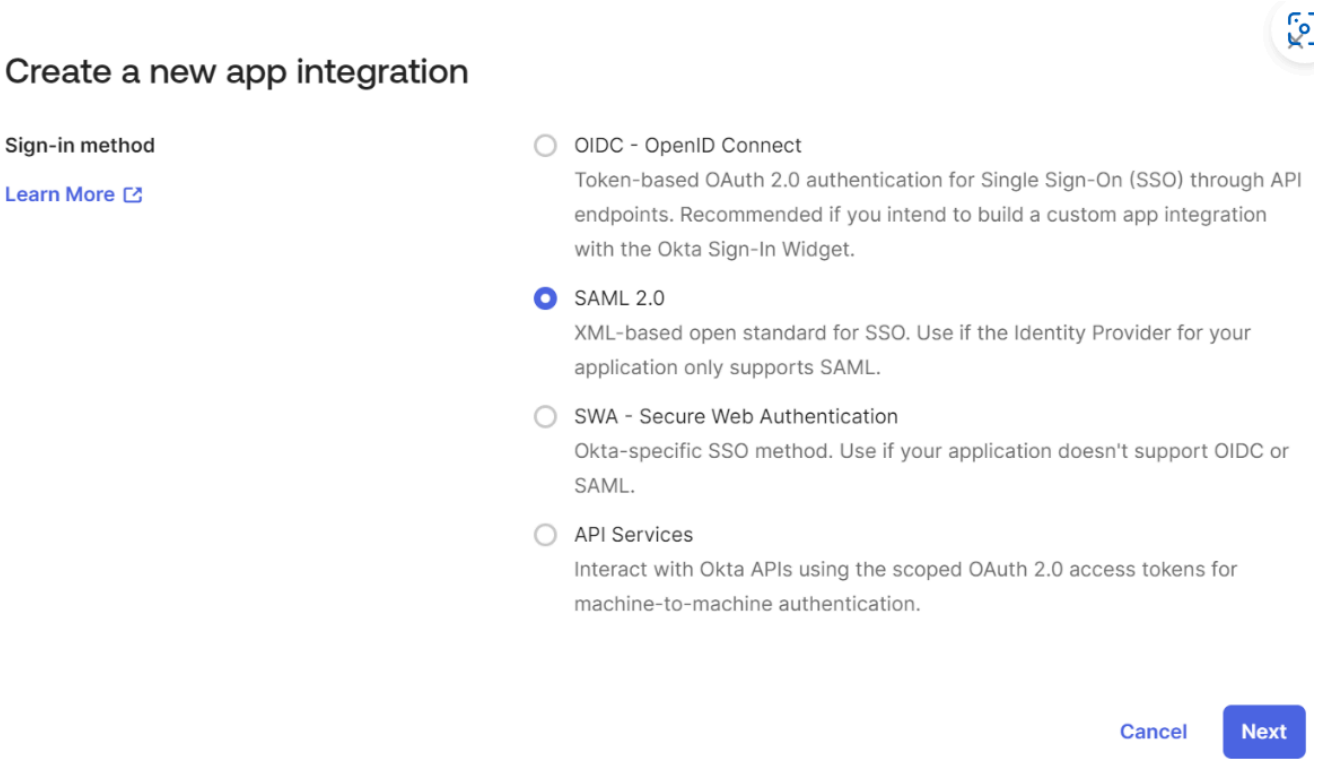
To setup clients to use their Okta as the IDP for the main PCS Web application (InvestranWeb), do the following:

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

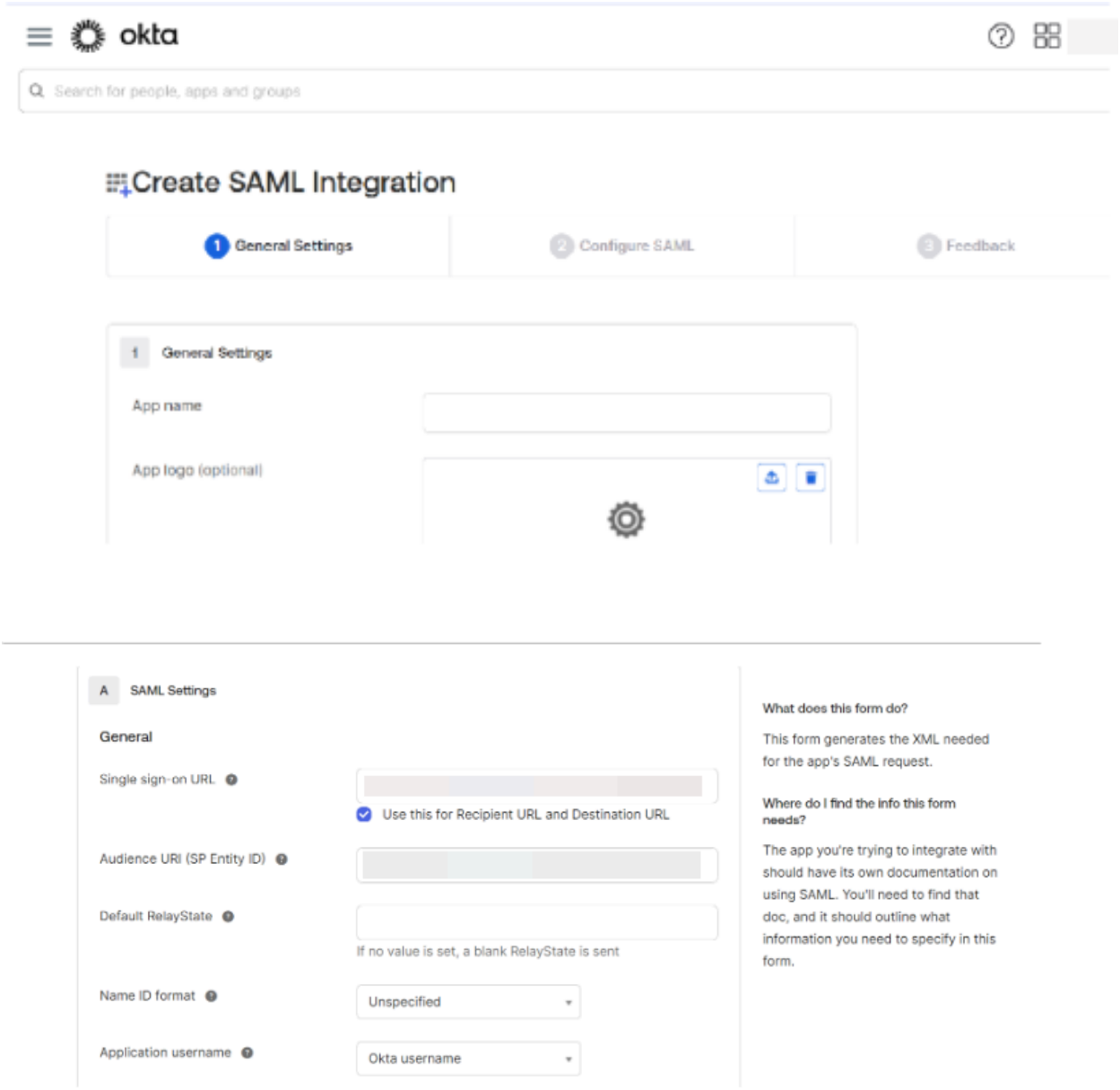
1. Sign in to Okta and navigate to **Applications** > **Applications** > **Create App Integration**.



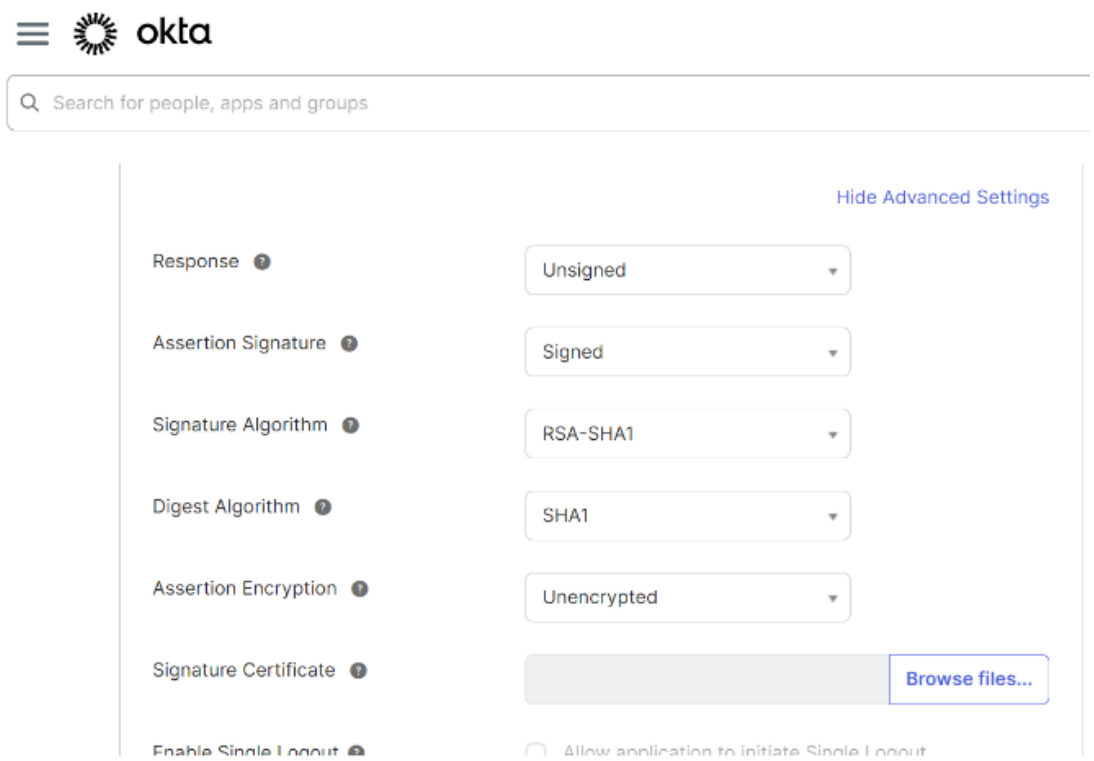
2. In the **Create a New app integration** dialog box, select **SAML 2.0** and click **Next** to create SAML integration.



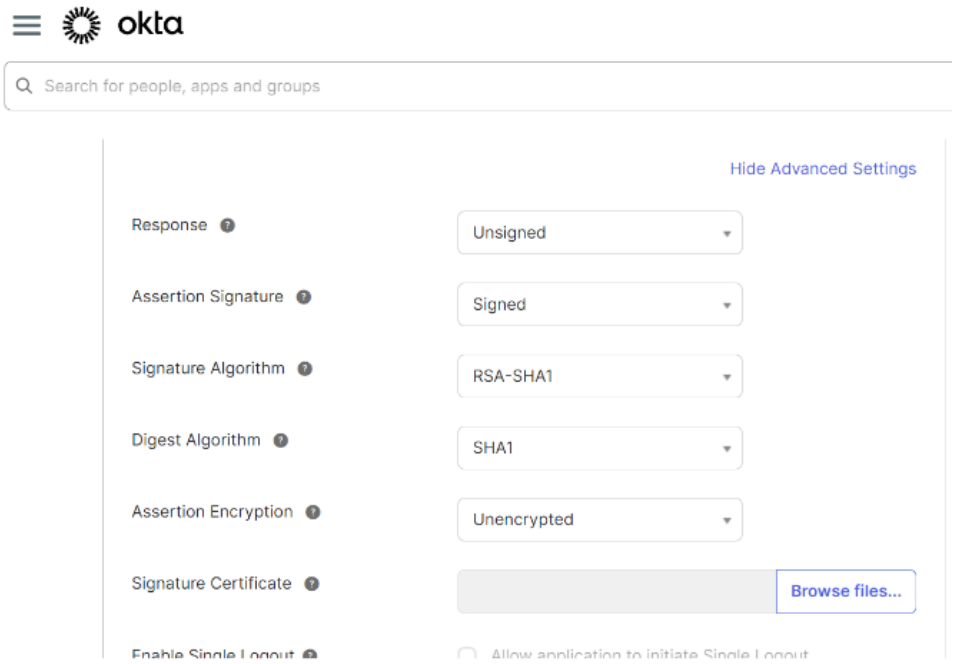
3. In the **Create SAML Integration** screen, in the **General Setting** section, provide the following details:
    i. App name: Specify name of the app in the text field.
    ii. App logo: Upload logo in the field.
    iii. Single sign-on: Add sign-on URL in the text field. In addition, select the **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.



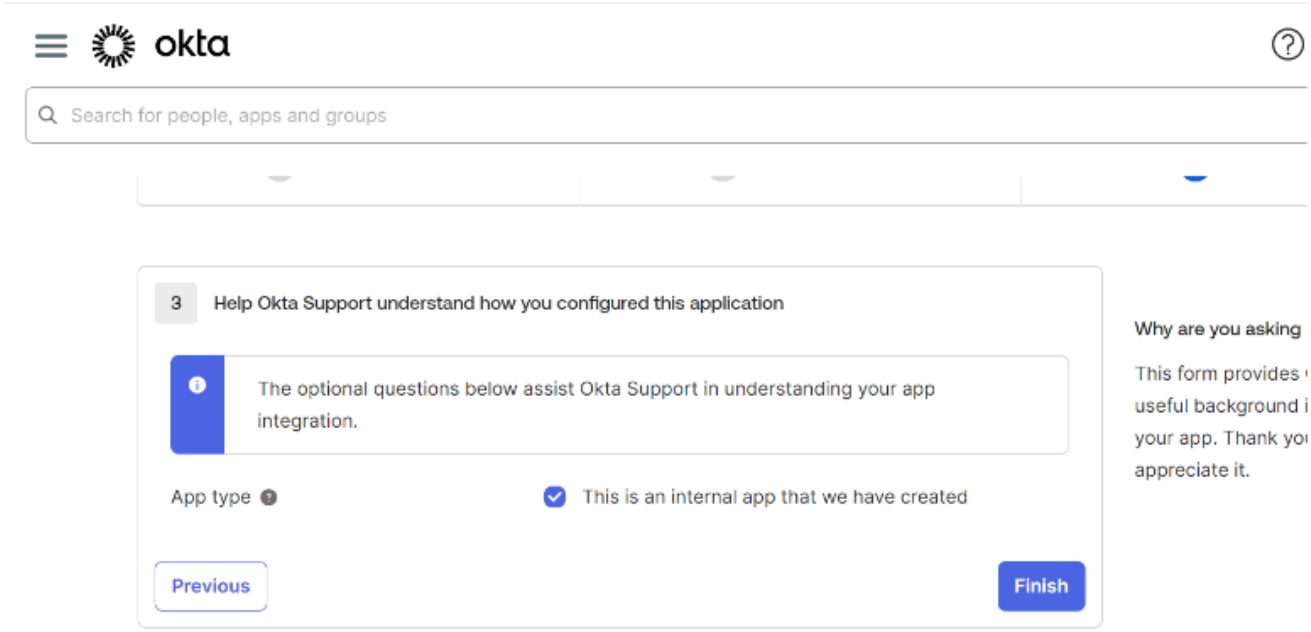
4. Click **Show Advanced Settings** and map the following values as shown in the figure:



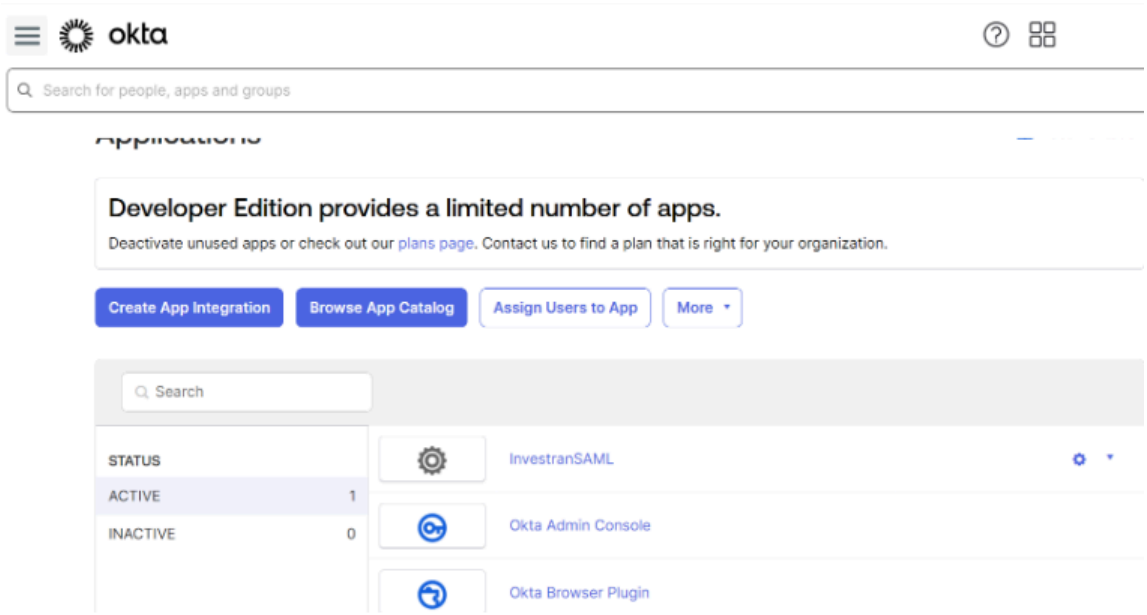
5. In the **Attributes Statement**, add the **Name** and select the corresponding **Name format** and **Value**.



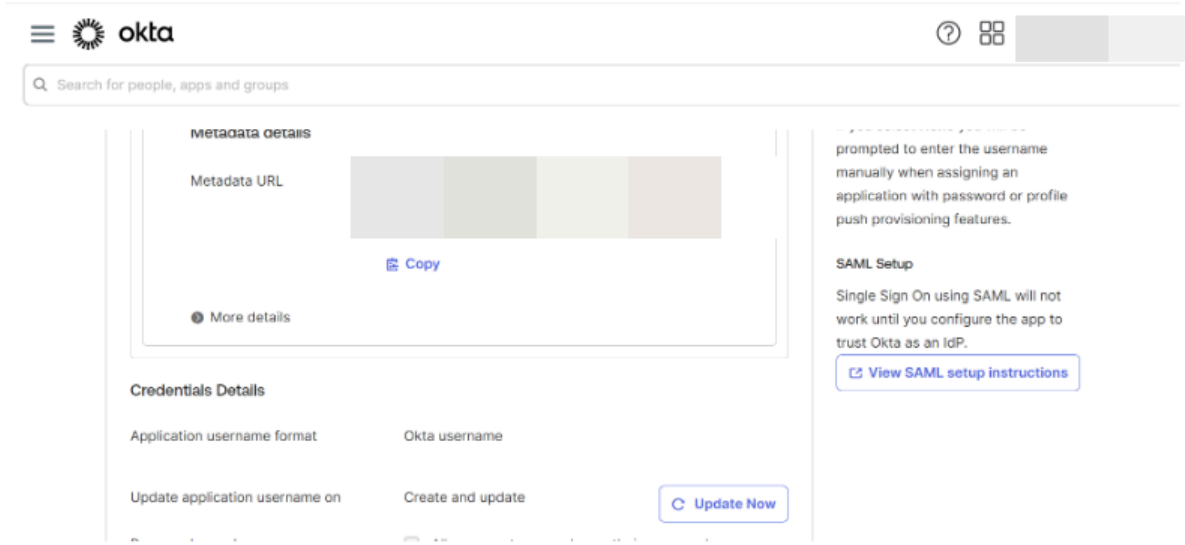
6. Click on **Preview SAML Assertion** to review the changes and click **Next**.
7. In addition, select **This is an internal app that we have created** checkbox and click **Finish** to complete the SAML integration.



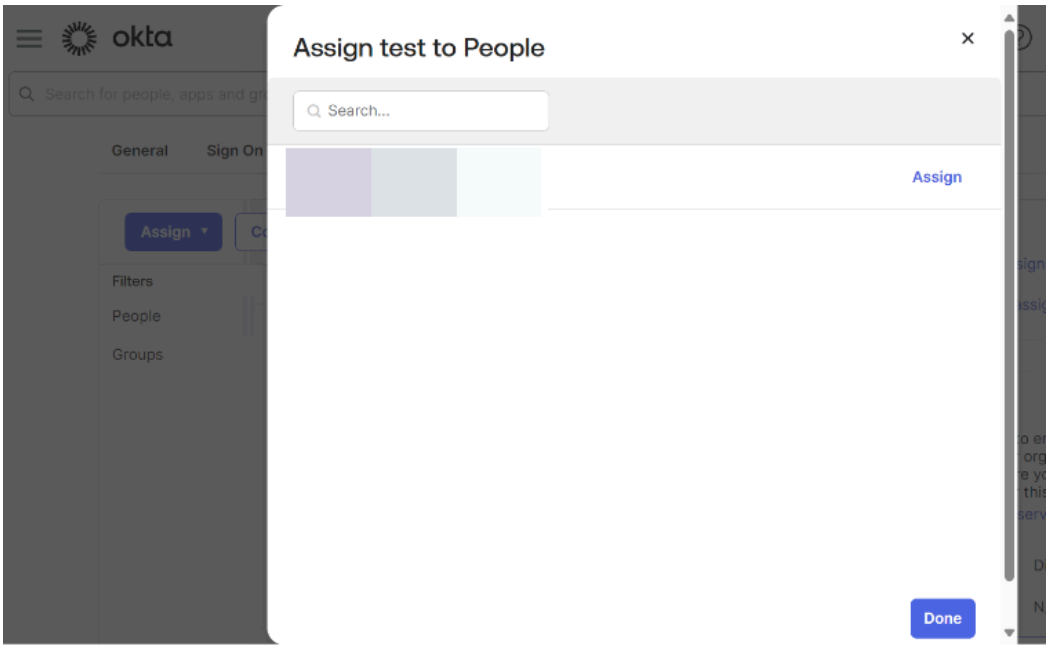
8. Go to **Application** and select the newly created application. The applications opens.



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.



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.

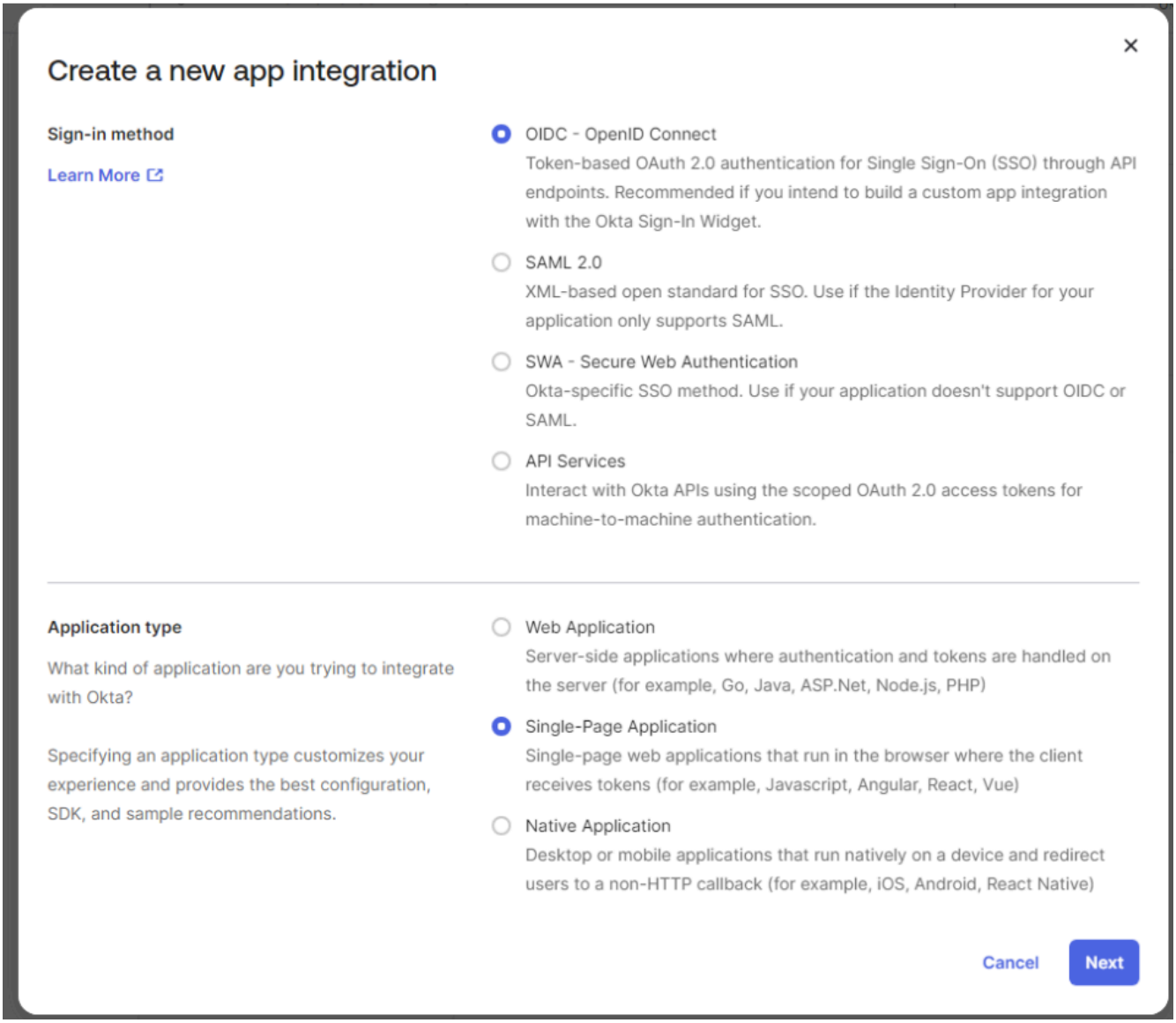


Configuring Excel Add-In (OIDC), APIs Tenant Information Service, RW IDP Service and RS

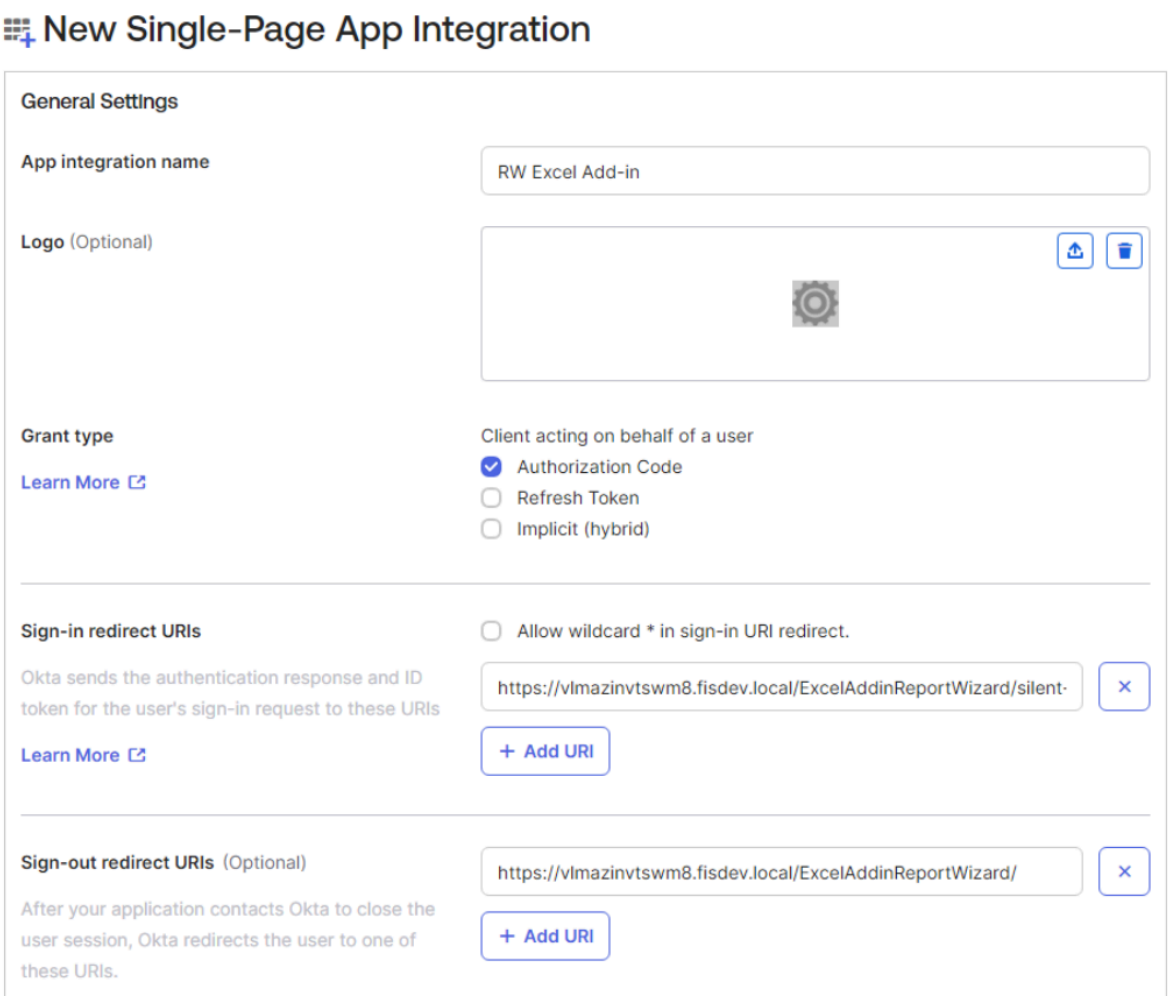
To setup Excel Add-In, APIs and Tenant Information Service using Okta as Identity Provider. The following are the steps:

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

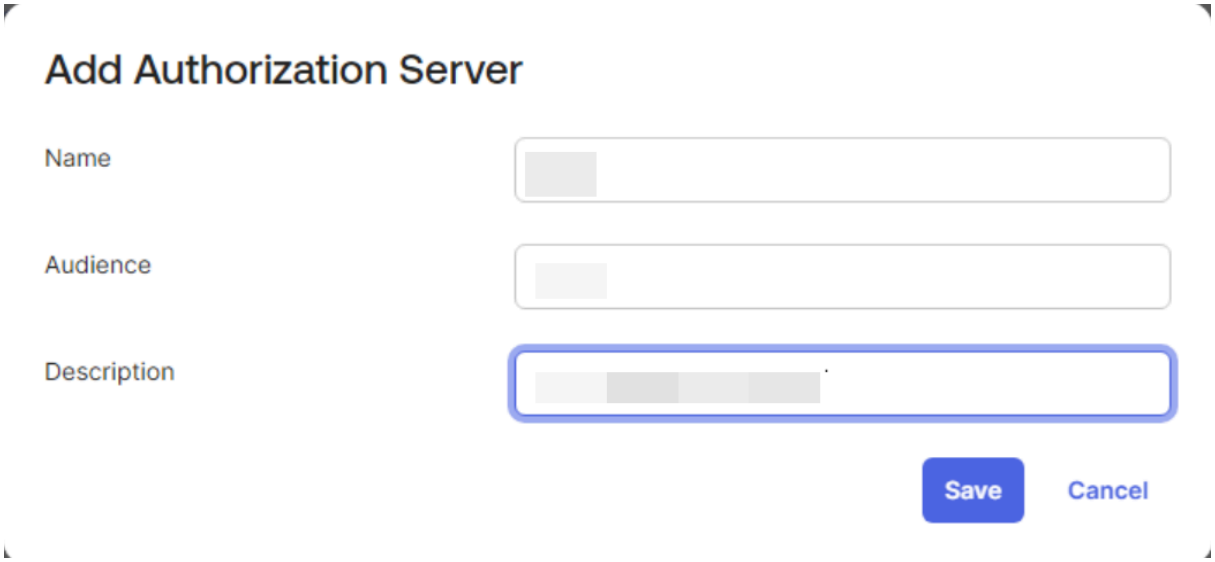
1. Sign in to Okta subscription and navigate to Applications.
2. In the **Create a new app Integration** dialog box, select **Sign-in method** as **OIDC - OpenID Connect** and **Application type** as **Single-Page Application** and click **Next**.



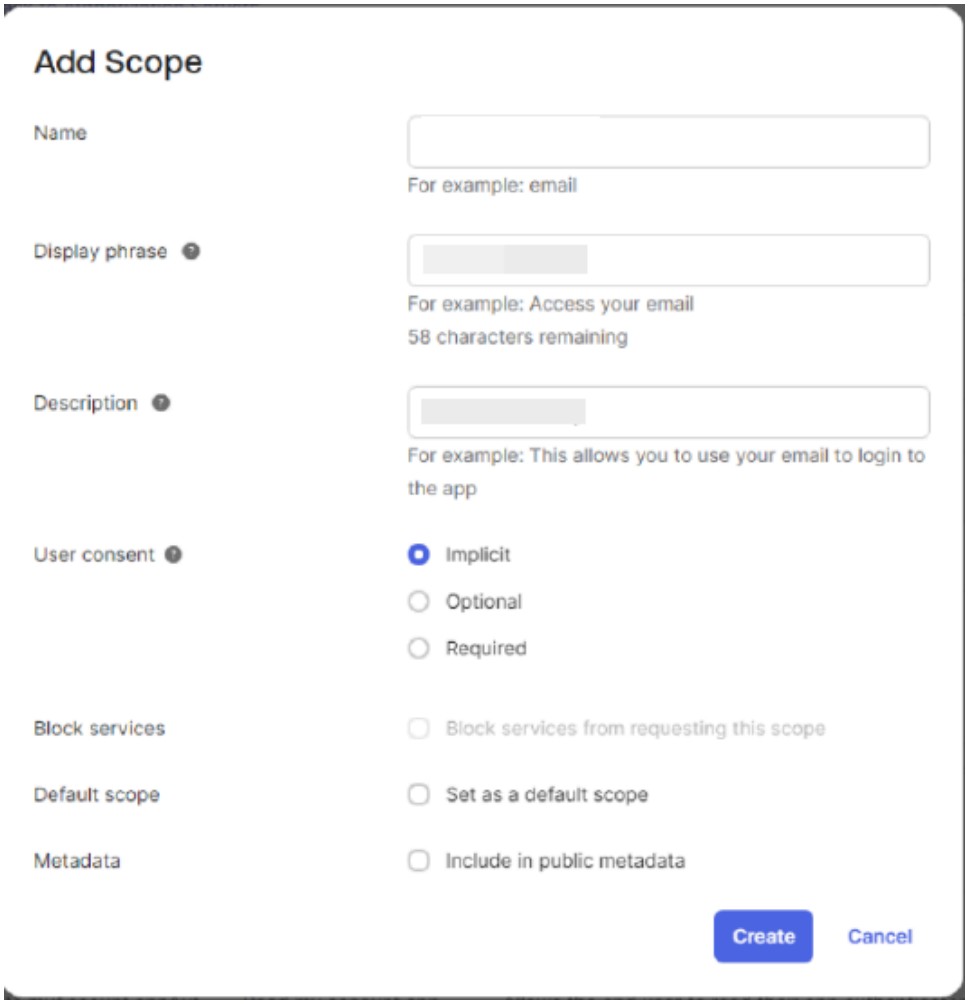
3. In the **New Single-Page App Integration** dialog box, provide the following details:
    i. App integration name: Specify a name in the text field.
    ii. Logo: Upload a logo in the field.
    iii. Grant type: Select Authorization Code.
    iv. Sign-in redirect URIs: Add the sign-in URI
    v. Sign-out redirect URIs: Add the sign-out URI



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.



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.



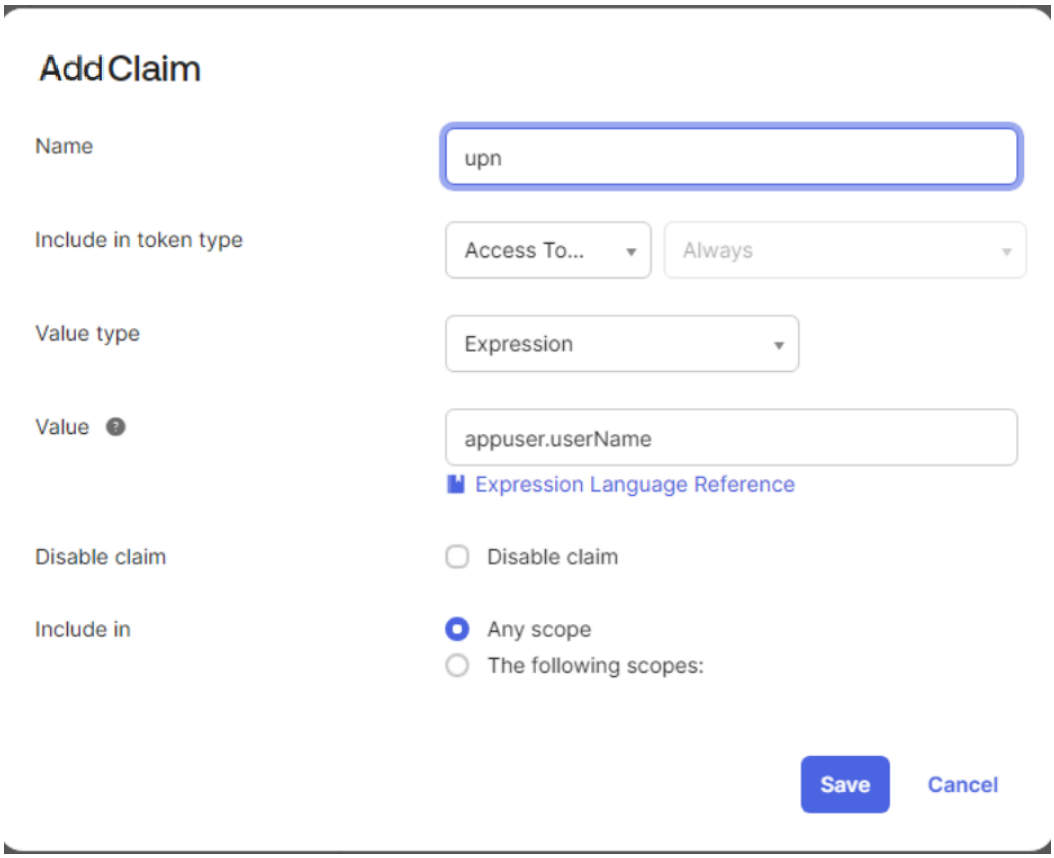
8. Go to **Claims** and in the **Add Claim** dialog box, provide the following details:
    i. Name: Specify the name in the text field.
    ii. Include in token type: Select a token type from the drop down list.
    iii. Value type: Select the value type from the drop down list.
    iv. Value: Specify 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:

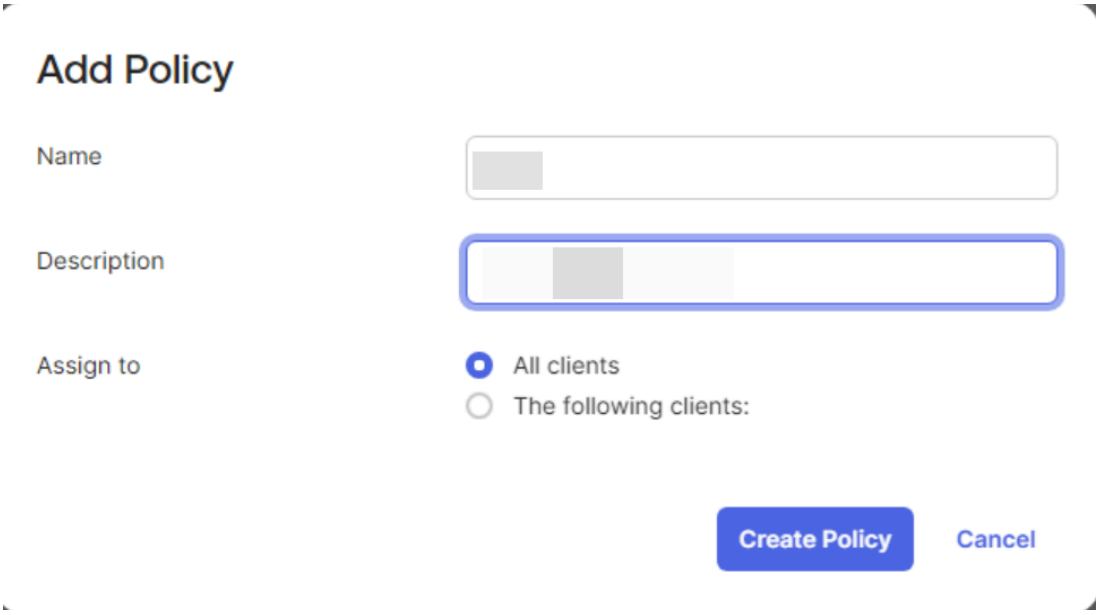
Field Name	Entry 1	Entry 2
Include In	Any scope	Any scope
Include in token	Access Token	Id token

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

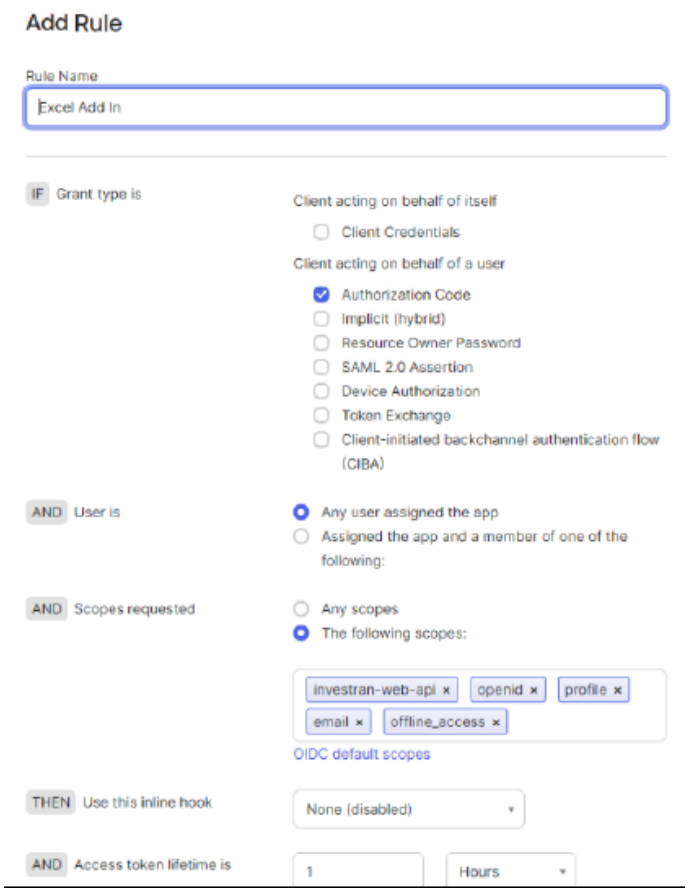
Note: Both the entries that is Entry 1 and Entry 2 are for the same claim that is one for Access Token and other for Id Token.



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**.



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.



13. Click **Save**.

Setting up Client Credential Flow for Authentication among Microservices

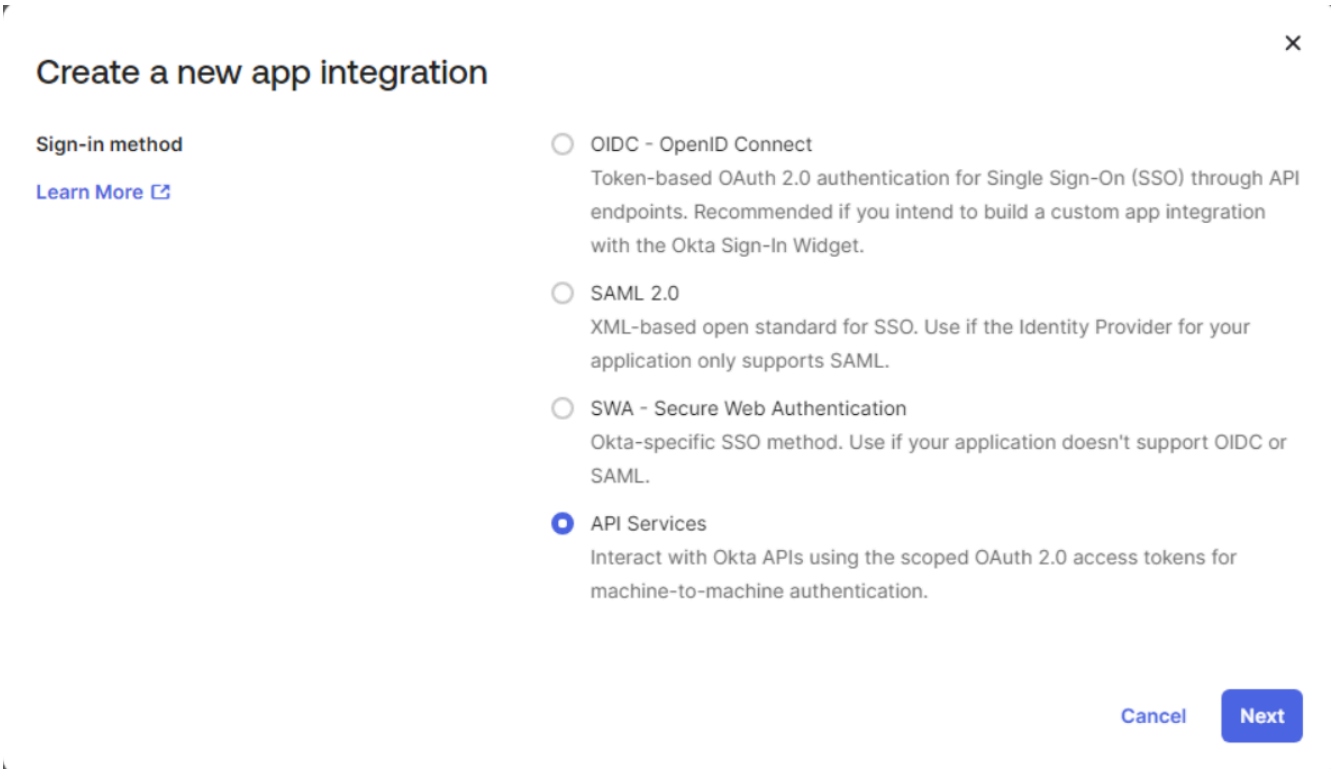
In order to authenticate with other services there are certain services that utilize the Client Credential Flow.

Register App for Client Credential Flow

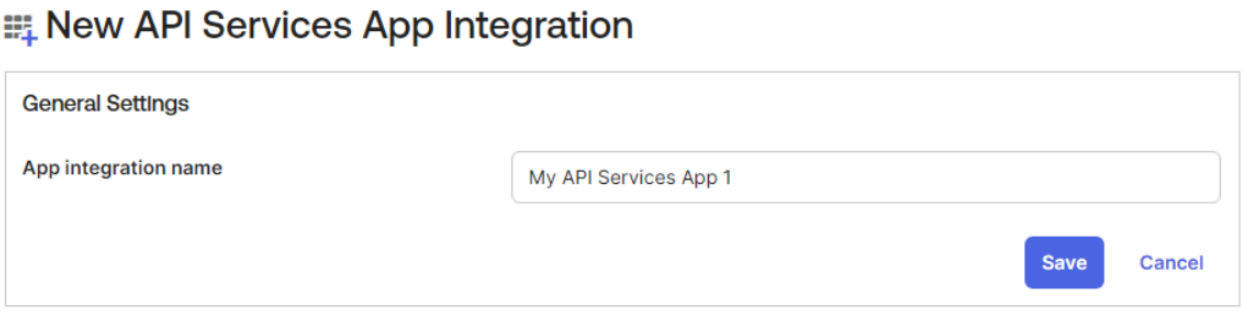
> Note: User must have an active Okta subscription.

The following are the steps to register in Okta:

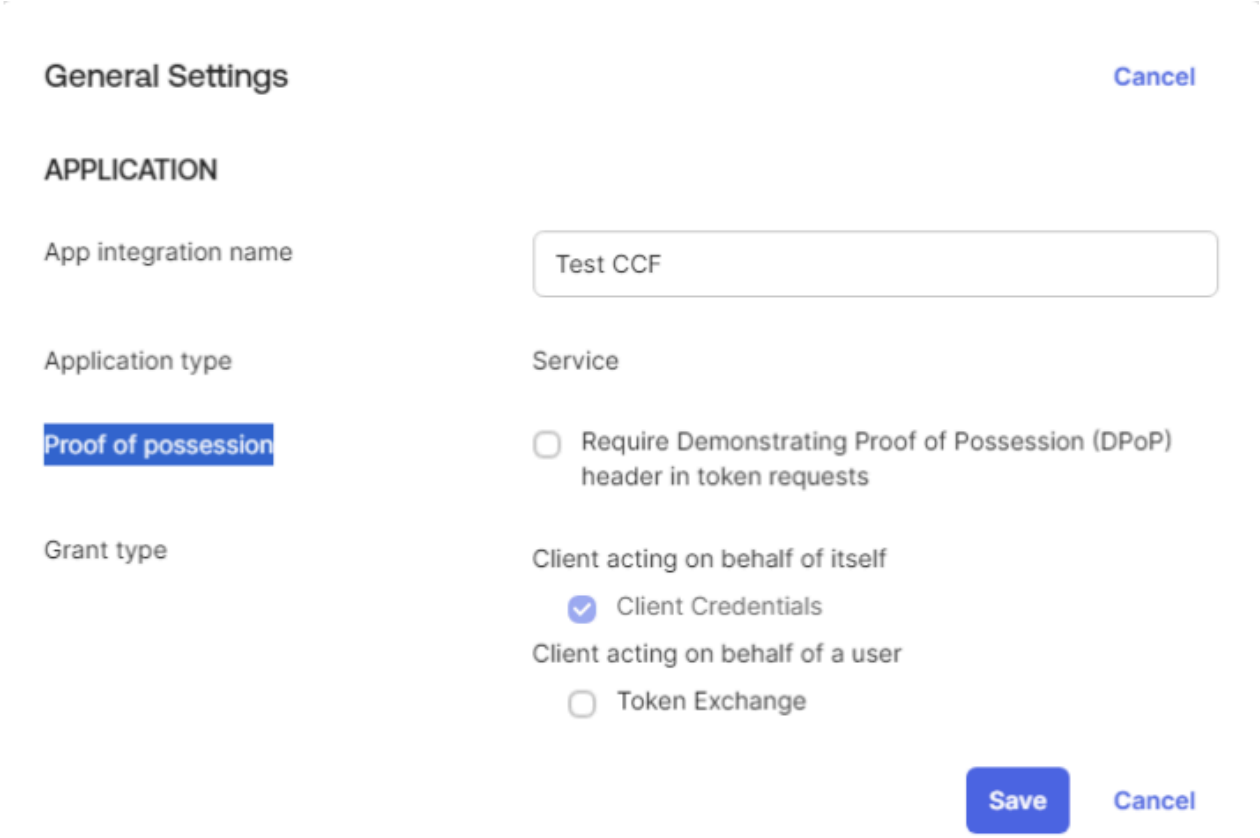
1. Sign in to your Okta subscription and navigate to **Applications** > **Create a new app Integration**.
2. In the **Create a new app Integration** screen, select **API Services** and click **Next**.



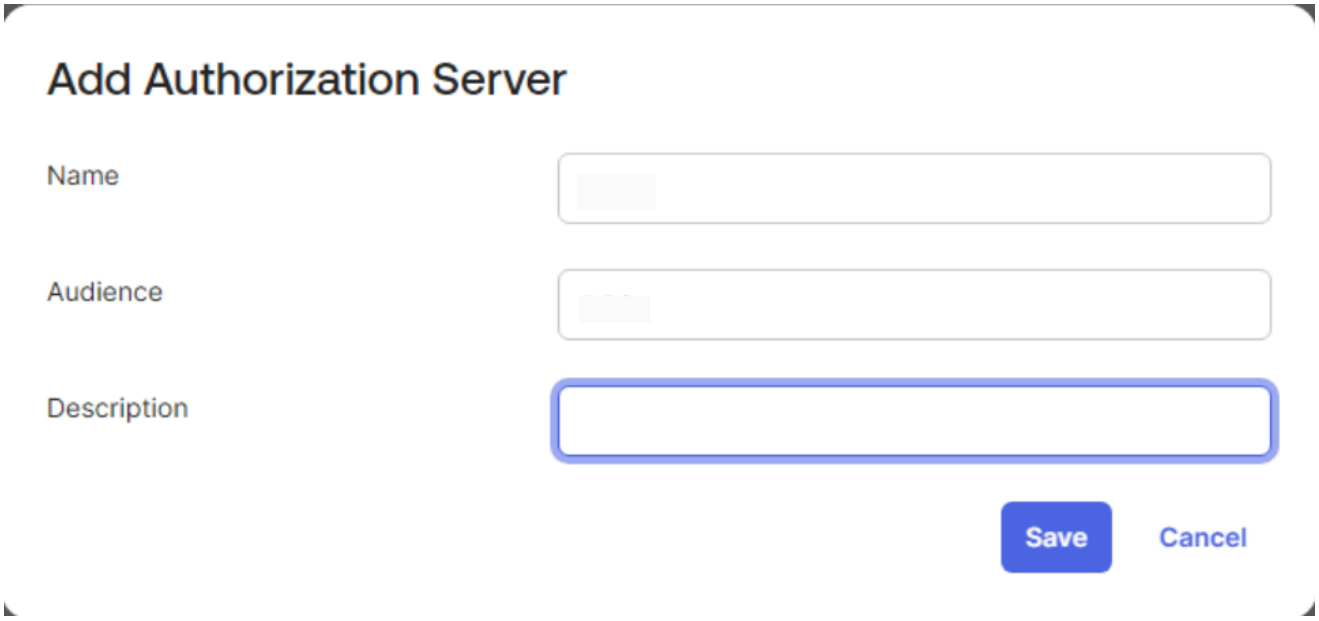
3. In the **New API Services App Integration** dialog box, specify a name in the **App integration name** text field and click **Save**.



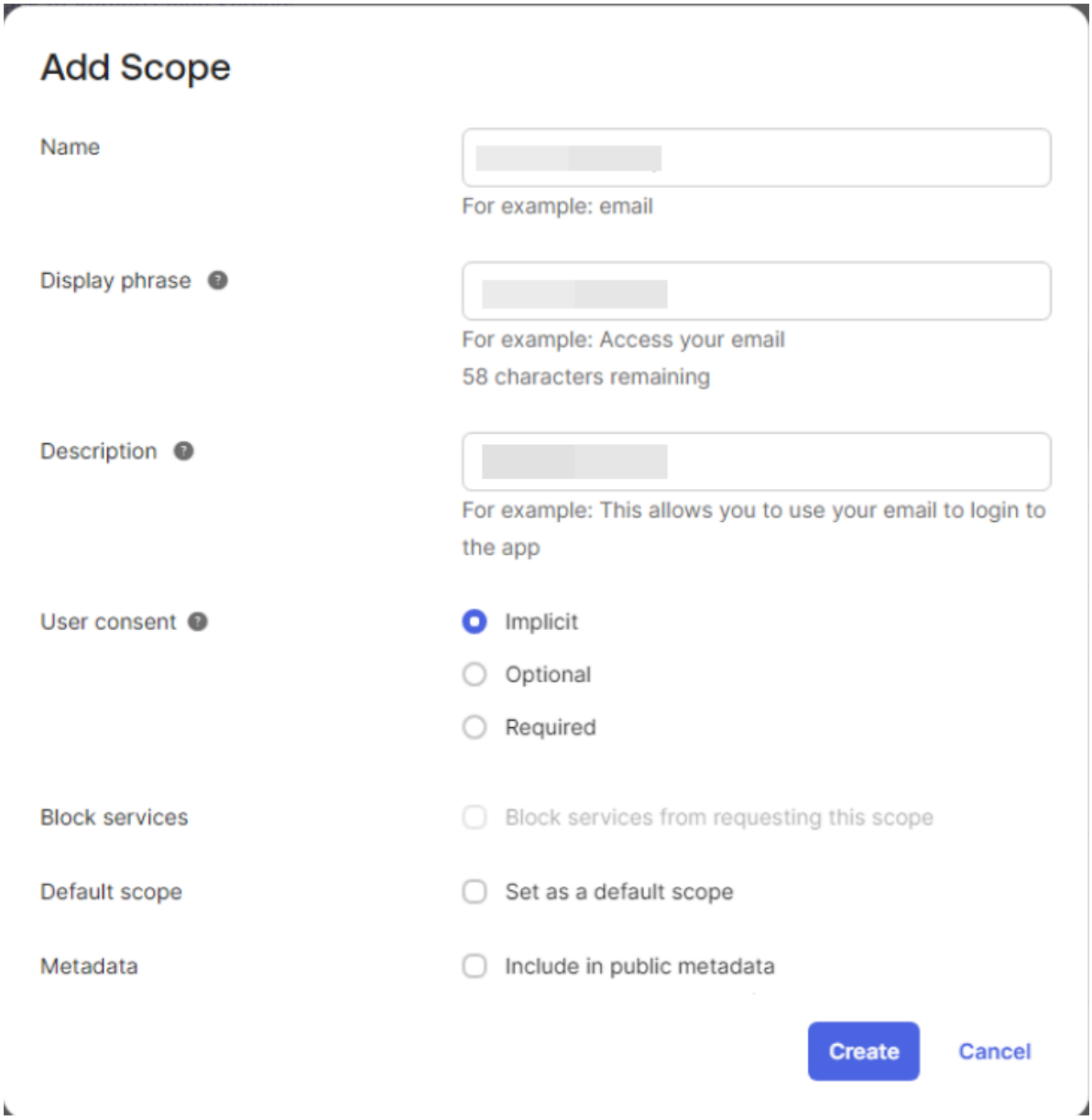
4. Once the application is created, in the **General Settings** section unselect **Proof of possession**, and select **Grant type** as **Client Credentials**, and click **Save**.



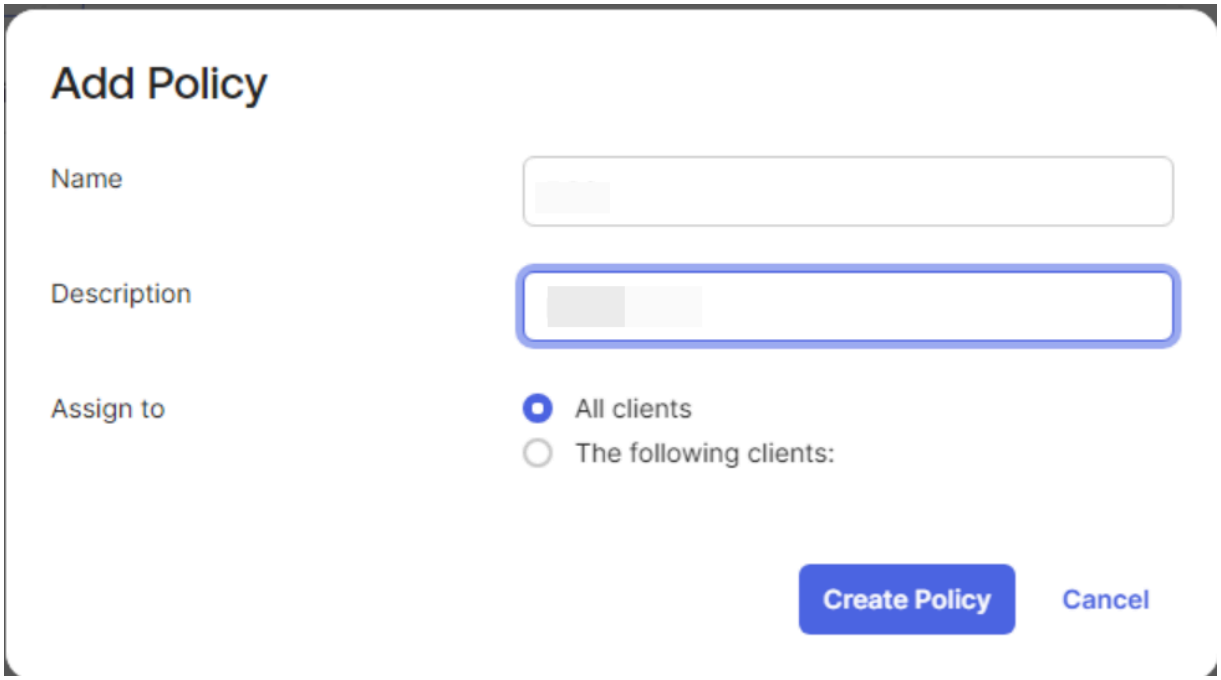
5. Navigate to **Security** > **API**> **Add Authorization Server**. In the **Add Authorization Server** dialog box, specify a name and text in the **Name**, **Audience** and **Description** text fields, and click **Save**.



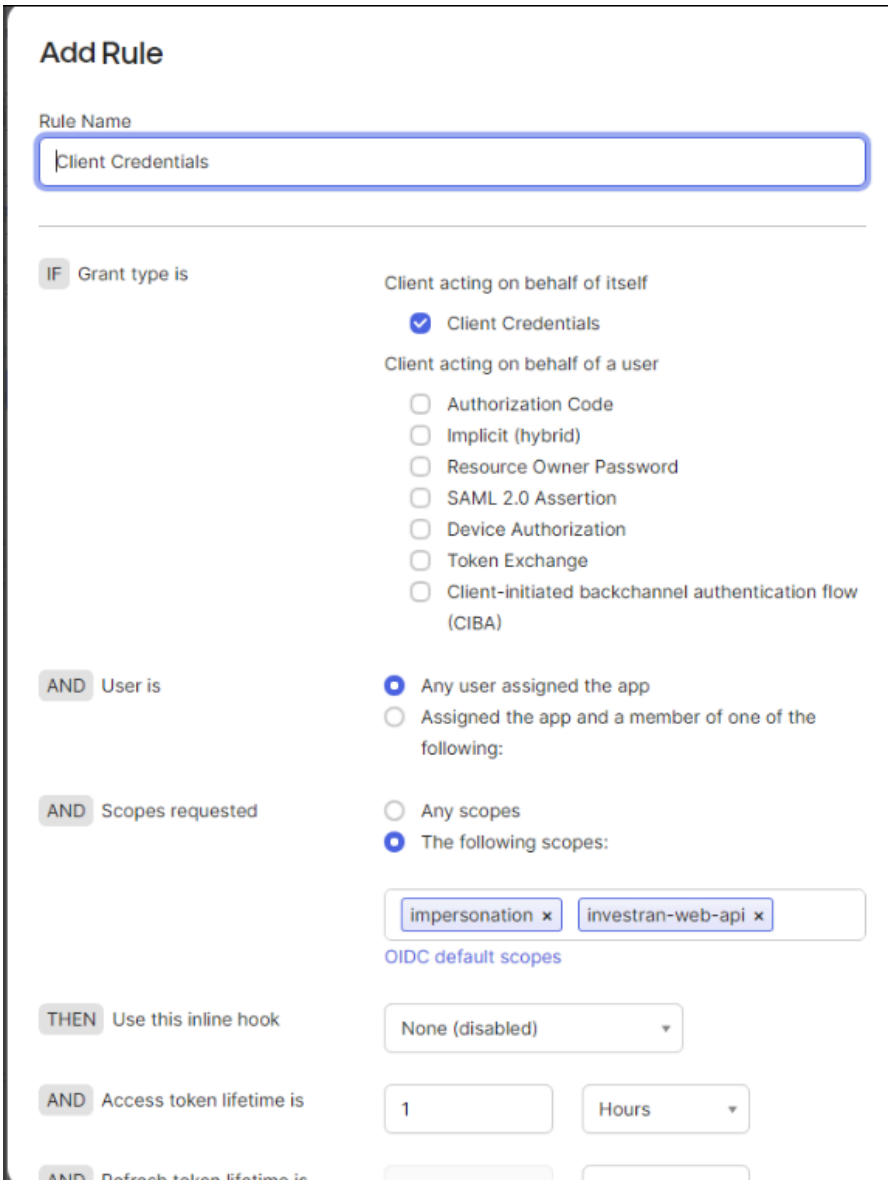
6. Go to **Scope** tab and click **Add Scope**. In the **Add Scope** dialog box, specify a name and text in the **Name**, **Display phrase**, and **Description** text fields. For example, specify name as "impersonation" in the text field.
7. In addition, select the User consent as **Implicit** and click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



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.



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**.



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": "[CERTICATE_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"

....

}
```