REST API Implementation Guide

6.6.0 Release

Table of Contents

Introduction 1

REST API Overview 2

OneStream Web API Endpoints 2

Authentication 2

DataManagement 2

DataProvider 2

OneStream REST API Implementation 3

Authentication 3

OneStream WebAPI Endpoints 4

Configure OneStream API for External Authentication 23

Azure AD Configuration 23

Setup Postman for access_token requests 25

Update the Server Config Utility 26

Okta Configuration 27

Create M2M Application Registration (grant_type = client_credentials) 27

Create U2M Application Registration (grant_type = password) in Okta 28

Update the Server Config Utility 33

PingFederate Configuration 34

Update the Server Config Utility 35
Configure the AUD Value 36

Introduction

The purpose of the OneStream REST API Implentation Guide is to provide information about the implementation, authentication and application programming interfaces available to extending the functionality of OneStream.

OneStream Web API is a RESTful web service designed to expose OneStream Data Automation functions when interacting with third-party API client applications.

OneStream Web API must be installed on a web server and be configured for external authentication providers supporting OAuth2.0/OpenID Connect authorization protocol. Identity Providers currently supported are Okta, Azure AD and PingFederate.

OneStream Web API is API client agnostic. It accepts and outputs data in JSON format making it possible for every API client application that supports this format to also interact with the service.

REST API Overview

OneStream Web API Endpoints

Authentication

Authentication endpoint. Represents a RESTful service for Authentication.

POST api/Authentication/LogonAndReturnCookie

Used primarily by the Enablement Team to verify Web API installation completed successfully. Returns a one-time cookie value that holds authentication state or a message indicating failure along with a proper HTTP code.

DataManagement

DataManagement endpoint. Represents a RESTful service of Data Management.

- POST api/DataManagement/ExecuteSequence:

  Executes a Data Management Sequence and returns a success/failure message along with a proper HTTP code.

- POST api/DataManagement/ExecuteStep

  Executes a Data management Step and returns a success/failure message along with a proper HTTP code.

DataProvider

DataProvider endpoint represents a RESTful service of Data Provider.

- POST api/DataProvider/GetAdoDataSetForAdapter:

  Executes a Data Provider HTTP Post request and returns a JSON representation of a DataSet for a given Dashboard Adapter.

- POST api/DataProvider/GetAdoDataSetForCubeViewCommand

Executes a Data Provider HTTP Post request and returns a JSON representation of a DataSet for a given Cube View.

- POST api/DataProvider/GetAdoDataSetForSqlCommand

  Executes a Data Provider HTTP Post request and returns a JSON representation of a DataSet for a given Sql query. **Administrator role is required for this functionality.**

- POST api/DataProvider/GetAdoDataSetForMethodCommand

  Executes a Data Provider HTTP Post request and returns a JSON representation of a DataSet for a given pre-defined list of method commands. **Administrator role is required for this functionality.**

OneStream REST API Implementation

Authentication

To secure REST API with OAuth 2.0, follow the links below for how to configure authentication with external providers supported by OneSream.

- Azure Configuration
- Okta Configuration
- PingFederate Configuration

Access tokens obtained from the any of the external providers above have, albeit different, relatively short expiration times. To avoid copying the entire token value to the Authorization/Token text box, it's recommended to create a variable that holds the value. For every call to the external provider, the value of the access token returned will be copied to the variable.

- Create a global variable in Postman, name it appropriately, for instance webapi_access_token
- In the Tests tab of the POST request to the external provider copy the script below:

```
var data = pm.response.json();
pm.environment.set("webapi_access_token", data.access_token);
```

OneStream WebAPI Endpoints

This API implementation is client agnostic therefore every API test capable third-party tool can be pointed to OneStreamWeb API endpoints. This tutorial is using Postman. Note that all arguments in the body are **required** unless otherwise specified.

Versioning This implementation will start with Api-version=5.2.0

Data Management Execute Sequence endpoint

1. Create new POST request in Postman,
2. Url= http(s)://[servername]: [port]/onestreamapi/api/DataManagement/ExecuteSequence?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

   {
   "**BaseWebServerUrl**": [your web server url],
   "**ApplicationName**":[your application name],
   "**SequenceName**": [existing sequence name],
   "**CustomSubstVarsAsCommaSeparatedPairs**": [comma separated list of key value pairs as substitution variables with the following format: "VariableName1= [VariableValue1],VariableName2=[VariableValue2],..."] - *Optional*
   }
6. Click Send and observe the response at the bottom pane. If successful, a message of "Data Management Sequence [sequence name] was completed" will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.

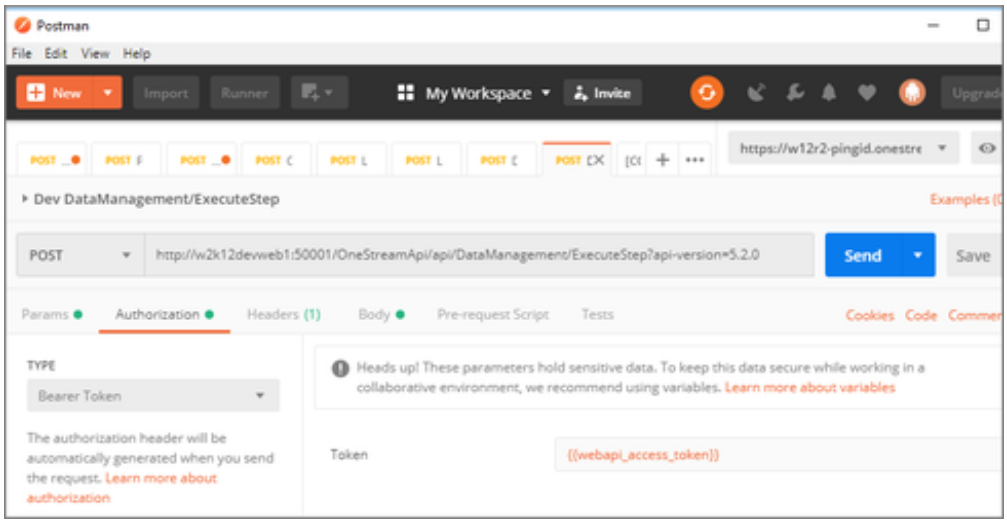
Data Management Execute Step endpoint

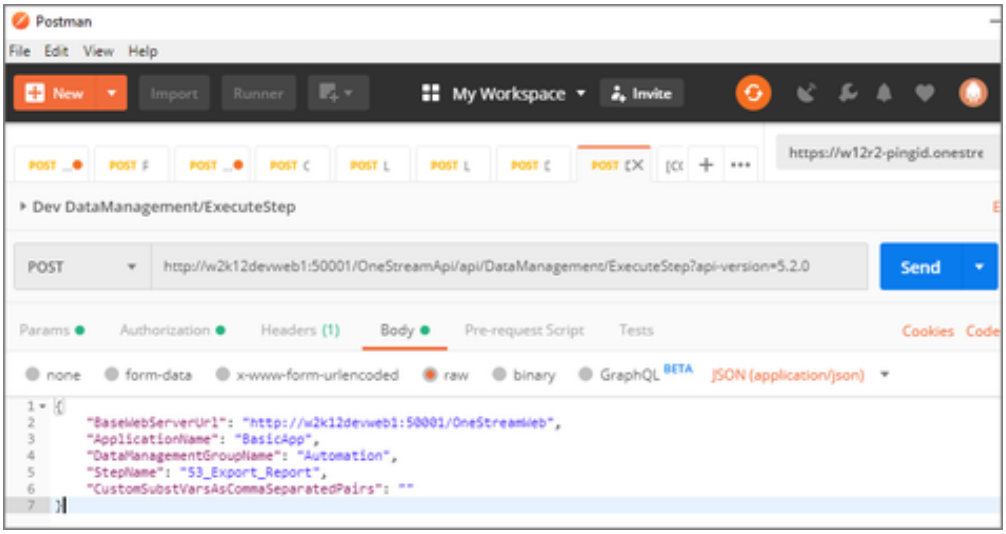
1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/onestreamapi/api/DataManagement/ExecuteStep?api-version=5.2.0

3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

```
{
      "BaseWebServerUrl": [your web server url ],
      "ApplicationName":[your application name],
      "DataManagementGroupName": [an existing data management group name],
      "StepName": [existing step name],
      "CustomSubstVarsAsCommaSeparatedPairs": [comma separated list of key value pairs as substitution variables with the following format: "VariableName1=[VariableValue1],VariableName2=[VariableValue2],..."] - Optional
 }
```

6. Click Send and observe the response at the bottom pane. If successful, a message of "Data Management Step [step name] was completed" will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.





Data Provider GetAdoDataSetForAdapter endpoint

1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/onestreamapi/api/DataProvider/GetAdoDataSetForAdapter?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

```
{
     "BaseWebServerUrl": [your web server url ],
     "ApplicationName":[your application name],
     "WorkspaceName": Reserved for future use. Use an empty string. -
Optional,
     "AdapterName": [existing adapter name],

     "ResultDataTableName": [name of resulting table in the DataSet],

     "CustomSubstVarsAsCommaSeparatedPairs": [comma separated list of key
value pairs as substitution variables with the following format:
"VariableName1=[VariableValue1],VariableName2=[VariableValue2],..."] -
Optional
 }
```

```
Example:
{
    "BaseWebServerUrl": "http://localhost:50528/OneStream",
    "ApplicationName": "GolfStream_v37",
    "IsSystemLevel": "False",
    "AdapterName": "ActivityClassListing_PLP",
    "ResultDataTableName": "ResultsTable",
    "CustomSubstVarsAsCommaSeparatedPairs": ""
}
```

6. Click Send and observe the response at the bottom pane. If successful, a JSON data table will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.

   This is a returned response from the request using the above body example in Postman:


```
{
  "ResultsTable": [
    {
      "ClassID": "100_Salary",
      "Name": "100 - Salary",
      "Description": "100 - Salary",
      "ValueType": 0,
      "ValueTypeName": "Wage Percentage",
      "ClassItemID": "79b612b9-8cb4-49ca-9a0d-d13c7683a7f2",
      "Description1": "100 - Salary",
      "WeightOrValue": "1",
      "FKAccountID": "Salary_Exp",
      "Flow": "None",
      "IC": "None",
      "UD1": "None",
```

"UD2": "None",

"UD3": "None",

"UD4": "None",

"UD5": "None",

"UD6": "None",

"UD7": "None",

"UD8": "None",

"Sequence": 10.0,

"FKClassID": "100_Salary"

},

…

]}}

Data Provider GetAdoDataSetForCubeViewCommand endpoint

1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/onestreamapi/api/DataProvider/GetAdoDataSetForCubeViewCommand?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

```
{
     "BaseWebServerUrl": [your web server url ],
     "ApplicationName":[your application name],
     "CubeViewName": [existing Cube View name],
     "DataTablePerCubeViewRow ": [if true returns a Data Table Per Cube View row - bool],

     "ResultDataTableName": [name of resulting table in the DataSet],
```

"CubeViewDataTableOptions": [set of formatting bolean options for the returned table - *Optional*],

"CustomSubstVarsAsCommaSeparatedPairs": [comma separated list of key value pairs as substitution variables with the following format: "VariableName1=[VariableValue1],VariableName2=[VariableValue2],..."] - *Optional*
}

Example:

```
{
    "BaseWebServerUrl": "http://localhost:50528/OneStream",
    "ApplicationName": "GolfStream_v37",
    "CubeViewName": "Gross Margin",
    "DataTablePerCubeViewRow": false,
    "ResultDataTableName": "ResultDataTableNames",
    "CustomSubstVarsAsCommaSeparatedPairs": "",
    "CubeViewDataTableOptions": {
                "IncludeTitle": false,
                "IncludeHeaderLeftLabel1" : true,
                "IncludeHeaderLeftLabel2" : true,
                "IncludeHeaderLeftLabel3" : true,
                "IncludeHeaderLeftLabel4" : true,
                "IncludeHeaderCenterLabel1" : true,
                "IncludeHeaderCenterLabel2" : true,
                "IncludeHeaderCenterLabel3" : true,
                "IncludeHeaderCenterLabel4" : true,
                "IncludeHeaderRightLabel1" : true,
                "IncludeHeaderRightLabel2" : true,
                "IncludeHeaderRightLabel3" : true,
                "IncludeHeaderRightLabel4" : true,
```

```
        "IncludePovCube" : true,
        "IncludePovEntity" : true,
        "IncludePovParent" : true,
        "IncludePovCons" : true,
        "IncludePovScenario" : true,
        "IncludePovTime" : true,
        "IncludePovView" : true,
        "IncludePovAccount" : true,
        "IncludePovFlow" : true,
        "IncludePovOrigin" : true,
        "IncludePovIC" : true,
        "IncludePovUD1" : true,
        "IncludePovUD2" : true,
        "IncludePovUD3" : false,
        "IncludePovUD4" : true,
        "IncludePovUD5" : false,
        "IncludePovUD6" : true,
        "IncludePovUD7" : false,
        "IncludePovUD8" : true,
        "IncludeMemberDetails": true,
        "IncludeRowNavigationLink" : true,
        "IncludeHasDataStatus" : true,
        "IncludeAnnotation" : true,
        "IncludeAssumptions" : true,
        "IncludeAuditComment" : true,
        "IncludeFootnote" : true,
        "IncludeVarianceExplanation" : true
}
```

```
}
```

6. Click Send and observe the response at the bottom pane. If successful, a JSON data table will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.

   This is a returned response from the request using the above body example in Postman:

```
{
    "ResultDataTableNames": [
        {
            "RowId": 0,
            "RowName": "Row1",
            "HeaderLeftLabel1": "",
            "HeaderLeftLabel2": "",
            "HeaderLeftLabel3": "",
            "HeaderLeftLabel4": "",
            "HeaderCenterLabel1": "",
            "HeaderCenterLabel2": "",
            "HeaderCenterLabel3": "",
            "HeaderCenterLabel4": "",
            "HeaderRightLabel1": "",
            "HeaderRightLabel2": "",
            "HeaderRightLabel3": "",
            "HeaderRightLabel4": "",
            "PovCubeId": 5,
            …
            "Col8VarianceExplanation": ""
        },
…
] } }
```

Data Provider GetAdoDataSetForSqlCommand endpoint

1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/onestreamapi/api/DataProvider/GetAdoDataSetForSqlCommand?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

   {
   "**BaseWebServerUrl**": [your web server url],
   "**ApplicationName**":[your application name],
   "**SqlQuery** ": [sql query statement used to return data],
   "**DbLocation**": [specify if data from an external database referenced in the configuration will need to be returned - string - defaults to "Application" - *Optional*],

   "**ResultDataTableName**": [name of resulting table in the DataSet],

   "**XFExternalDBConnectionNam** ": [specify if DbLocation is set to "External"],

   "**CustomSubstVarsAsCommaSeparatedPairs**": [comma separated list of key value pairs as substitution variables with the following format: "VariableName1= [VariableValue1],VariableName2=[VariableValue2],..."] - *Optional*
   }

   Example:

```
{
    "BaseWebServerUrl": "http://localhost:50528/OneStream",
    "ApplicationName": "GolfStream_v37",
    "SQLQuery": "Select TOP 100 * from Cube",
    "ResultDataTableName": "ResultDataTableName",
    "DBLocation": "Application",
    "XFExternalConnectionName": "",
    "CustomSubstVarsAsCommaSeparatedPairs": ""
}
```

6. Click Send and observe the response at the bottom pane. If successful, a JSON data table will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.

   This is a returned response from the request using the above body example in Postman:


```
{
    "ResultDataTableName": [
        {
            "CubeId": 0,
            "Name": "Houston",
            "Description": "Houston Clubs",
            "CubeType": 0,
            "IsTopLevelCube": false,
            "TimeDimProfileID": "664c9bd4-a314-4941-81be-513aeddac13a",
            "AccessGroupUniqueID": "e31054d8-83bf-4f79-b563-
0e450342de9e",
            "MaintenanceGroupUniqueID": "e31054d8-83bf-4f79-b563-
0e450342de9e",
            "ConsAlgorithmType": 0,
            "TransAlgorithmType": 0,
            "CalcNoneConsIfNoData": false,
            "CalcLocalCurrIfNoData": true,
            "CalcTransCurrsIfNoData": false,
            "CalcOwnerPreAdjIfNoData": false,
            "CalcShareIfNoData": false,
            "CalcElimIfNoData": false,
            "CalcOwnerPostAdjIfNoData": false,
            "BR1Name": "CorporateBusinessRules",
            "BR2Name": "",
            "BR3Name": "",
```

```
            "BR4Name": "",
            "BR5Name": "",
            "BR6Name": "",
            "BR7Name": "",
            "BR8Name": "",
            "DefaultCurrencyId": 176,
            "FxRateTypeIDForRevExp": "89ce1f1c-c1cb-438e-9825-
e00861a4fa5b",
            "FxRuleTypeIdForRevExp": 1,
            "FxRateTypeIDForAssetLiab": "89ce1f1c-c1cb-438e-9825-
e00861a4fa5b",
            "FxRuleTypeIdForAssetLiab": 0,
            "XmlData": ""
        },
...
] } }
```


Important: The Administrator role is required for this functionality.

Data Provider GetAdoDataSetForMethodCommand endpoint

1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/onestreamapi/api/DataProvider/GetAdoDataSetForMethodCommand?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

```
{
     "BaseWebServerUrl": [your web server url ],
     "ApplicationName":[your application name],
```

```
     "MethodQuery":[method query to return data],
     "XFCommandMethodTypeId": [pre-defined list of XF method commands
used by    XFDataProvider to fill a DataSet],
     "ResultDataTableName": [name of resulting table in the DataSet],

     "CustomSubstVarsAsCommaSeparatedPairs": [comma separated list of key
value pairs as substitution variables with the following format:
"VariableName1=[VariableValue1],VariableName2=[VariableValue2],..."] -
Optional
 }
```

Example:

```
{

    "BaseWebServerUrl": "http://localhost:50528/OneStream",

    "ApplicationName": "GolfStream_v37",

    "MethodQuery ": "{Houston}{Actual}{2018M1}{true}{}",

    "XFCommandMethodTypeId ": "CertificationForWorkflowUnit",

    "ResultDataTableName": "MyResultsTable",

    "CustomSubstVarsAsCommaSeparatedPairs": ""

}
```

XFCommandMethodTypeId may take any values from the list below:

"WorkflowCalculationEntities"

"WorkflowConfirmationEntities"

"WorkflowProfileAndDependentProfileEntities"

"WorkflowProfileEntities"

"WorkflowProfiles"

"WorkflowProfileRelatives"

"WorkflowStatus"

"WorkflowStatusTwelvePeriod"

"WorkflowAndEntityStatus

"JournalsForWorkflowUnit"

"FormsStatusForWorkflowUnit"

"ConfirmationForWorkflowUnit"

"CertificationForWorkflowUnit"

"ICMatchingForWorkflowUnit"

"ICMatchingForWorkflowUnitMultiPlug"

"ICMatchingForWorkflowUnitMultiPeriod"

"ICMatchingPlugAccountsForWorkflowUnit"

6. Click Send and observe the response at the bottom pane. If successful, a JSON data table will be returned otherwise a descriptive error message will show. More details will be logged in the Error and Activity logs.

   This is a returned response from the request using the above body example in Postman:


```
{
    "MyResultsTable": [
        {
            "ProfileName": "Houston",
            "ProfileKey": "2f3a719e-8e26-4d8c-8cc7-4544a4812673",
            "ProfileOrder": 1,
            "ScenarioName": "Actual",
            "ScenarioKey": 0,
            "TimeKey": 2018003000,
            "TimeName": "2018M1",
            "CertProfileKey": "003e0a15-6c9a-412c-90ba-64d31040c314",
            "CertName": "Plant Certification",
            "CertDescription": "Plant Certification",
            "CertSignOffState": "Inprocess",
            "CertIsCertified": false,
            "CertCanCertify": false,
            "CertIsParentCertified": false,
            "CertAreDependantsCertified": false,
```

```
                "CertAllAnswered": false,
                "CertQuestionCount": 3,
                "CertUnansweredCount": 3,
                "CertUnansweredRate": 1.0,
                "GroupKey": "7c7fedcd-f04a-4f5b-ba13-ed1097f449a9",
                "GroupName": "SOX Plant Controller",
                "GroupDescription": "SOX Plant Controller",
                "GroupSignOffState": "Inprocess",
                "GroupAllAnswered": false,
                "GroupQuestionCount": 3,
                "GroupUnansweredCount": 3,
                "GroupUnansweredRate": 1.0,
                "QuestionUniqueID": "8a92f59c-2419-49d2-87b7-1cdfb21c7072",
                "QuestionName": "Unusual Transactions",
                "QuestionCategory": "InternalAudit",
                "QuestionRiskLevel": "High",
                "QuestionFrequency": "AllTimePeriods",
                "TimeFilterForReqtFreq": "",

    "QuestionText": "Any unusual transactions booked? If so, explain. ",
        "QuestionResponse": "-1",
        "QuestionComments": "",
        "QuestionResponseOptional": false,
        "QuestionDeactivated": false,
        "QuestionDeactivationDate": "1900-01-01T00:00:00",
        "QuestionDisplayOrder": 10
    },
    {
```

"ProfileName": "Houston",
"ProfileKey": "2f3a719e-8e26-4d8c-8cc7-4544a4812673",
"ProfileOrder": 1,
"ScenarioName": "Actual",
"ScenarioKey": 0,
"TimeKey": 2018003000,
"TimeName": "2018M1",
"CertProfileKey": "003e0a15-6c9a-412c-90ba-64d31040c314",
"CertName": "Plant Certification",
"CertDescription": "Plant Certification",
"CertSignOffState": "Inprocess",
"CertIsCertified": false,
"CertCanCertify": false,
"CertIsParentCertified": false,
"CertAreDependantsCertified": false,
"CertAllAnswered": false,
"CertQuestionCount": 3,
"CertUnansweredCount": 3,
"CertUnansweredRate": 1.0,
"GroupKey": "7c7fedcd-f04a-4f5b-ba13-ed1097f449a9",
"GroupName": "SOX Plant Controller",
"GroupDescription": "SOX Plant Controller",
"GroupSignOffState": "Inprocess",
"GroupAllAnswered": false,
"GroupQuestionCount": 3,
"GroupUnansweredCount": 3,
"GroupUnansweredRate": 1.0,

```
    "QuestionUniqueID": "78e102c2-cda5-4c07-b853-416d83de5706",
    "QuestionName": "Audit Transactions",
    "QuestionCategory": "ExternalAudit",
    "QuestionRiskLevel": "High",
    "QuestionFrequency": "AllTimePeriods",
    "TimeFilterForReqtFreq": "",
    "QuestionText": "Any transactions to be reviewed by external audit? If so, explain. ",
    "QuestionResponse": "-1",
    "QuestionComments": "",
    "QuestionResponseOptional": false,
    "QuestionDeactivated": false,
    "QuestionDeactivationDate": "1900-01-01T00:00:00",
    "QuestionDisplayOrder": 20
  },
  {
    "ProfileName": "Houston",
    "ProfileKey": "2f3a719e-8e26-4d8c-8cc7-4544a4812673",
    "ProfileOrder": 1,
    "ScenarioName": "Actual",
    "ScenarioKey": 0,
    "TimeKey": 2018003000,
    "TimeName": "2018M1",
    "CertProfileKey": "003e0a15-6c9a-412c-90ba-64d31040c314",
    "CertName": "Plant Certification",
    "CertDescription": "Plant Certification",
    "CertSignOffState": "Inprocess",
    "CertIsCertified": false,
```

"CertCanCertify": false,
"CertIsParentCertified": false,
"CertAreDependantsCertified": false,
"CertAllAnswered": false,
"CertQuestionCount": 3,
"CertUnansweredCount": 3,
"CertUnansweredRate": 1.0,
"GroupKey": "7c7fedcd-f04a-4f5b-ba13-ed1097f449a9",
"GroupName": "SOX Plant Controller",
"GroupDescription": "SOX Plant Controller",
"GroupSignOffState": "Inprocess",
"GroupAllAnswered": false,
"GroupQuestionCount": 3,
"GroupUnansweredCount": 3,
"GroupUnansweredRate": 1.0,
"QuestionUniqueID": "3d9c4dcc-75fd-4568-b224-f7e428622917",
"QuestionName": "Key Data Review",
"QuestionCategory": "FinancialStatementReview",
"QuestionRiskLevel": "MediumLow",
"QuestionFrequency": "AllTimePeriods",
"TimeFilterForReqtFreq": "",
"QuestionText": "Have all key metrics been reviewed? ",
"QuestionResponse": "-1",
"QuestionComments": "",
"QuestionResponseOptional": false,
"QuestionDeactivated": false,
"QuestionDeactivationDate": "1900-01-01T00:00:00",

```
      "QuestionDisplayOrder": 30
    }
  ],
  "MyResultsTable_SignOffCert": [
    {
      "ProfileKey": "2f3a719e-8e26-4d8c-8cc7-4544a4812673",
      "ScenarioKey": 0,
      "TimeKey": 2018003000,
      "CertProfileKey": "003e0a15-6c9a-412c-90ba-64d31040c314",
      "SignOffState": "Inprocess",
      "Comments": "Sign-Off Initialized",
      "UserKey": "2b61ed59-63ae-46f2-89aa-a8ee9f14bacd",
      "UserName": "TestUserOkta",
      "UserIPAddress": "8d3d857e-cd62-4fd9-a2ec-43b46217a036",
      "TimeStamp": "2019-11-18T14:45:00.007"
    }
  ],
  "MyResultsTable_SignOffGroups": [
    {
      "ProfileKey": "2f3a719e-8e26-4d8c-8cc7-4544a4812673",
      "ScenarioKey": 0,
      "TimeKey": 2018003000,
      "CertProfileKey": "003e0a15-6c9a-412c-90ba-64d31040c314",
      "CertProfileName": "Plant Certification",
      "GroupKey": "7c7fedcd-f04a-4f5b-ba13-ed1097f449a9",
      "GroupName": "SOX Plant Controller",
      "SignOffState": "Inprocess",
```

```
        "Comments": "Sign-Off Initialized",
        "UserKey": "2b61ed59-63ae-46f2-89aa-a8ee9f14bacd",
        "UserName": "TestUserOkta",
        "UserIPAddress": "8d3d857e-cd62-4fd9-a2ec-43b46217a036",
        "TimeStamp": "2019-11-18T14:45:00.2"
      }
    ]
}
```


Important: The Administrator role is required for this functionality.

Authentication Execute LogonAndReturnCookie endpoint

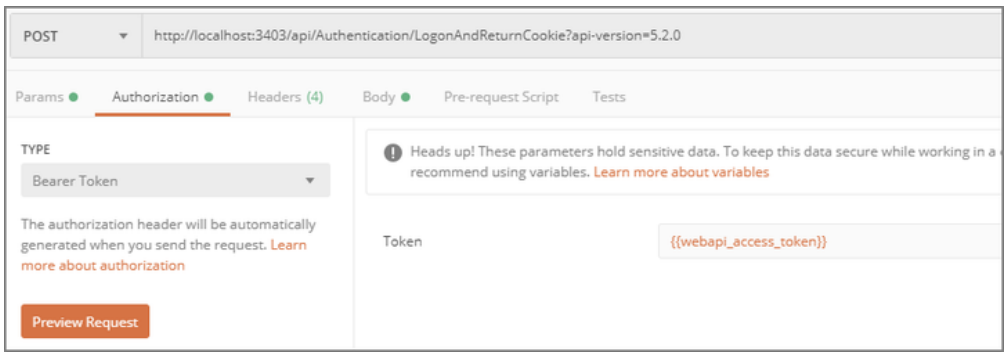
Returns a one-time cookie value that indicates authentication state. Used mostly by enablement team to verify the installation of web API completed successfully.

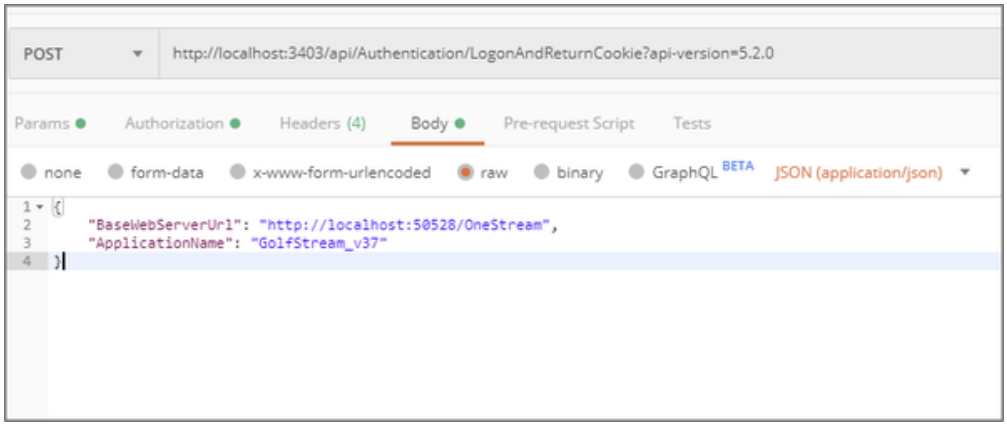
1. Create new POST request in Postman,
2. Url= http(s)://[servername]:[port]/OneStreamApi/api/Authentication/LogonAndReturnCookie?api-version=5.2.0
3. Authorization: Type=Bearer Token. Token={{webapi_access_token}}
4. Headers: Content-Type=application/json
5. Body (raw / jSON):

   Arguments:
   "**BaseWebServerUrl**": [your web server url],
   "**ApplicationName**" : [name of Application attempted to access]

   <response code="200">Returns a JSON representation of the resulting DataSet.</response>
   <response code="400">Bad Request. Missing Authentication arguments. </response>
   <response code="500">Error Message. Authentication Failed. Please check the Error Log for more details</response>

Click Send and observe the response at the bottom pane. If successful, a one-time cookie value that indicates authentication state will be returned. Otherwise the error message will be shown. More details will be logged in the Error and Activity logs.





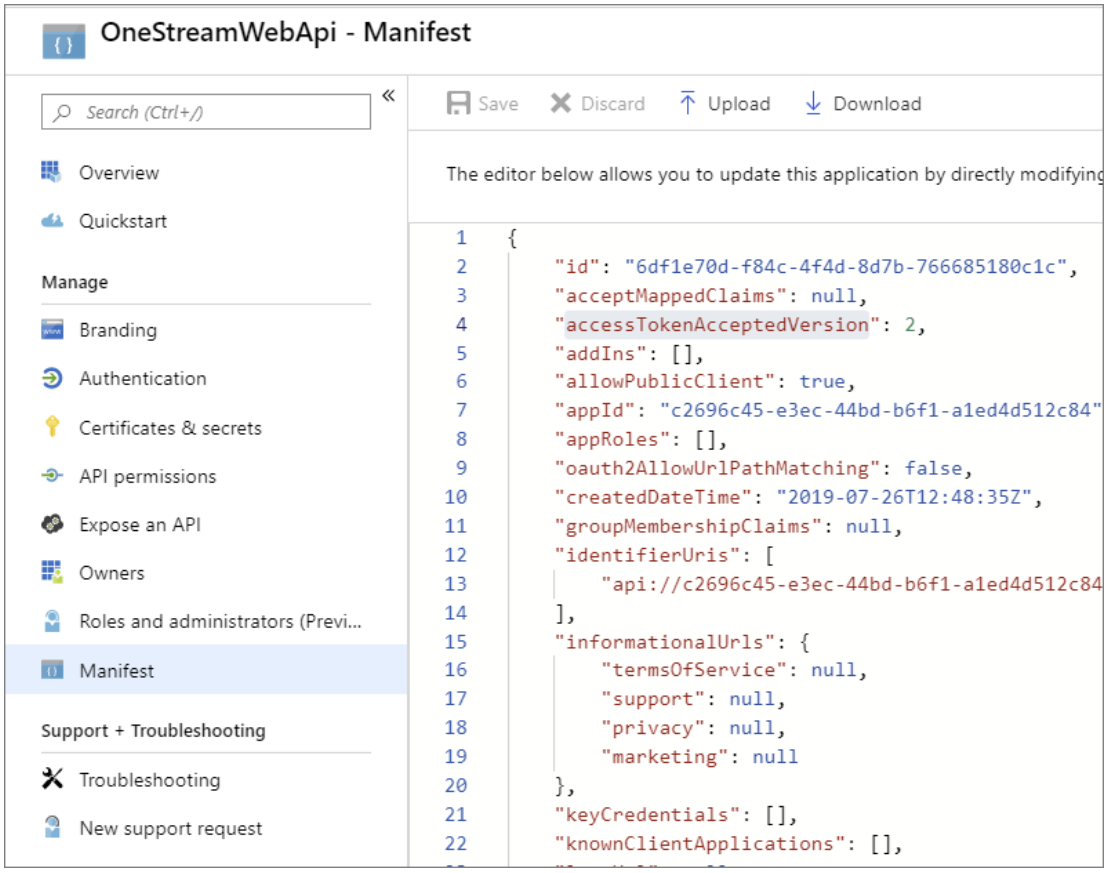
Configure OneStream API for External Authentication

We support Azure AD, Okta, and Ping Federate for authenticating the REST API. Follow the instructions below based on the authentication provider you are using.

Azure AD Configuration

Create Application Registration in Azure

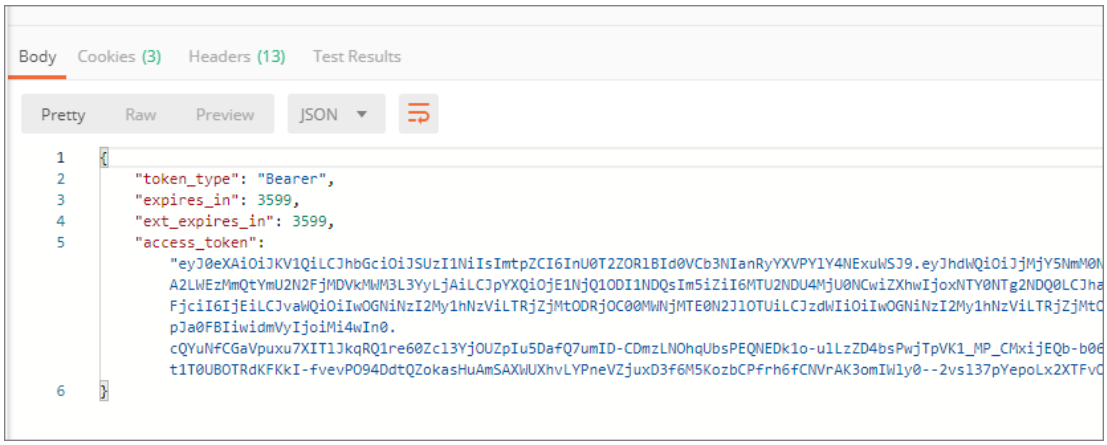
- Sign-in to the Azure portal.
- In the left-hand navigation pane, select the **Azure Active Directory** service, and then select **App registrations > New registration**.
- When the **Register an application** page appears, enter your application's registration name. Click Register.
- In application's **Overview** tab, note {Client Id}, {Tenant Id}
- In **Authentication** tab, Advanced settings, check boxes for Access and ID tokens. In Default Client Type, select Yes for Treat application as a public client. In Supported Account types, select **Accounts in this organizational directory only (Default Directory)**. Save.
- In **Certificates & secrets**, add **New client secret** and note the value. Save.
- In **Expose an API tab**, add a custom scope needed for user-machine use case. Note the scope name and the {AppId Uri} values. Save.
- Our implementation supports v2.0 Azure endpoints, so in Manifest tab, find accessTokenAcceptedVersion. Set value to 2 if different. Save.



Setup Postman for access_token requests

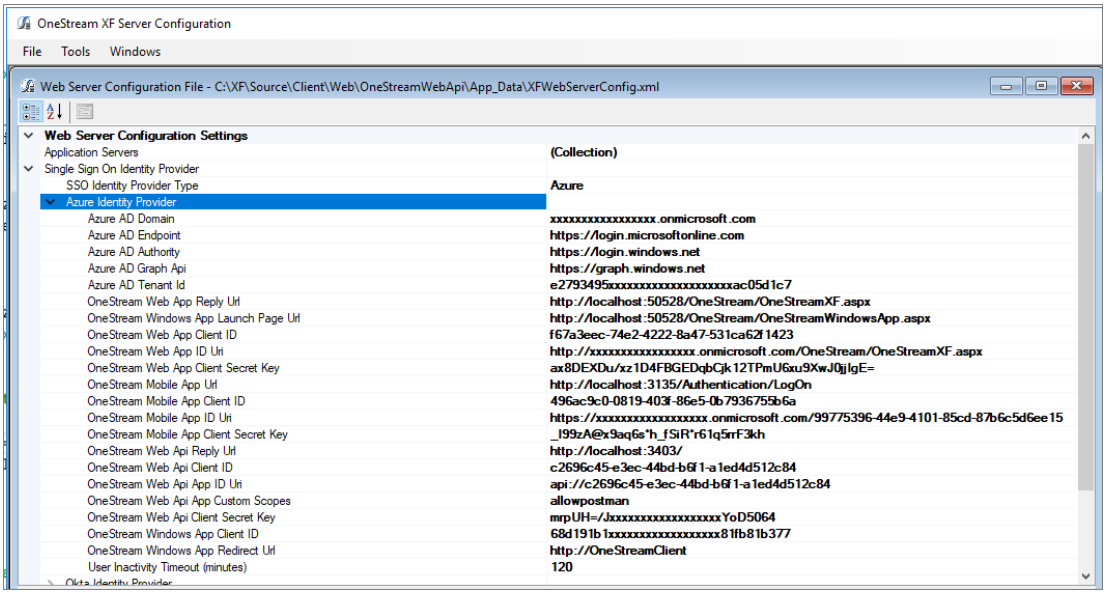
- Create a new POST request. Set url to https://login.microsoftonline.com/{Tenant Id}/oauth2/v2.0/token with tenantid value from #4 above
- In Authorization tab, select Basic Auth for type. In Username and Password fields enter respectively ClientId and Client secret from the app registration section above
- In Headers tab, enter the following keys:
    1. Accept:application/json
    2. Authorization:Basic
    3. Content-Type:application/x-www-form-urlencoded

- In Body, enter either:
    1. grant_type:client_credentials
    2. scope:[AppId Uri]/.default for machine to machine use case or
    3. grant_type:password
    4. username:[Azure AD user name]
    5. password:[Azure AD user password]
    6. scope:[custom scope]
- Click Send and notice the value of access_token in the response



Update the Server Config Utility

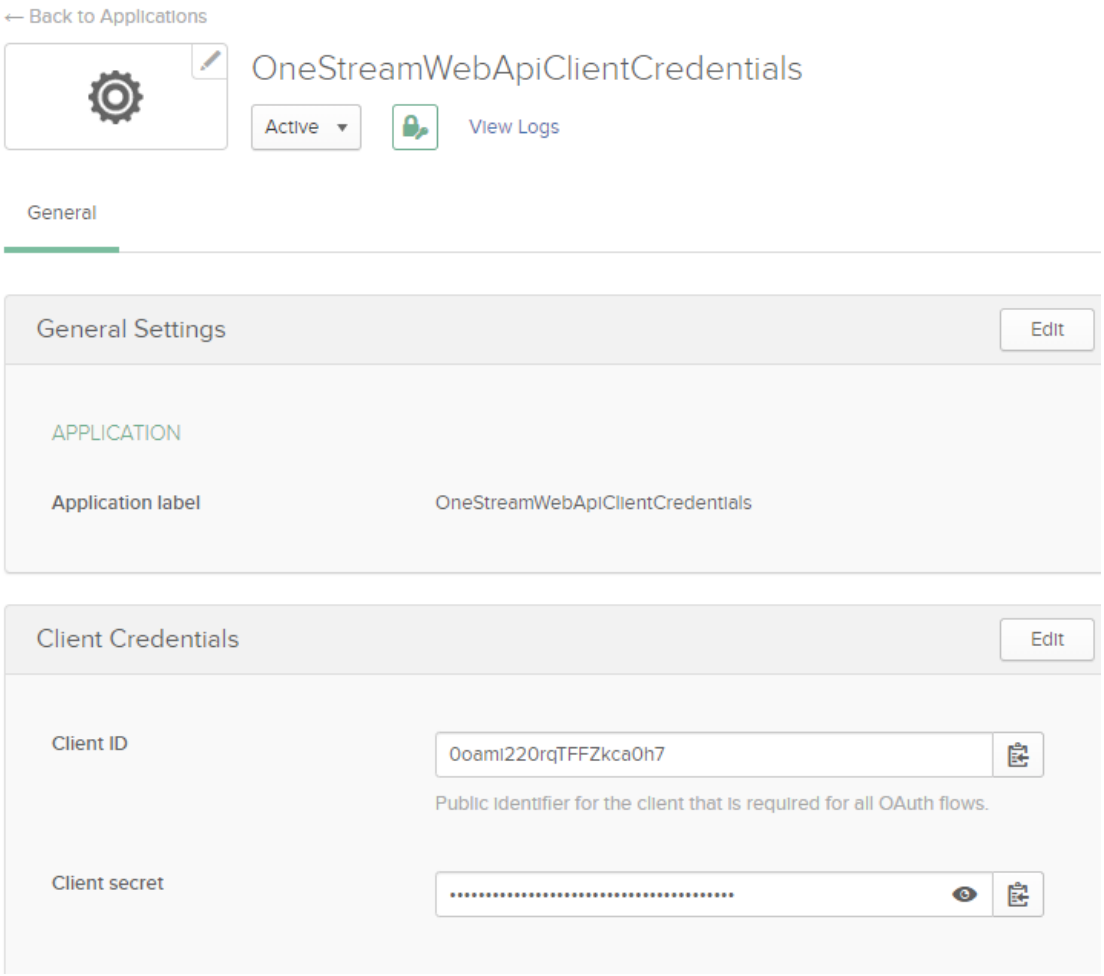
Next go to the Server config utility and enter the values for the four Web Api properties.



Okta Configuration

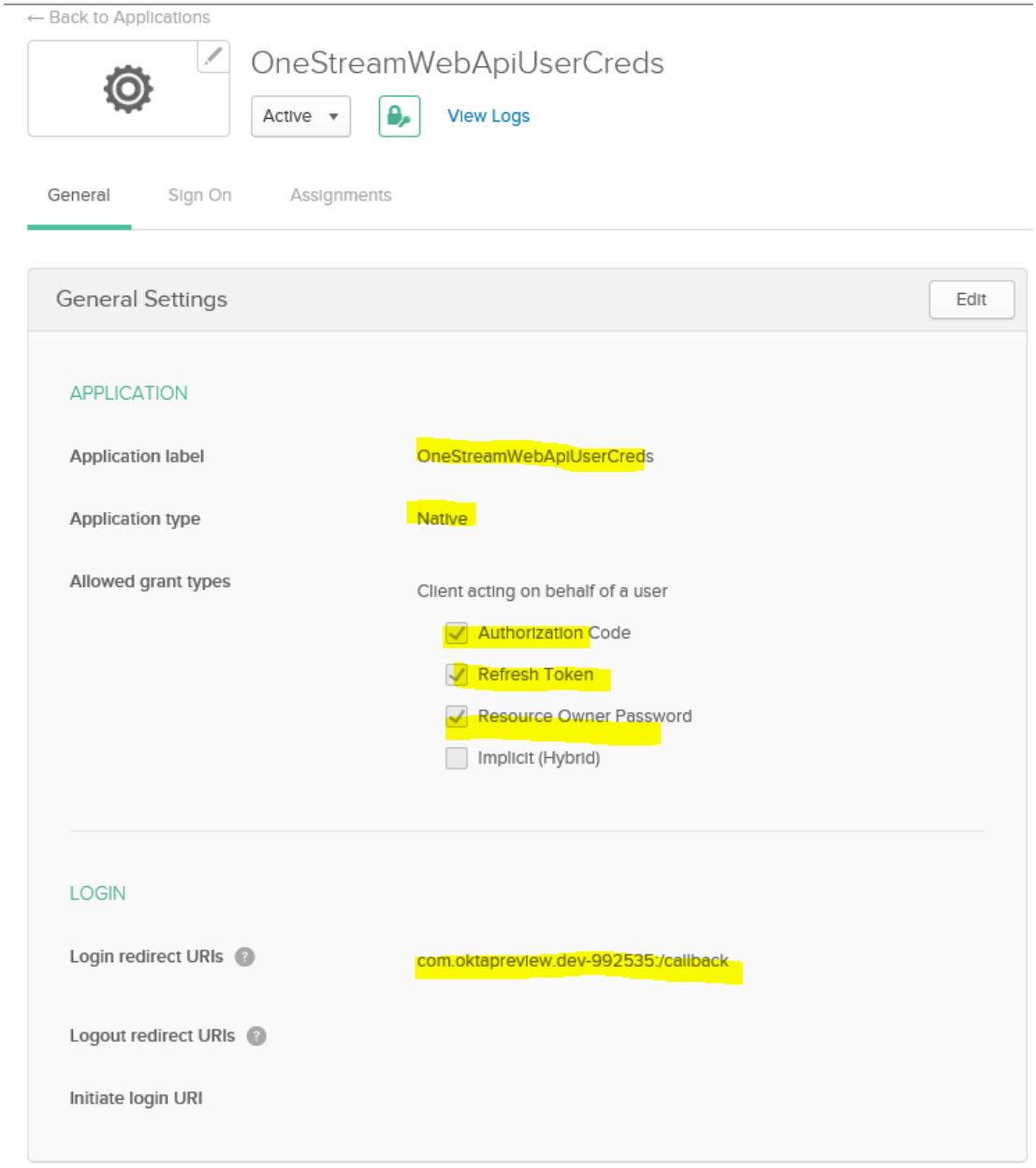
Create M2M Application Registration (grant_type = client_credentials)

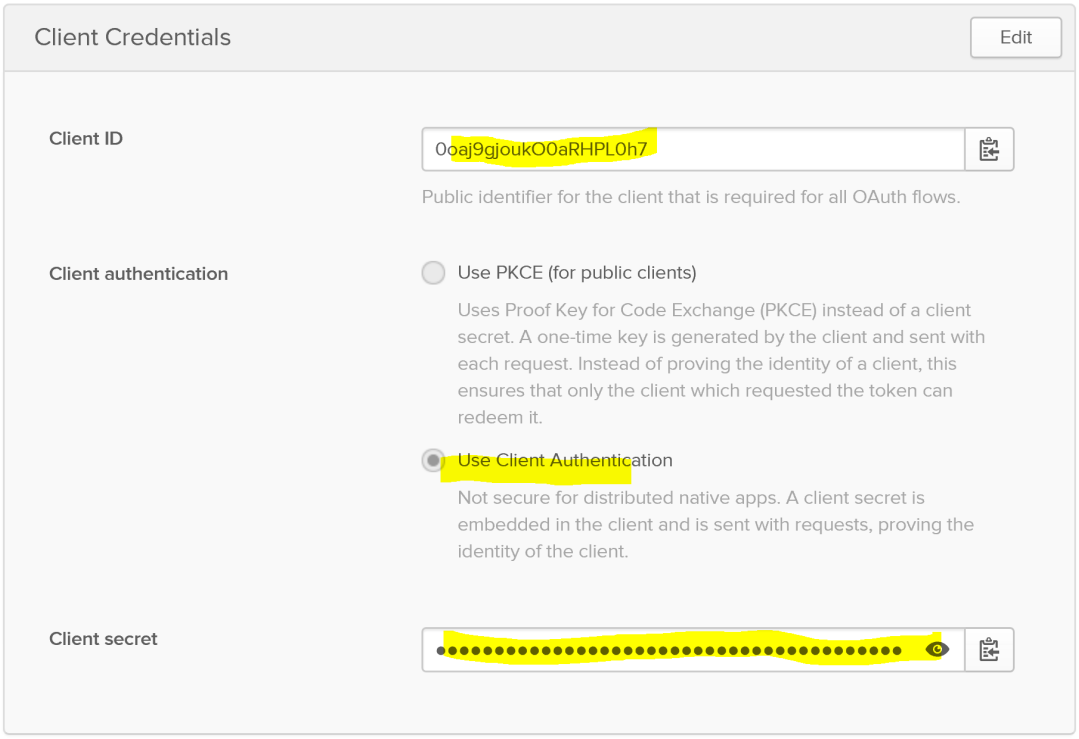
- Create a new OAuth Service App.
- Add a name, then click OK
- Take note of the Client ID and Client Secret. Copy values and save for later.



Create U2M Application Registration (grant_type = password) in Okta

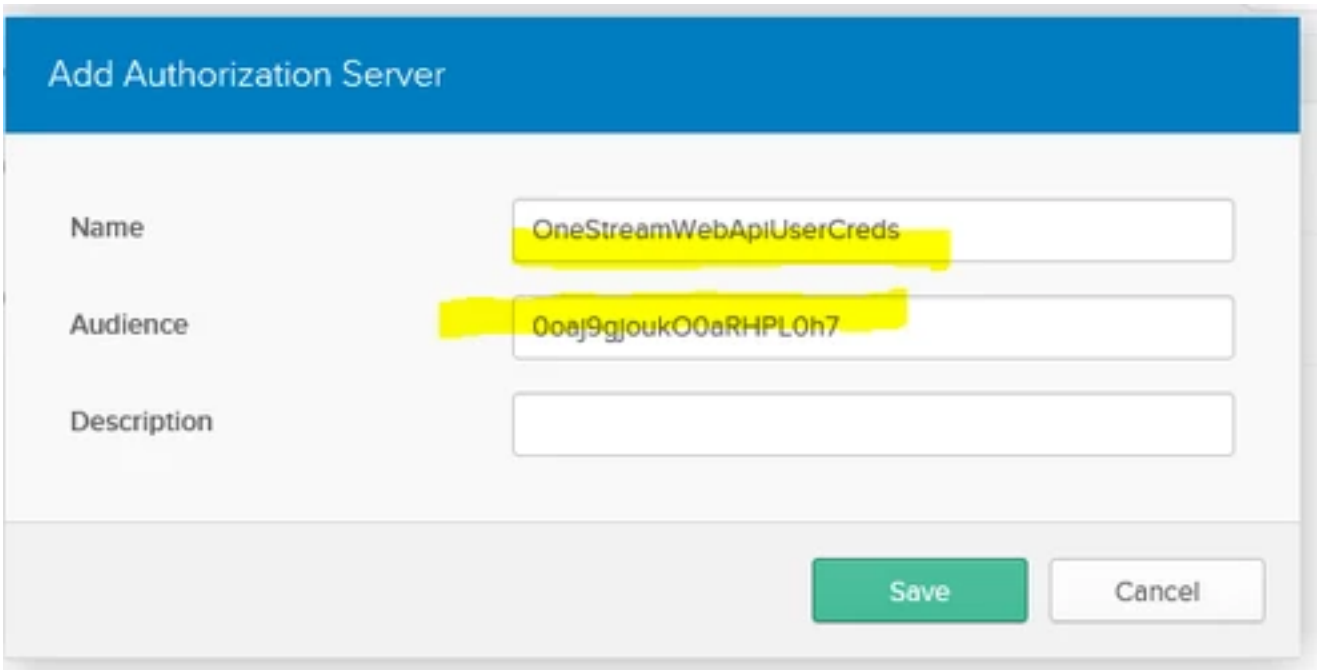
- Create a Native App.
- Give it a Label and make sure the allowed grant types are checked.
- Also take note of the Logout Redirect URIs, Client ID, and Client Secret. Copy values and save for later.
- Make sure Use Client Authentication is selected.

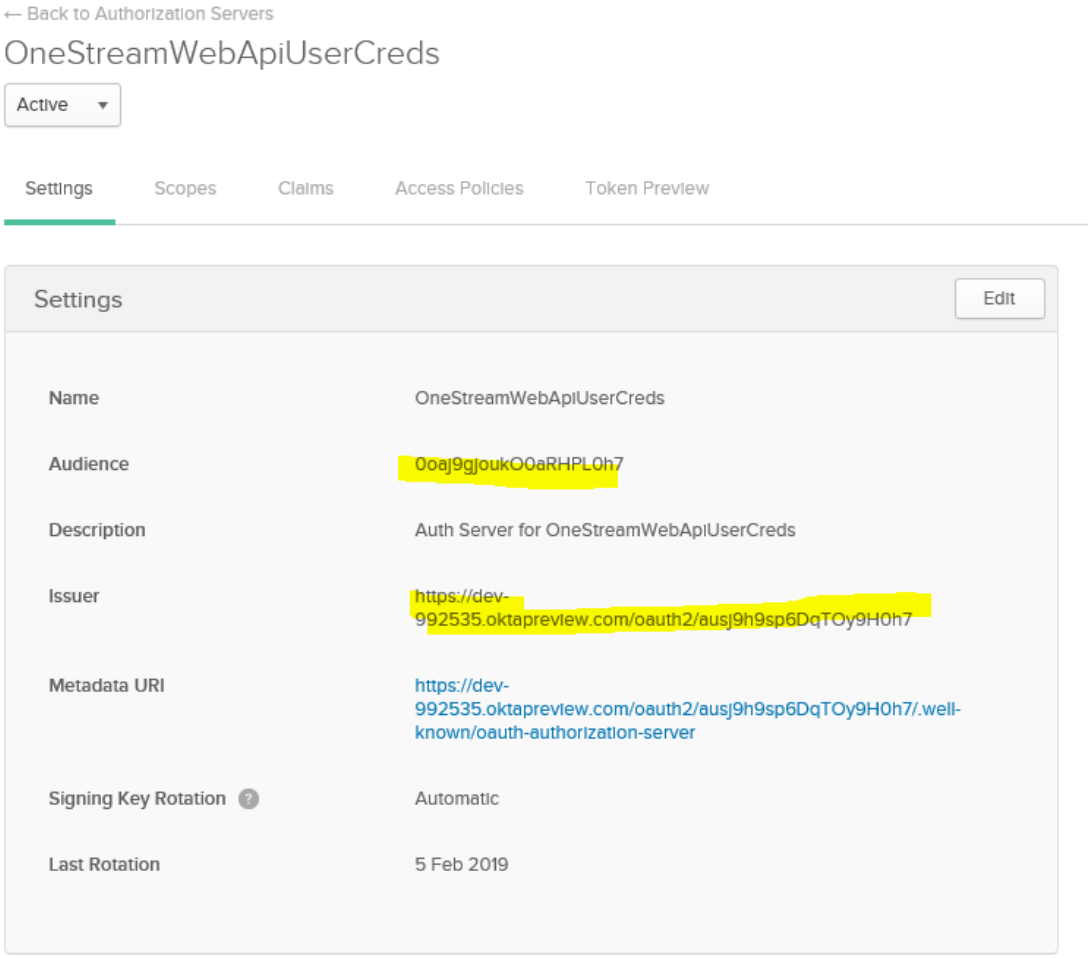

← Back to Applications
OneStreamWebApiUserCreds
Active
View Logs
General
Sign On
Assignments
General Settings
Edit
APPLICATION
Application label
OneStreamWebApiUserCreds
Application type
Native
Allowed grant types
Client acting on behalf of a user
Authorization Code
Refresh Token
Resource Owner Password
Implicit (Hybrid)
LOGIN
Login redirect URIs
com.oktapreview.dev-992535:/callback
Logout redirect URIs
Initiate login URI



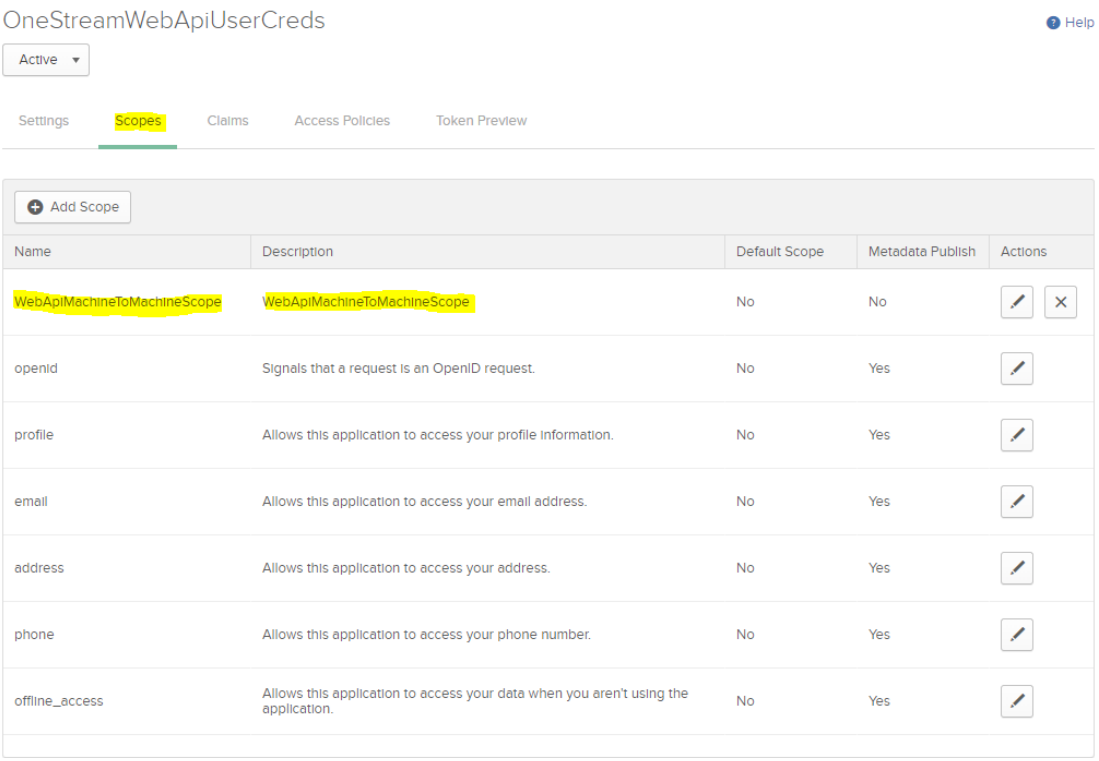
Next go to the API > Authorization servers.

Click on Add Authorization Server. Give it a name and make Client ID the Audience. Click Save.


Add Authorization Server
Name
OneStreamWebApiUserCreds
Audience
0oaj9gjoukO0aRHPL0h7
Description
Save
Cancel

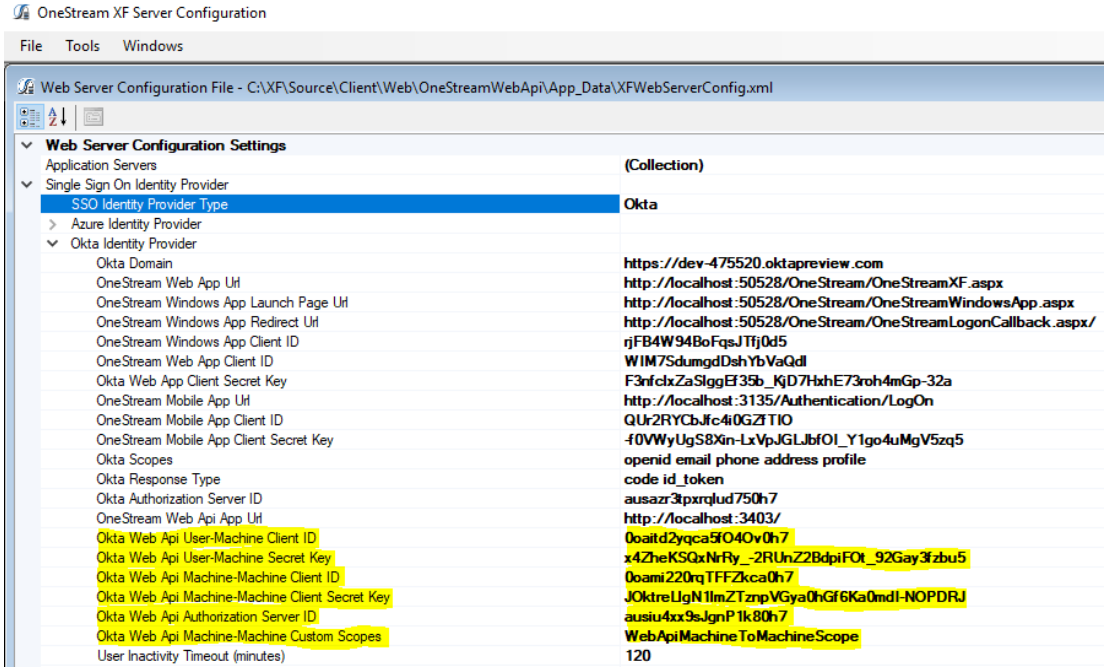


Add a custom scope that will be needed for the Machine-to-Machine scenario. Example:



Update the Server Config Utility

Next go to the Server config utility and enter the values for the four Web Api properties. Save and restart IIS.

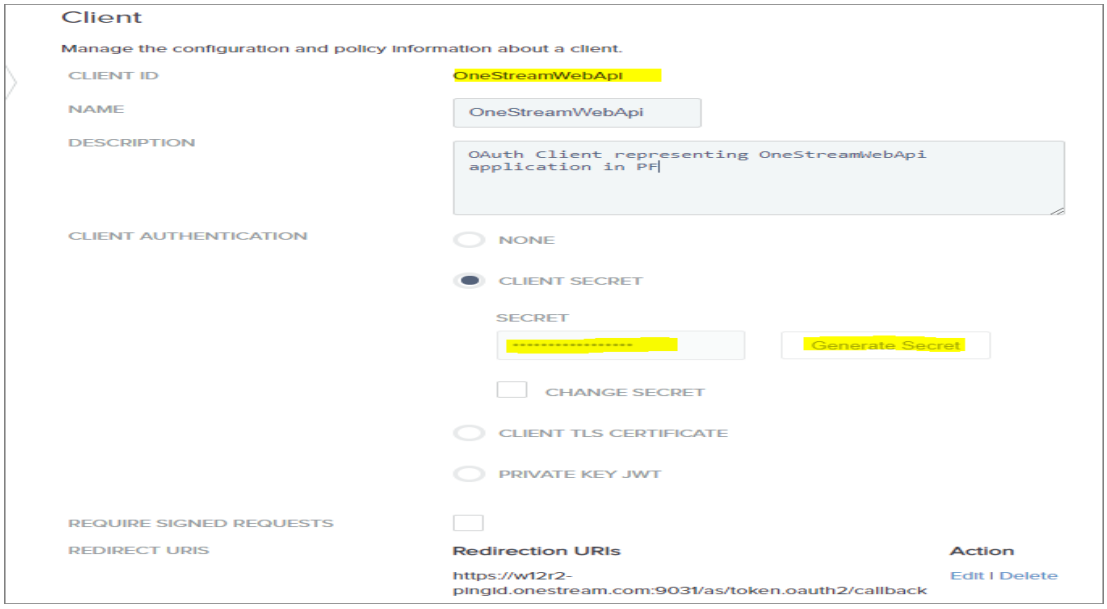


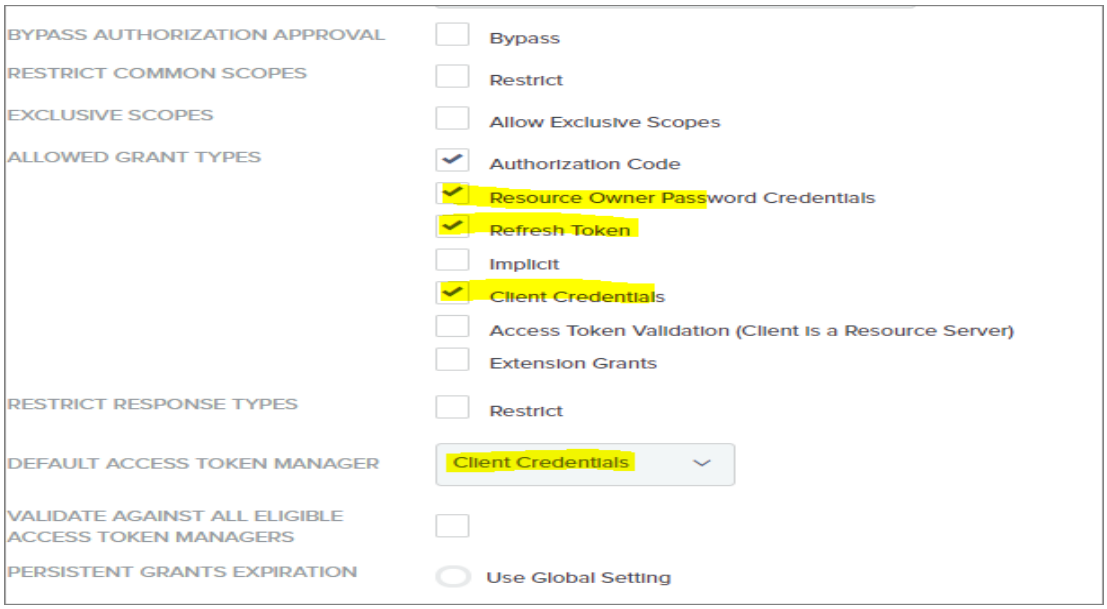
PingFederate Configuration

Add a new client in PF Admin Console representing OneStreamWeb Api application.

Create U2M Application Registration (grant_type = password) in PingFederate

- Create a new Access Token Manager (ATM) and take note of the Access Token Manager ID value (Client Credentials in this instance). See Appendix 9.10 for instructions on how to setup a ATM in Ping Admin console.
- Set Client Authentication to Client Secret option. Click Generate Secret, then Update. Take note of both Client ID and Client Secret values. In Allowed Grant Types, check Authorization Code, Resource Owner Password Credentials, Client Credentials.
- Set Default Access Token Manager to the value noted above #1.
- Save





Update the Server Config Utility

Next go to the Server config utility and enter the values for the four Web API properties. Save and restart IIS.

Configure the AUD Value

In certain PingFederate installations, the Audience value is not used in the authentication process. Normal processing will cause authentication to fail if this value is not used. The **Validate Audience** option allows for disabling audience validation for these installations.

By default, this setting is **True**, which means the audience will be validated.

1. In the Web Server Configuration file, select **Single Sign On Identity Provider**.
2. In PingFederate Identity Provider, click the ellipsis for more details.
3. In Validate Audience, select **False** to disable Audience validation.