



REST API

Implementation Guide

8.1.3 Release

Copyright © 2023 OneStream Software LLC. All rights reserved.

Any warranty with respect to the software or its functionality will be expressly given in the Subscription License Agreement or Software License and Services Agreement between OneStream and the warrantee. This document does not itself constitute a representation or warranty with respect to the software or any related matter.

OneStream Software, OneStream, Extensible Dimensionality and the OneStream logo are trademarks of OneStream Software LLC in the United States and other countries. Microsoft, Microsoft Azure, Microsoft Office, Windows, Windows Server, Excel, .NET Framework, Internet Information Services, Windows Communication Foundation and SQL Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. DevExpress is a registered trademark of Developer Express, Inc. Cisco is a registered trademark of Cisco Systems, Inc. Intel is a trademark of Intel Corporation. AMD64 is a trademark of Advanced Micro Devices, Inc. Other names may be trademarks of their respective owners.

Table of Contents

Introduction	1
REST API Overview	2
OneStream Web API Endpoints	2
Authentication	2
DataManagement	2
DataProvider	2
Call State For Long Running Requests	3
Authentication	4
Application API	6
Data Provider API v7.2.0	8
OneStream REST API Implementation	11
OneStream WebAPI Endpoints	11
Configure OneStream API for External Authentication	30
Azure AD (Microsoft Entra ID) Configuration	30
Configure the REST API Application Registration in Azure AD (Microsoft Entra ID)	30
Set Up the Web Server Configuration in OneStream	32
Configure the User in OneStream	34
Azure AD (Microsoft Entra ID) Endpoints	35

Set Up Postman Access Token Requests	36
Okta Configuration	37
Configure the REST API Application Registration in Okta	37
Add Authorization Servers and Scopes in Okta	38
Set Up the Web Server Configuration in OneStream	39
Configure the User in OneStream	41
PingFederate Configuration	41
Configure the REST API Application Registration in PingFederate	42
Set Up the Web Server Configuration in OneStream	43
Configure the User in OneStream	45
Configure the AUD Value	46

Introduction

This guide provides information about the implementation, authentication, and application programming interfaces available to extend OneStream functionality.

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

For customers in a self-hosted environment, Web API must be installed on a web server and configured for external authentication providers supporting OAuth2.0/OpenID Connect authorization protocol. Identity providers currently supported are Azure AD (Microsoft Entra ID), Okta, and PingFederate.

All customers in a OneStream-hosted environment using authentication with OneStream IdentityServer, including those using native authentication and any OIDC or SAML 2.0 compliant external identity providers, can use personal access tokens (PATs) to access OneStream REST API. See the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer and using PATs.

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

In this topic:

- [OneStream Web API Endpoints](#)
- [OneStream REST API Implementation](#)
- [Configure OneStream API for External Authentication](#)

OneStream Web API Endpoints

URLs are relative to query parameter api-version=5.2.0, unless otherwise noted.

Authentication

Authentication endpoint. Represents a RESTful service for Authentication.

- POST api/Authentication/LogonAndReturnCookie
Used primarily to verify Web API installation completed successfully. Returns an authentication message 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.**

Call State For Long Running Requests

To prevent proxy appliance time-out, a polling method was introduced for long running requests. When this is enabled, all requests use the polling method based on the configured setting in the XFAppServerConfig.xml indicating how long the request has to complete. This allows long running requests to complete without the proxy appliances returning a 502 Bad Gateway as a response to the request inactivity that causes the proxy to terminate the connection.

How It Works

XFCallState polling must first be enabled in the XFAppServerConfig.xml in the EnvironmentSettings block.

File: XFAppServerConfig.xml

```
1 <EnvironmentSettings EnvironmentName="Engineering">
2   <EnvironmentColor>Green</EnvironmentColor>
3   <CanUseClientUpdater>true</CanUseClientUpdater>
4   <CanUseAdministratorUser>true</CanUseAdministratorUser>
5   <UseDetailedErrorLogging>true</UseDetailedErrorLogging>
6   <EnableHelp>true</EnableHelp>
7   <EnableFileShareUploads>true</EnableFileShareUploads>
8   <UseCallStateForLongRunningRequests>false</UseCallStateForLongRunningRequests>
9   <CallStateNetworkTimeoutNumSeconds>120</CallStateNetworkTimeoutNumSeconds>
10  <EnableAzureRelay>false</EnableAzureRelay>
```

The configured values of `UseCallStateForLongRunningRequests` and `CallStateNetworkTimeoutNumSeconds` manage the `XFCallState` functionality. When `UseCallStateForLongRunningRequests` is set to `true`, call state polling is used. The configured value for `CallStateNetworkTimeoutNumSeconds` determines the time to wait before using call state to complete the request, default 120 seconds.

Authentication

To secure REST API with OAuth 2.0 for customers in a self-hosted environment, configure authentication with one of these supported external providers:

- [Azure AD \(Microsoft Entra ID\) Configuration](#)
- [Okta Configuration](#)
- [PingFederate Configuration](#)

For customers in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer and using personal access tokens (PATs).

Access tokens from any of the above providers have short expiration times. To avoid copying the entire token value to the Authorization/Token text box, 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);
```

Authentication API

Method	Endpoint	Description
Post	Authentication/Logon	Logs on and returns a SessionInfo (SI) object for use with other Rest API calls that accept an SI as an argument. This endpoint performs a logon only and does not open an application. This is the equivalent of entering login credentials in the Desktop App before selecting and opening an application.

Authentication/Logon

POST https://{BaseWebServer}/api/Authentication/Logon?api-version=7.2.0

Query Parameters

Key	Value	Required
api-version	7.2.0	Yes

Authorization

Type	Value	Required
Bearer Token	(your access token)	Yes

Headers

Key	Value	Required
Content-Type	application/json	Yes

Request Body

Key	Type	Description	Required
BaseWebServerURL	string	Your URL for the web service	Yes

REST API Overview

Sample Request

```
{
  "BaseWebServerUrl": "https:// golfstream.onestreamcloud.com/OneStreamWeb"
}
```

Sample Response

```
{
  "Message": "Logon succeeded.",
  "Logon_SessionInfo": {
    "XfBytes": " QB8AACNodHRwOi8vbG9jYWxob3N0OjUwMDAxL09uZVN0cmVhbVdlYhQAAAB7izp1jCP3BUVr8bjD2f6KmmL5BKzhOVWUZU1MikEYOVek0ZUIT0tUQV9NMk27tnn6+VZaR544CK1YPCFeWSBWCTmQ2ggAAAAAAAAAAAAAAAFZW4tVVMAAAAAAAAAAAAAAAAAAAAAP//////////8P//////////AwAAABn8//8Z/P//Gfz//xn8//8Z/P//Gfz//xn8//8Z/P//Gfz//w=="
  },
  "Authorized applications": [
    "GolfStreamDemo_2022",
    "OFC_ECA_ProductMgmt",
    "OneStream_GolfStream"
  ]
}
```

Application API

Method	Endpoint	Description
Post	Application/OpenApplication	Opens specified application. Requires a valid sessionInfo token obtained from the Authentication/Logon method.

Application/OpenApplication

POST https://{BaseWebServer}/api/Application/OpenApplication?api-version=7.2.0

Query Parameters

Key	Value	Required
-----	-------	----------

REST API Overview

api-version	7.2.0	Yes
-------------	-------	-----

Authorization

Type	Value	Required
Bearer Token	(your access token)	Yes

Headers

Key	Value	Required
Content-Type	application/json	Yes

Request Body

Key	Type	Description	Required
ApplicationName	string	Name of the application to open	Yes
SI	array (bytes)	The SessionInfo (SI) object obtained from Authentication/Logon endpoint.	Yes

Sample Request

```
{
  "ApplicationName": "GolfStreamDemo_2022",
  "SI": {
    "XfBytes": "QB8AACNodHRwOi8vbG9jYWxob3N0QjUwMDAxL09uZVN0cmVhbVd1YhQAAAB7izp1jCP3BUVr8bjD2f6KmmL5BKzhOVWUzU1MikeYOVekOZUIT0tUQV9NMk27tnn6+VZaR544CK1YPCFeWsbWCTmQ2ggAAAAAAAAAAAAAAAAAAAAAAAFZW4tVVMAAAAAAAAAAAAAAAAAAAAAAAP////////////////////////////////////////8P////////////////////////////////AwAAABn8//8Z/P//Gfz//xn8//8Z/P//Gfz//xn8//8Z/P//Gfz//xn8//8Z/P//Gfz//w=="
  }
}
```

Sample Response

```
{
  "Message": "Open application succeeded.",
  "Application SessionInfo": {
    "XfBytes": "QB8AACNodHRwOi8vbG9jYWxob3N0OjUwMDAxL09uZVN0cmVhbVd1YhQAAAAep0GewgsakcN4GJDmuwyaIMazfN/aHyhnXNLgg+hUxy6cpQIT0tUQV9NMk27tnn6+VZaR544CK1YPCFe0BusL1iM2ggUAAAArL9Q04ePExHJxVU89Y1MAeNxr8UT25lU3RyZWFTX0dvdGZTdHJlYW3xShfEXWxvRb0x2hWDSCd0BwVuLVVTAIAAAAAAAAAAAAAAAAAAAAAAAD////////wAAAAACAFABAABQAFD///8AAAAAYHddwMAAABCAfAAGfz//5z///+c////FQAQACYAIAARAGAAAwCQABn8//8Z/P//Gfz//xn8//8="
  }
}
```

Data Provider API v7.2.0

Method	Endpoint	Description
Post	DataProvider/ GetadoDataSetForAdapter	Executes a Data Provider HTTP Post request and returns a JSON representation of a DataSet for a given Dashboard Adapter. Requires a SessionInfo (SI) object obtained from Application/OpenApplication endpoint.

DataProvider/GetAdoDataSetForAdapter

POST https://{BaseWebServer}/api/DataProvider/GetAdoDataSetForAdapter?api-version=7.2.0

Query Parameters

Key	Value	Required
api-version	7.2.0	Yes

Authorization

Type	Value	Required
Bearer Token	(your access token)	Yes

Headers

Key	Value	Required
Content-Type	application/json	Yes

Request Body

Key	Type	Description	Required
IsSystemLevel	boolean	An indication of whether the Dashboard Adapter is defined at the System Level (True) or for the specified Application (False).	Yes
AdapterName	string	The name of the Dashboard Adapter used for data retrieval.	Yes
ResultDataTableName	string	Name of the resulting table in the DataSet	Yes
CustomSubstVarsAsCommaSeparatedPairs	string	Comma separated list of Variable name/value pairs requiring a user prompt. These must be specified using the following format: "VariableName1=[VariableValue1],VariableName2=[VariableValue2],...".	No
SI	array (bytes)	The SessionInfo (SI) object obtained from Application/OpenApplication endpoint.	Yes

Sample Request

```
{
  "IsSystemLevel": true,
  "AdapterName": "Sales Mix (WF)",
  "ResultDataTableName": "ResultsTable",
  "CustomSubstVarsAsCommaSeparatedPairs": "",
  "SI": {
    "XfBytes": " QB8AACNodHRwOi8vbG9jYVxob3N0OjUwMDAxL09uZVN0cmVhbVdlYhQAAAAep0GewgsakcN4GJDmuwyaaIMazfN/aHyhnXNLgg+hUxy6cpQIT0tUQV9NMk27tnn6+VZaR544CK1YPCFe0BusL1iM2ggUAAArL9Q04ePExHJxVU89Y1MAeNxrh8UT25lU3RyZWftX0dzbGZTdHJlYW3xShfEXWxvRb0x2hWDSCd0BWVuLVVTAAAAAAd////////wAAAAACAFABAABQAFD///8AAAAAYHddwMAAABCAfAAGfz//5z///+c///FQAQACYAIAARAGAAAwCQABn8//8Z/P//Gfz//xn8//8="
  }
}
```

Sample Response

```
{
  "ResultsTable": [
    {
      "RowId": 0,
      "RowName": "Row1",
      "PovCubeNameAndDesc": "GolfStream - Corporate",
      "Pov00EntityNameAndDesc": "Total GolfStream",
      "Pov02ScenarioNameAndDesc": "Actual - Actual",
      "Pov03TimeNameAndDesc": "2011M2 - Feb 2011",
      "Pov04ViewNameAndDesc": "YTD",
      "RowHdr0NameAndDesc": "Drivers",
      "RowHdr0Indent": 0,
      "Col0Hdr0NameAndDesc": "60000 - Operating Sales",
      "Col0Hdr0Indent": 0,
      "Col0Value": 25552270.482000000000000000,
      "Col0ValueAsText": "25,552,270.48"
    },
    {
      "RowId": 1,
      "RowName": "Row1",
      "PovCubeNameAndDesc": "GolfStream - Corporate",
      "Pov00EntityNameAndDesc": "Total GolfStream",
      "Pov02ScenarioNameAndDesc": "Actual - Actual",
      "Pov03TimeNameAndDesc": "2011M2 - Feb 2011",
      "Pov04ViewNameAndDesc": "YTD",
      "RowHdr0NameAndDesc": "Fairway Woods",
      "RowHdr0Indent": 0,
      "Col0Hdr0NameAndDesc": "60000 - Operating Sales",

```

```
        "Col0Hdr0Indent": 0,  
        "Col0Value": 17476089.966000000000000000,  
        "Col0ValueAsText": "17,476,089.97"  
    }  
]  
}
```

OneStream REST API Implementation

In this topic:

- [Authentication](#)
- [OneStream WebAPI Endpoints](#)

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 ],
```

```
"WorkspaceName": [your workspace name], - Optional
"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.

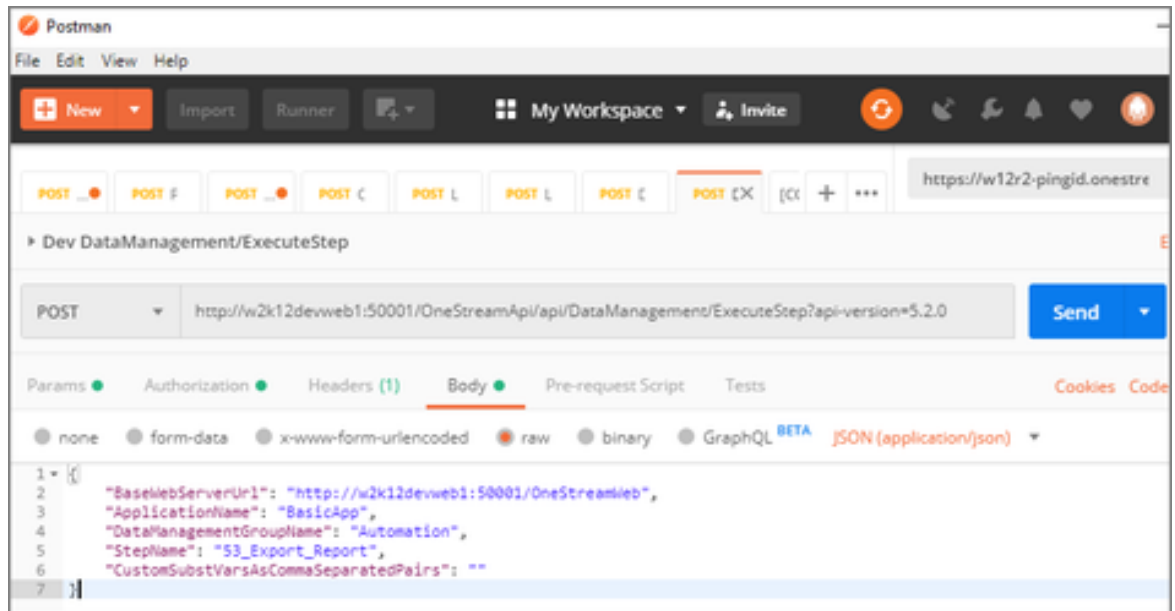
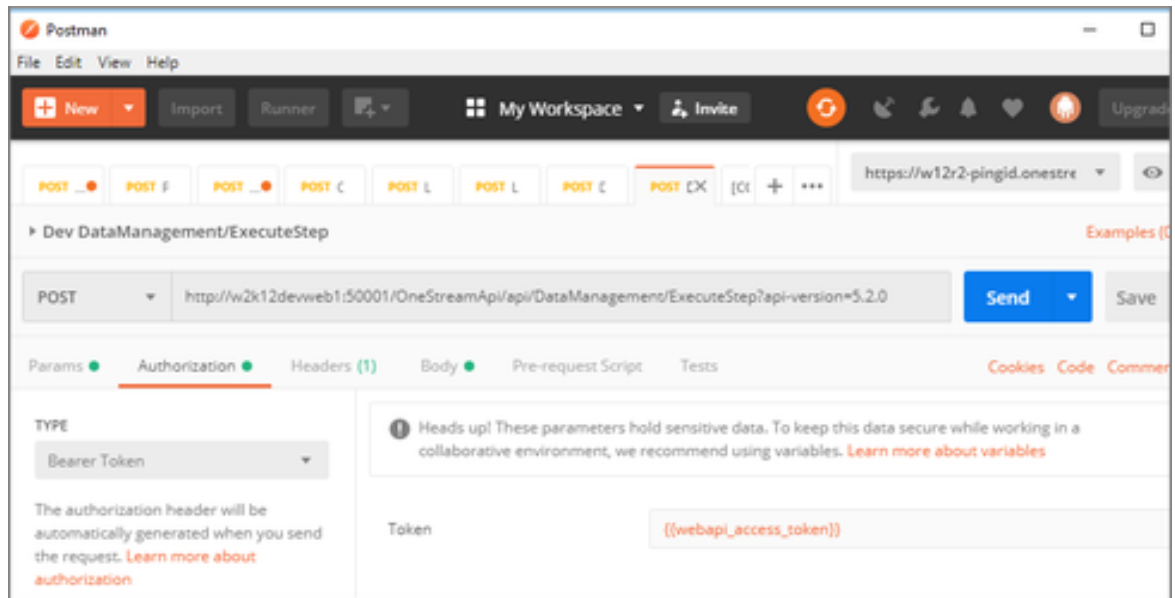
Data Management Execute Step endpoint

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.

REST API Overview



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 boolean 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",
"TimeFilterForReqFreq": "",
"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 message that indicates authentication state. Used mostly 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 message 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.

REST API Overview

POST ▼ http://localhost:3403/api/Authentication/LogonAndReturnCookie?api-version=5.2.0

Params ● Authorization ● Headers (4) Body ● Pre-request Script Tests

TYPE

Bearer Token ▼

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

! Heads up! These parameters hold sensitive data. To keep this data secure while working in a recommend using variables. [Learn more about variables](#)

Token

{{webapi_access_token}}

POST ▼ http://localhost:3403/api/Authentication/LogonAndReturnCookie?api-version=5.2.0

Params ● Authorization ● Headers (4) Body ● Pre-request Script Tests

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL BETA

JSON (application/json) ▼

1 {

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

3 "ApplicationName": "GolfStream_v37"

4 }

REST API Overview

The screenshot displays a REST client interface with a POST request to the URL `https://archqa1.onestreamtest.com/OneStreamApi/api/Authentication/LogonAndReturnCookie?api-version=5.2.0`. The interface includes tabs for Params, Authorization, Headers (13), Body, Pre-request Script, Tests, and Settings. The Tests tab is active, showing two test steps: `1 var data = pm.response.json();` and `2 pm.environment.set("webapi_access_token", data.access_token);`. Below the request, the response is shown in the Body tab, with sub-tabs for Pretty, Raw, Preview, and Visualize. The Pretty view shows the response text: `1 Authentication succeeded.`

```
POST https://archqa1.onestreamtest.com/OneStreamApi/api/Authentication/LogonAndReturnCookie?api-version=5.2.0
```

Params ● Authorization ● Headers (13) Body ● Pre-request Script Tests ♦ Settings

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

Body Cookies Headers (11) Test Results ●

Pretty Raw Preview Visualize Text ▾

```
1 Authentication succeeded.
```

Configure OneStream API for External Authentication

For customers in a self-hosted environment, we support REST API authentication with Azure Active Directory (Azure AD [Microsoft Entra ID]), Okta, and PingFederate. Perform the configuration for your provider:

- [Azure AD \(Microsoft Entra ID\) Configuration](#)
- [Okta Configuration](#)
- [PingFederate Configuration](#)

For customers in a OneStream-hosted environment, see the *Identity and Access Management Guide* for information about authentication with OneStream IdentityServer and using personal access tokens (PATs).

Azure AD (Microsoft Entra ID) Configuration

To configure OneStream REST API to support Azure AD (Microsoft Entra ID) authentication, follow these steps:

1. [Configure the REST API Application Registration in Azure AD \(Microsoft Entra ID\)](#).
2. [Set Up the Web Server Configuration in OneStream](#).
3. [Configure the User in OneStream](#).

To enable single sign-on with Azure AD (Microsoft Entra ID) for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, see the *Installation and Configuration Guide*.

Configure the REST API Application Registration in Azure AD (Microsoft Entra ID)

To configure the REST API application registration, you need to copy the application (client) ID from Azure AD (Microsoft Entra ID) and paste it into the Web Server Configuration in OneStream.

Configure OneStream API for External Authentication

1. Log into your Azure AD account.
2. On the Home screen, click the **App registrations** icon.
3. On the **App registrations** page, click the **+ New registration** tab.
4. On the **Register an application** page, complete the following fields:
 - a. Enter a name for the application.
 - b. For **Supported account types**, select **Accounts in this organization directory only**.
5. Click the **Register** button.
6. On the page for the application, in the **Manage** list on the left, select **Authentication**.
7. In the **Advanced settings**, under **Allow public client flows**, set the **Enable the following mobile and desktop flows** option to **Yes**.
8. Click the **Save** button.
9. In the **Manage** list on the left, select **Certificates & secrets**.
10. In the **Client secrets** tab, click **+ New client secret**.
11. In the **Add a client secret** dialog box, enter a description and select an expiration time in the drop-down menu. Click the **Add** button.
12. On the **Certificates & secrets** page, copy the value for the client secret.

IMPORTANT: The client secret value may only be available to copy for a limited time, so copy it immediately after it is created.
13. In the **Manage** list on the left, select **Expose an API**.
14. In **Scopes defined by this API**, click **+ Add a scope**.
15. In the **Add a scope** dialog box, the application ID URI is automatically generated. Click the **Save and continue** button.

NOTE: You can add a scope in this dialog box if needed.

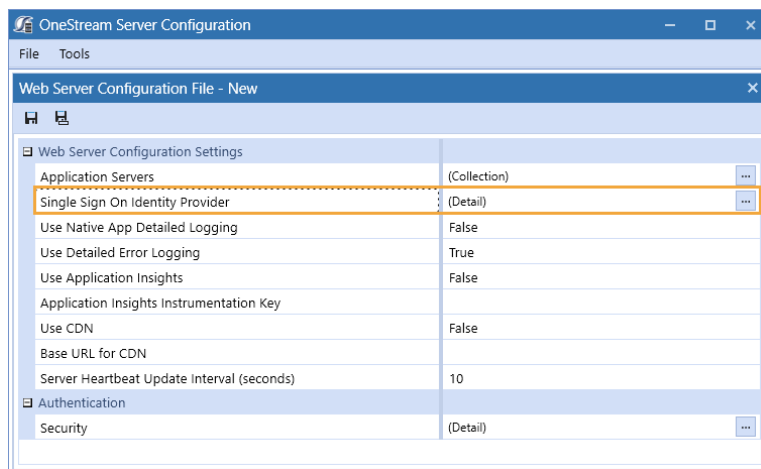
16. On the **Expose an API** page, copy the application ID URI.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.
2. Go to **File > New Web Server Configuration File**.

NOTE: Alternatively, you can open an existing file to edit it.

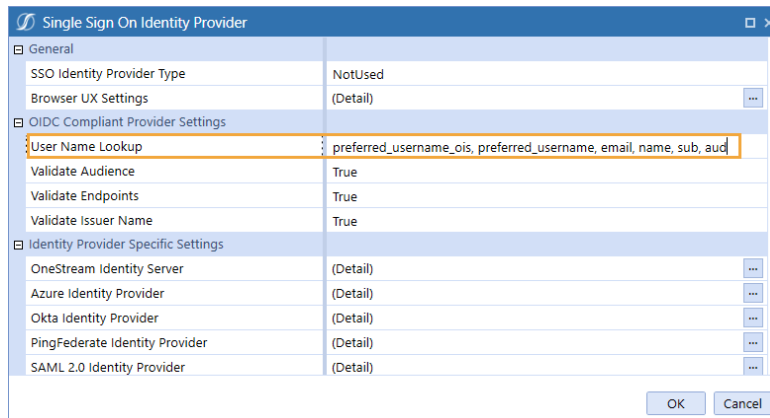
3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.



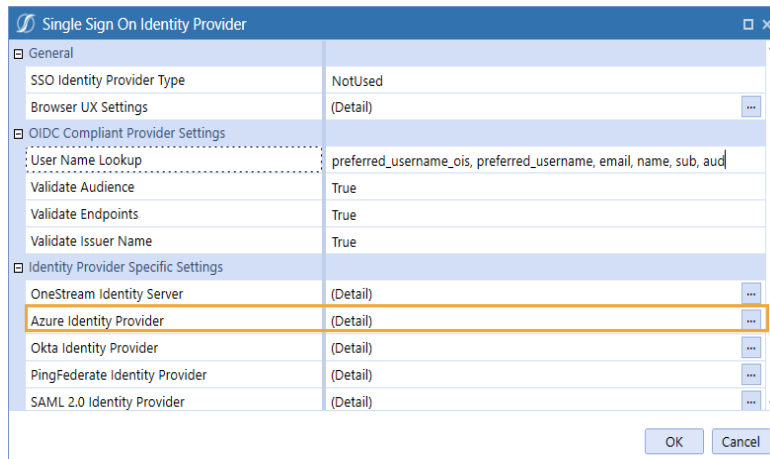
4. In the **User Name Lookup** field, type **aud** to include this claim in the ordered lookups.

NOTE: The claim **aud** indicates the audience that the token is intended for.

Configure OneStream API for External Authentication



5. Click the ellipsis to the right of **Azure Identity Provider**.



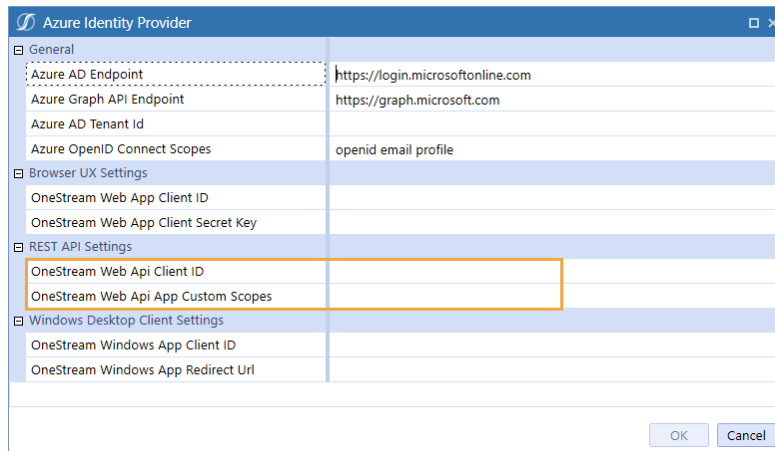
6. In the **Azure Identity Provider** dialog box, in the **REST API Settings** section, complete the following fields :

- **OneStream Web Api Client ID**: Enter the application (client) ID from Azure AD. See [Configure the REST API Application Registration in Azure AD \(Microsoft Entra ID\)](#) step 16.

TIP: To view the application (client) ID in Azure AD, go to the page for the application and select **Overview** in the list on the left.

Configure OneStream API for External Authentication

- **OneStream Web Api App Custom Scopes:** Enter custom scopes, or leave as default (null).



The screenshot shows the 'Azure Identity Provider' configuration window. The 'General' section is expanded, showing fields for 'Azure AD Endpoint' (https://login.microsoftonline.com), 'Azure Graph API Endpoint' (https://graph.microsoft.com), 'Azure AD Tenant Id', and 'Azure OpenID Connect Scopes' (openid email profile). The 'Browser UX Settings' section is also expanded, showing 'OneStream Web App Client ID' and 'OneStream Web App Client Secret Key'. The 'REST API Settings' section is expanded, showing 'OneStream Web Api Client ID' and 'OneStream Web Api App Custom Scopes'. The 'OneStream Web Api App Custom Scopes' field is highlighted with an orange border. The 'Windows Desktop Client Settings' section is also expanded, showing 'OneStream Windows App Client ID' and 'OneStream Windows App Redirect Url'. The 'OK' and 'Cancel' buttons are at the bottom right.

7. Click the **OK** button.
8. Save changes and reset IIS.

NOTE: Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Configure the User in OneStream

1. In the OneStream Desktop application, go to **System > Security > Users > <user>**.
2. In the **Authentication** properties, complete the following fields for REST API authentication through Azure AD.
 - **External Authentication Provider:** In the drop-down menu, select the Azure AD configuration.
 - **External Provider User Name:** Enter the application (client) ID from Azure AD. See [Configure the REST API Application Registration in Azure AD \(Microsoft Entra ID\)](#) step 16.

TIP: To view the application (client) ID in Azure AD, go to the page for the

Configure OneStream API for External Authentication

application and select **Overview** in the list on the left.

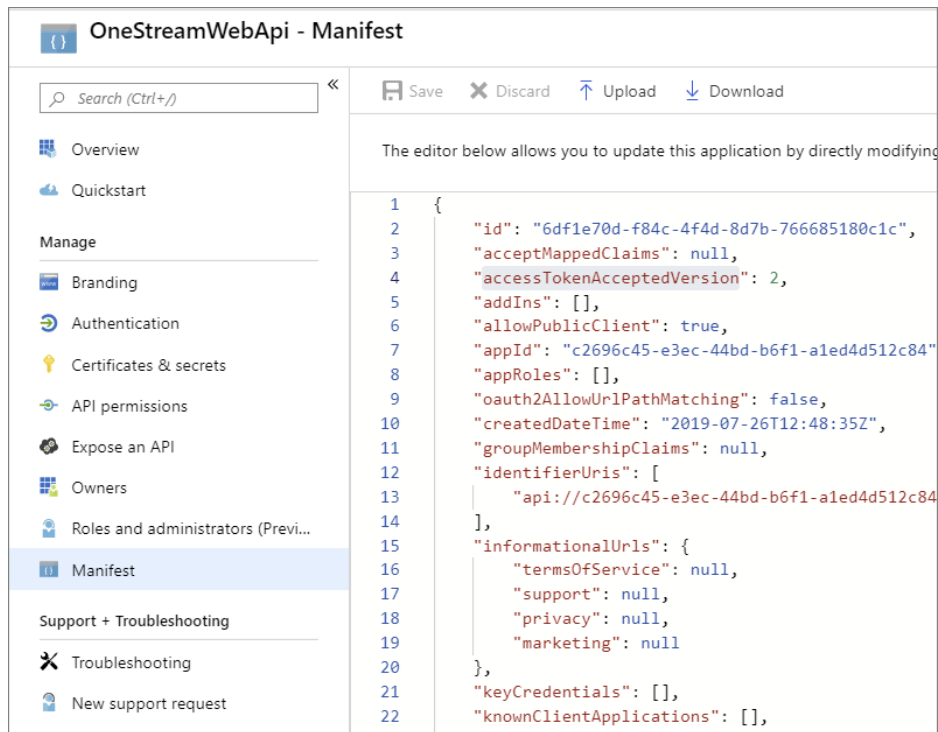
- **Internal Provider Password:** Enter a password.

3. Click the **Save** icon.

Azure AD (Microsoft Entra ID) Endpoints

We support v2.0 Azure AD endpoints.

1. On **Manifest**, find **accessTokenAcceptedVersion**.
2. Set the value to **2**.



3. Click **Save**.

Set Up Postman Access Token Requests

1. Create a new POST request. Set url to `https://login.microsoftonline.com/{TenantId}/oauth2/v2.0/token` with tenant ID value.

TIP: To view the directory (tenant) ID in Azure AD, go to the page for the application and select **Overview** in the list on the left.

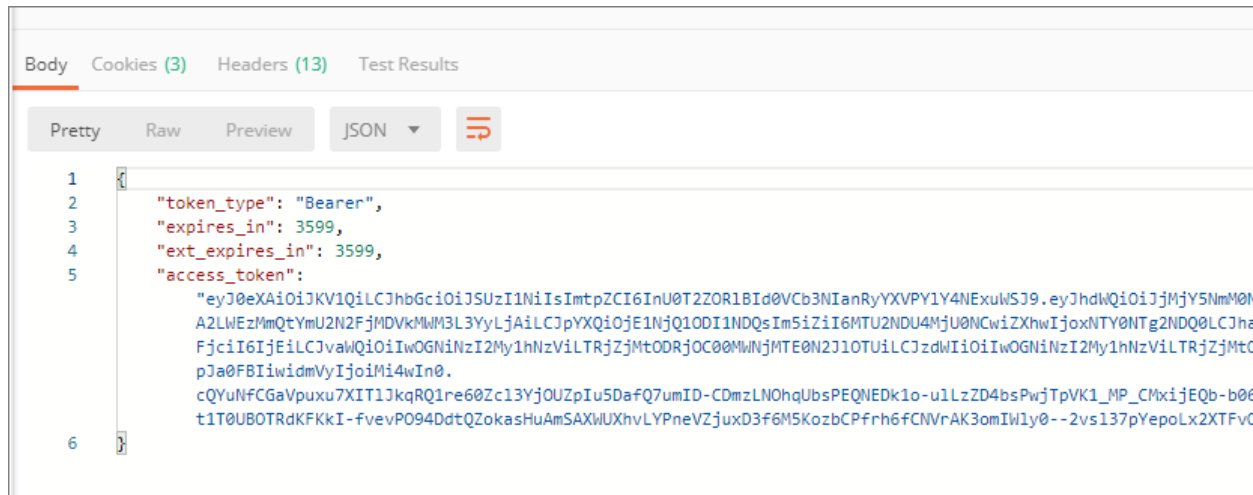
2. In the Authorization tab, select Basic Auth for type. In the Username and Password fields, enter the client ID and client secret from the application registration, respectively. See [Configure the REST API Application Registration in Azure AD \(Microsoft Entra ID\)](#) step 12.

TIP: To view the application (client) ID in Azure AD, go to the page for the application and select **Overview** in the list on the left.

3. In the **Headers** tab, enter the following keys:
 - Accept: application/json
 - Authorization: Basic
 - Content-Type: application/x-www-form-urlencoded
4. In **Body**, enter either option 1 or option 2:
 - a. Option 1:
 - a. grant_type: client_credentials
 - b. scope: {AppId Uri}/.default for machine to machine use case
 - b. Option 2:
 - a. grant_type: password
 - b. username: {Azure ad user name}
 - c. password: {Azure ad user password}
 - d. scope: {custom scope}

5. Click **Send** and notice the value of access_token in the response.

Configure OneStream API for External Authentication



```
1 {
2   "token_type": "Bearer",
3   "expires_in": 3599,
4   "ext_expires_in": 3599,
5   "access_token":
6     "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6InU0T2ZOR1BId0VCb3NIanRyYXVPY1Y4NExuW5J9.eyJhdwQiOiJmYjY5NmM0NA2LWZzZmM0YmU2N2FjMDVhMmM3L3YyLjAiLCJpYXQiOiJlbnQ1ODI1NDQsIm5iZiI6MTU2NDU4MjU0NCwiZXhwIjozNTY0NTg2NDQ0LCJhaFjc0FBIiwidmVyaW91IjoiMi4wIn0.cQYUuNFCGaVpuxu7XIT1JkqRQ1re60Zc13YjOUZpIu5DaFQ7umID-CDmzLN0hqUbsPEQNEk1o-ulLzZD4bsPwjTpVK1_MP_CMxiJfEQb-b06t1T0UBOTRdKFKKI-fvevP094DdtQZokasHuAmSAXwUXhvLYPneVZjuxD3f6M5KozbCPfrh6fCNVrAK3omIWly0--2vs137pYepoLx2XTFvC"
```

Okta Configuration

To configure OneStream REST API to support Okta authentication for M2M application registration (grant_type = client_credentials), follow these steps:

1. [Configure the REST API Application Registration in Okta.](#)
2. (Optional) [Add Authorization Servers and Scopes in Okta.](#)
3. [Set Up the Web Server Configuration in OneStream.](#)
4. [Configure the User in OneStream.](#)

To enable single sign-on with Okta for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, see the *Installation and Configuration Guide*.

Configure the REST API Application Registration in Okta

To configure the REST API application registration, you need to copy the client ID from Okta and paste it into the Web Server Configuration in OneStream.

1. Log into your Okta account.
2. In the **Applications** list on the left, select **Applications**.

3. Click **Create App Integration**.
4. In the **Create a new app integration** dialog box, for **Sign-in method**, select **API Services**.
5. Click the **Next** button.
6. On the **New API Services App Integration** page, in the **App integration name** field, enter the name of the Okta API application.
7. Click the **Save** button.
8. Copy the client ID. You will need to paste this into the Web Server Configuration in OneStream.

Add Authorization Servers and Scopes in Okta

To configure authorization servers, copy the authorization server ID from the issuer URI and the custom scopes from Okta and paste them into the Web Server Configuration in OneStream.

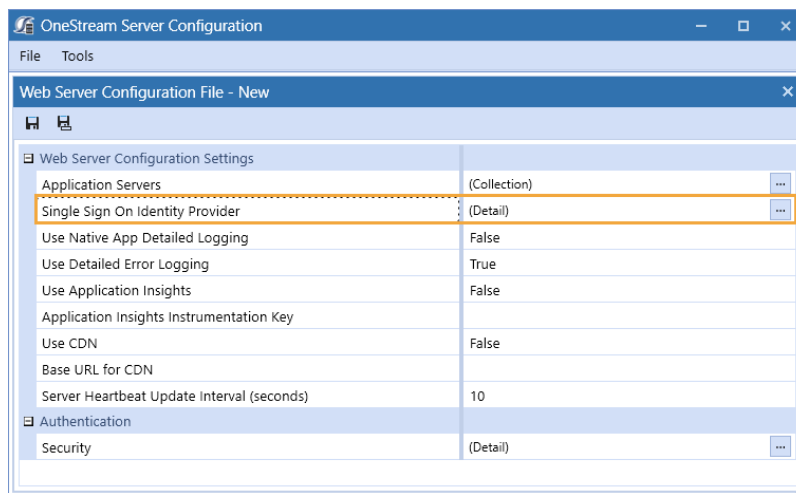
1. Log into your Okta account.
2. In the **Security** list on the left, select **API**.
3. Click the **Add Authorization Server** button.
4. Enter a name and, in the **Audience** field, enter the client ID from the Okta application. See [Configure the REST API Application Registration in Okta](#) step 8.
5. Click the **Save** button. The **API** page displays the list of authorization servers and the corresponding issuer URIs. You will need to paste the authorization server ID from the issuer URI into the Web Server Configuration in OneStream.
6. To add a custom scope to support the Machine-to-Machine scenario, on the **API** page, select the authorization server.
7. Select the **Scopes** tab.
8. Click the **Add Scope** button.
9. Enter the information and click the **Create** button. You will need to paste these custom scopes into the Web Server Configuration in OneStream.

Set Up the Web Server Configuration in OneStream

1. Open the **OneStream Server Configuration Utility** application.
2. Go to **File > New Web Server Configuration File**.

NOTE: Alternatively, you can open an existing file to edit it.

3. In the **Web Server Configuration Settings** section, click the ellipsis to the right of **Single Sign On Identity Provider**.



4. Click the ellipsis to the right of **Okta Identity Provider**.

Configure OneStream API for External Authentication

Single Sign On Identity Provider	
General	
SSO Identity Provider Type	Okta
Browser UX Settings	(Detail)
OIDC Compliant Provider Settings	
User Name Lookup	preferred_username_ois, preferred_username, email, name, sub
Validate Audience	True
Validate Endpoints	True
Validate Issuer Name	True
Identity Provider Specific Settings	
OneStream Identity Server	(Detail)
Azure Identity Provider	(Detail)
Okta Identity Provider	(Detail)
PingFederate Identity Provider	(Detail)
SAML 2.0 Identity Provider	(Detail)
OK Cancel	

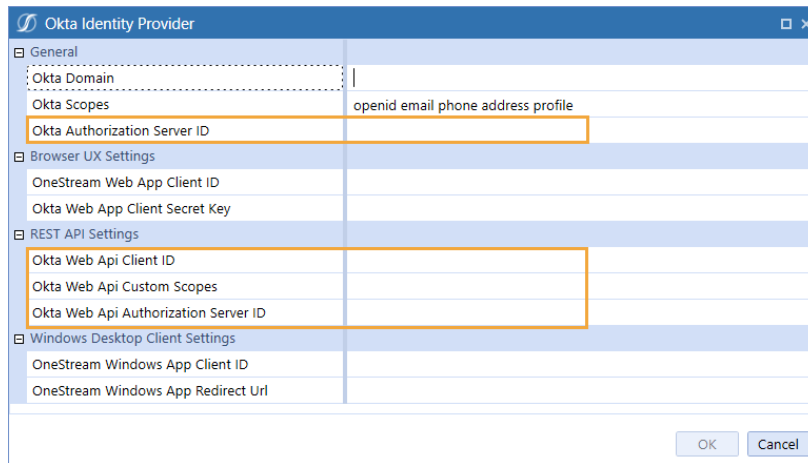
5. In the **Okta Identity Provider** dialog box, in the **General** and **REST API Settings** sections, complete the following fields:

- **Okta Authorization Server ID:** Enter the authorization server ID from the issuer URI in Okta. See [Add Authorization Servers and Scopes in Okta](#) step 5. Alternatively, use the default value by either typing **default** or leaving as default (null).

TIP: To view the list of authorization servers and the corresponding issuer URIs in Okta, in the **Security** list on the left, select **API**.

- **Okta Web Api Client ID:** Enter the client ID from the Okta application. See [Configure the REST API Application Registration in Okta](#) step 8.
- **Okta Web Api Custom Scopes:** Enter custom scopes, or leave as default (null). See [Add Authorization Servers and Scopes in Okta](#) step 9.
- **Okta Web Api Authorization Server ID:** Enter the server ID if using a custom authentication server, or leave as default (null).

Configure OneStream API for External Authentication



6. Click the **OK** button.
7. Save changes and reset IIS.

NOTE: Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Configure the User in OneStream

1. In the OneStream Desktop application, go to **System > Security > Users > <user>**.
2. In the **Authentication** properties, complete the following fields for REST API authentication through Okta.
 - **External Authentication Provider:** In the drop-down menu, select the Okta configuration.
 - **External Provider User Name:** Enter the client ID from Okta. See [Configure the REST API Application Registration in Okta](#) step 8.
 - **Internal Provider Password:** Enter a password.
3. Click the **Save** icon.

PingFederate Configuration

To configure OneStream REST API to support PingFederate authentication, follow these steps:

1. [Configure the REST API Application Registration in PingFederate.](#)
2. [Set Up the Web Server Configuration in OneStream.](#)
3. [Configure the User in OneStream.](#)

To enable single sign-on with PingFederate for the OneStream Desktop application, which includes the Windows Client application and the Excel Add-In, using OIDC protocol, see the *Installation and Configuration Guide*.

Configure the REST API Application Registration in PingFederate

To configure the REST API application registration, you need to enter the same client ID in PingFederate and the Web Server Configuration in OneStream. You also need to copy the client secret from PingFederate and paste it into the Web Server Configuration in OneStream.

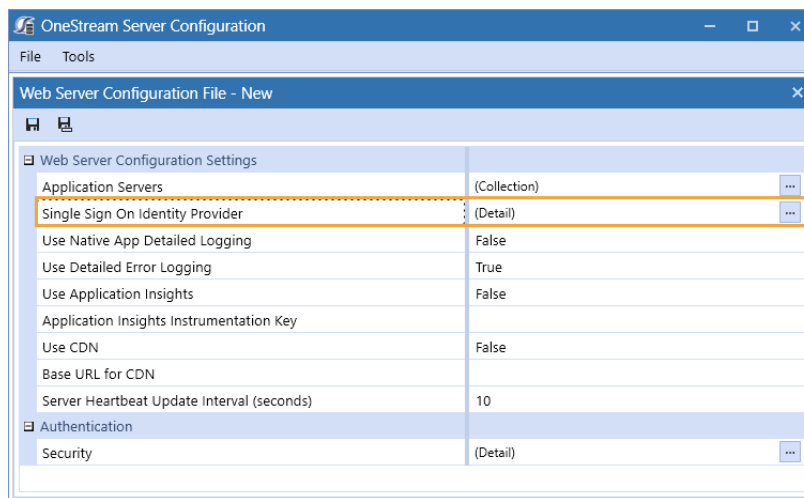
1. Log into your PingFederate account.
2. In the menu on the left, click **OAuth Server**.
3. Under the **CLIENTS** list, click the **Create New** button.
4. On the **Client** page, complete the following fields:
 - **CLIENT ID**: Enter a client ID, which is a unique name or identifier for the application registration.
 - **NAME**: Enter the name of the client.
 - **CLIENT AUTHENTICATION**: Select **CLIENT SECRET**.
 - **CLIENT SECRET**: Select **CHANGE SECRET** and then click the **Generate Secret** button.
 - **ALLOWED GRANT TYPES**: Select **Client Credentials**.
 - **REQUIRE PROOF KEY FOR CODE EXCHANGE (PKCE)**: Select this option.
5. Click the **Save** button.

Set Up the Web Server Configuration in OneStream

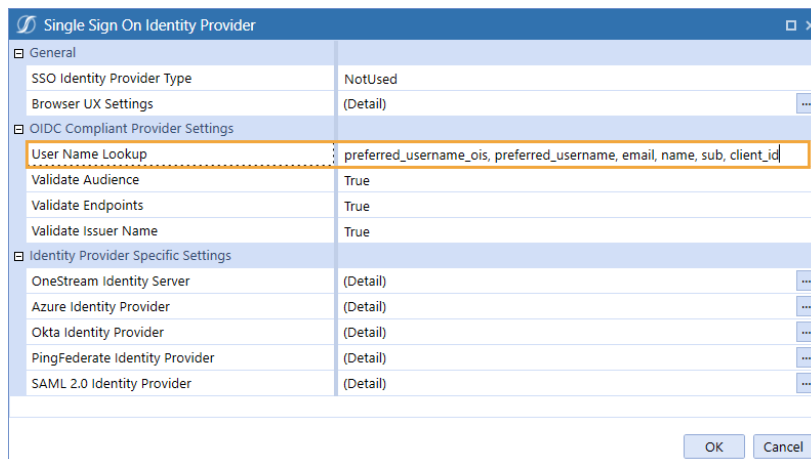
1. Open the **OneStream Server Configuration Utility** application.
2. Go to **File > New Web Server Configuration File**.

NOTE: Alternatively, you can open an existing file to edit it.

3. Click the ellipsis to the right of **Single Sign On Identity Provider**.



4. In the **User Name Lookup** field, type **client_id** to include this claim in the ordered lookups.



Configure OneStream API for External Authentication

5. Click the ellipsis to the right of **PingFederate Identity Provider**.

The screenshot shows the 'Single Sign On Identity Provider' dialog box. It has a tree view on the left with sections: General, OIDC Compliant Provider Settings, and Identity Provider Specific Settings. The 'Identity Provider Specific Settings' section is expanded, showing a list of providers: OneStream Identity Server, Azure Identity Provider, Okta Identity Provider, PingFederate Identity Provider (highlighted with an orange border), and SAML 2.0 Identity Provider. Each provider has a '(Detail)' button and an ellipsis button to its right. The 'PingFederate Identity Provider' ellipsis button is the one to be clicked.

Section	Item	Value	Action
General	SSO Identity Provider Type	NotUsed	
	Browser UX Settings	(Detail)	...
OIDC Compliant Provider Settings	User Name Lookup	preferred_username_ois, preferred_username, email, name, sub, client_id	
	Validate Audience	True	
	Validate Endpoints	True	
	Validate Issuer Name	True	
Identity Provider Specific Settings	OneStream Identity Server	(Detail)	...
	Azure Identity Provider	(Detail)	...
	Okta Identity Provider	(Detail)	...
	PingFederate Identity Provider	(Detail)	...
	SAML 2.0 Identity Provider	(Detail)	...

OK Cancel

6. In the **PingFederate Identity Provider** dialog box, in the **REST API Settings** section, complete the following fields :
- **OneStream Web Api Client ID**: Enter the client ID you entered in PingFederate. See [Configure the REST API Application Registration in PingFederate](#) step 4.
 - **OneStream Web Api Scopes**: Enter custom scopes.
 - **OneStream Web Api JWKS Path**: Enter the path on the PingFederate server to publish a JSON Web Key Set with the keys and certificates used for signature verification.

Configure OneStream API for External Authentication

The screenshot shows the 'PingFederate Identity Provider' configuration window. The 'REST API Settings' section is highlighted with an orange box, indicating the fields to be configured for OneStream API. The 'General' section shows 'PingFederate Domain' and 'PingFederate Scopes' (openid email phone address profile). The 'Browser UX Settings' section shows 'OneStream Web App Client ID' and 'OneStream Web App Client Secret Key'. The 'Windows Desktop Client Settings' section shows 'OneStream Windows App Client ID' and 'OneStream Windows App Redirect Url'. 'OK' and 'Cancel' buttons are at the bottom right.

7. Click the **OK** button.
8. Save changes and reset IIS.

NOTE: Reset IIS after you save any changes to the Application Server Configuration or Web Server Configuration.

Configure the User in OneStream

1. In the OneStream Desktop application, go to **System > Security > Users > <user>**.
2. In the **Authentication** properties, complete the following fields for REST API authentication through PingFederate.
 - **External Authentication Provider:** In the drop-down menu, select the PingFederate configuration.
 - **External Provider User Name:** Enter the client ID you entered in PingFederate. See [Configure the REST API Application Registration in PingFederate](#) step 4.
 - **Internal Provider Password:** Enter a password.
3. Click the **Save** icon.

Configure the AUD Value

In some 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. Open the **OneStream Server Configuration Utility** application.
2. Go to **File > Open Web Server Configuration File**.
3. Find the Web Server Configuration file and click the **Open** button.
4. Click the ellipsis to the right of **Single Sign On Identity Provider**.
5. In **Validate Audience**, select **False** to disable Audience validation.

The screenshot shows the 'Single Sign On Identity Provider' configuration window. It has a blue title bar and a light blue background. The window is divided into sections: 'General', 'OIDC Compliant Provider Settings', and 'Identity Provider Specific Settings'. The 'Validate Audience' setting under 'OIDC Compliant Provider Settings' is highlighted with a dashed orange box and set to 'False'. Other settings include 'SSO Identity Provider Type' (NotUsed), 'Browser UX Settings' ((Detail)), 'User Name Lookup' (preferred_username_ois, preferred_username, email, name, sub), 'Validate Endpoints' (True), and 'Validate Issuer Name' (True). The 'Identity Provider Specific Settings' section lists 'OneStream Identity Server', 'Azure Identity Provider', 'Okta Identity Provider', 'PingFederate Identity Provider', and 'SAML 2.0 Identity Provider', each with a '(Detail)' link. At the bottom right are 'OK' and 'Cancel' buttons.

Section	Setting	Value
General	SSO Identity Provider Type	NotUsed
	Browser UX Settings	(Detail)
OIDC Compliant Provider Settings	User Name Lookup	preferred_username_ois, preferred_username, email, name, sub
	Validate Audience	False
	Validate Endpoints	True
	Validate Issuer Name	True
Identity Provider Specific Settings	OneStream Identity Server	(Detail)
	Azure Identity Provider	(Detail)
	Okta Identity Provider	(Detail)
	PingFederate Identity Provider	(Detail)
	SAML 2.0 Identity Provider	(Detail)