



Explore API Documentation

API Connection Information Version 1.5

Last Modified: 1/31/2024

Last Modified By: Natalie Takacs

Change Log

This changelog only contains significant or notable changes to the revision. Any editorial type changes or minor changes are not included.

Revision	Change Date	Description	Modified By
1.0	07/10/2019	Initial draft	Catie Williams
1.1	10/15/2019	Added SSIS Connection Information	Catie Williams
1.2	03/19/2020	Updated API links to align with our current version strategy	Natalie Takacs
1.3	12/8/2020	Added OData/APIM filtering instructions to API Management section	Natalie Takacs
1.4	11/16/2022	Added verbiage about Explore vs SelfService and integration vs reporting APIs in Overview section	Natalie Takacs
1.5	7/17/2023	Added note about APIM test vs prod links, Power BI server supported refresh capabilities, permissions for access, and removed outdated information	Natalie Takacs
1.6	1/31/2024	Updated APIM connection information	Natalie Takacs

Contents

Change Log	2
Overview.....	3
In-App.....	3
PowerBI	4
Excel	9
OData link filtering	15
Postman.....	15
Get	16
Token Name	16
Grant Type.....	16
Access Token URL:	16
Client ID.....	17
Client Secret	17
Client Authentication.....	17
Request Token.....	17

Overview

This document provides instructions to connect to the reporting APIs that are available via OData.

You will notice that when connecting to the reporting APIs there are two different “schemas”

SelfService_: The APIs prefixed with SelfService are designed to satisfy extraction requirements. These APIs more closely match the source data, though we do some denormalization to combine IDs and text into the same table; and ensure field/table naming more closely matches the user interface. Records that have been deleted or updated in the products’ user interface are soft-deleted in the database. These APIs will expose both “active” and “Inactive” (soft-deleted) records. An example of this use case would be writing ETL against the data to extract it into a data warehouse that is used to combine data for reporting. However, it is important to note that these APIs can also be used for BI use cases, such as connecting to Power BI.

Explore_: The APIs prefixed with Explore are designed to satisfy more of a BI use case. These APIs report only active and latest records as reflected in the system. And in some cases, they may have more business logic implemented in the code behind them. They also will have naming conventions similar to the user interface in the applications. An example of this use case would be connecting in Power BI.

Please note that the reporting APIs were NOT designed for integrating between InEight and other interfaces. The reporting APIs read data from the InEight consolidated reporting database which aggregates data from the different products’ source databases and therefore is not the source of truth. Additionally, in the acquisition of this data, there may be a refresh interval that occurs in the background to pull data from source to reporting database. This means there could be a lag in the data as you pass it from the reporting API to interface, resulting in the possibility that your extract may not match what the product interface reads.

For these reasons, it is not supported, nor is it a best practice to utilize the reporting APIs for integration use cases.

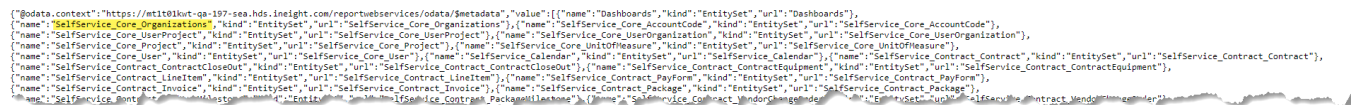
In-App

To preview the available APIs within ProjectSuite, a user must be logged into the application.

After logging into the application, adding the following text to the base URL, will provide a preview of the APIs available: reportwebservices/odata

Ex: <https://TENANT.hds.ineight.com/reportwebservices/odata>

This is a screenshot of what would be observed, with the yellow highlighted information representing the name of the API.



```

{"@odata.context": "https://tnt181bit-qa-187-sea.hds.ineight.com/reportwebservices/odata/$metadata", "value": [{"name": "Dashboards", "kind": "EntitySet", "url": "Dashboards"}, {"name": "SelfService_Core_Organizations", "kind": "EntitySet", "url": "SelfService_Core_Organizations"}, {"name": "SelfService_Core_AccountCode", "kind": "EntitySet", "url": "SelfService_Core_AccountCode"}, {"name": "SelfService_Core_UserProject", "kind": "EntitySet", "url": "SelfService_Core_UserProject"}, {"name": "SelfService_Core_UserOrganization", "kind": "EntitySet", "url": "SelfService_Core_UserOrganization"}, {"name": "SelfService_Core_Project", "kind": "EntitySet", "url": "SelfService_Core_Project"}, {"name": "SelfService_Core_UnitOfMeasure", "kind": "EntitySet", "url": "SelfService_Core_UnitOfMeasure"}, {"name": "SelfService_Core_User", "kind": "EntitySet", "url": "SelfService_Core_User"}, {"name": "SelfService_Calendar", "kind": "EntitySet", "url": "SelfService_Calendar"}, {"name": "SelfService_Contract", "kind": "EntitySet", "url": "SelfService_Contract"}, {"name": "SelfService_Contract_ClosedOut", "kind": "EntitySet", "url": "SelfService_Contract_ClosedOut"}, {"name": "SelfService_Contract_ContractEquipment", "kind": "EntitySet", "url": "SelfService_Contract_ContractEquipment"}, {"name": "SelfService_Contract_LineItem", "kind": "EntitySet", "url": "SelfService_Contract_LineItem"}, {"name": "SelfService_Contract_PayForm", "kind": "EntitySet", "url": "SelfService_Contract_PayForm"}, {"name": "SelfService_Contract_Invoice", "kind": "EntitySet", "url": "SelfService_Contract_Invoice"}, {"name": "SelfService_Contract_Package", "kind": "EntitySet", "url": "SelfService_Contract_Package"}, {"name": "SelfService_Contract_PackageItem", "kind": "EntitySet", "url": "SelfService_Contract_PackageItem"}, {"name": "SelfService_Contract_VendorChangeOrder", "kind": "EntitySet", "url": "SelfService_Contract_VendorChangeOrder"}, {"name": "SelfService_Contract_VendorChangeOrderItem", "kind": "EntitySet", "url": "SelfService_Contract_VendorChangeOrderItem"}]}

```

After the name of the API has been determined, the name can be added to the URL to preview the data: reportwebservices/odata/SelfService_Core_Organizations

Ex: https://TENANT.hds.ineight.com/reportwebservices/odata/SelfService_Core_Organizations

Sample:

```

{"Organization_id":1166,"Organization_name":"New_27_child12","Parent_id":1163,"Organization_description":"New_27_child11ee1","Display_id":"New_27_child1","Organization_path":"S1000
New_27_child12","Organization_hierarchy":"":1:1163:1166:", "Is_active":true}, {"Organization_id":1167,"Organization_name":"June27_1","Parent_id":1,"Organization_description":"June27_1
June27_1","Organization_hierarchy":"":1:1167:", "Is_active":true}, {"Organization_id":1168,"Organization_name":"June_SSID","Parent_id":1,"Organization_description":"June_Name", "Displa
June_SSID","Organization_hierarchy":"":1:1168:", "Is_active":true}, {"Organization_id":1169,"Organization_name":"June_Display1","Parent_id":1,"Organization_description":"June_Name", "D
June_Display1","Organization_hierarchy":"":1:1169:", "Is_active":true}, {"Organization_id":1170,"Organization_name":"June_PK1","Parent_id":1,"Organization_description":"June_Name", "Dis
June_PK1","Organization_hierarchy":"":1:1170:", "Is_active":true}, {"Organization_id":1171,"Organization_name":"June_PK1","Parent_id":1,"Organization_description":"June_Name", "Displ
June_PK1","Organization_hierarchy":"":1:1171:", "Is_active":true}, {"Organization_id":1172,"Organization_name":"June_PK11","Parent_id":1,"Organization_description":"June_Name", "Displa
June_PK11","Organization_hierarchy":"":1:1172:", "Is_active":true}, {"Organization_id":1173,"Organization_name":"MAC_parent","Parent_id":1,"Organization_description":"MAC_parent", "Dis
MAC_parent","Organization_hierarchy":"":1:1173:", "Is_active":true}, {"Organization_id":1174,"Organization_name":"MAC_child","Parent_id":1173,"Organization_description":"MAC_child", "D
MAC_child","Organization_hierarchy":"":1:1173:1174:", "Is_active":true}, {"Organization_id":1175,"Organization_name":"new_27_06d","Parent_id":1,"Organization_description":"new_27_06
new_27_06d","Organization_hierarchy":"":1:1175:", "Is_active":true}, {"Organization_id":1176,"Organization_name":"Userstory_19_7","Parent_id":1,"Organization_description":"Userstory_1
Userstory_19_7","Organization_hierarchy":"":1:1176:", "Is_active":true}, {"Organization_id":1177,"Organization_name":"Userstory_19_7_1","Parent_id":1176,"Organization_description":"Userstory_19_7", "Display_id":"Userstory_19_7_1", "Organization_path":null
{"Organization_id":1178,"Organization_name":"Userstory!@#%&*+", "Parent_id":1176,"Organization_description":"Userstory!@#%&*+", "Display_id":"Userstory!@#%&*+", "Organization_pa
Userstory!@#%&*+", "Organization_hierarchy":"":1:1176:1178:", "Is_active":true}, {"Organization_id":1179,"Organization_name":"J27Proj", "Parent_id":1,"Organization_description":"J27Pr
J27Proj","Organization_hierarchy":"":1:1179:", "Is_active":true}, {"Organization_id":1180,"Organization_name":"Org_Name_Updating", "Parent_id":1179,"Organization_description":"Org_Desc

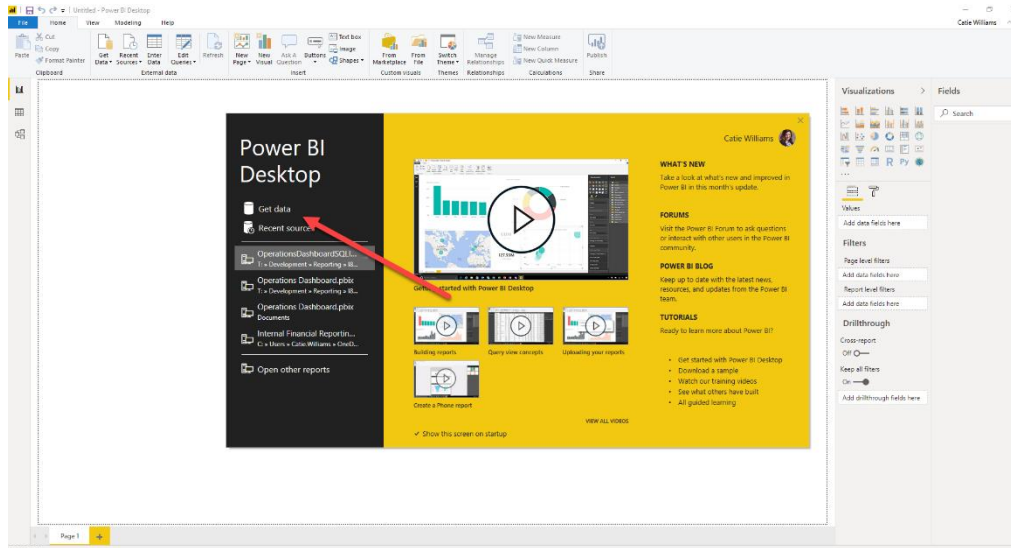
```

Querying the API in the browser is a way to determine if the API is available – any 400 errors would indicate there is an issue within API Management. However, if the API returns results here, but not in a separate tool, there may be an issue with the connection method.

PowerBI

PowerBI provides an out of the box connector that can be used to connect to the Reporting APIs.

After opening PowerBI Desktop, select “Get Data”



The connector to use is **OData feed**

At this time, the URL needs to be entered for the environment manually.

If the environment is Production, the prefix URL is:

<https://api.ineight.com/reporting/v1/>

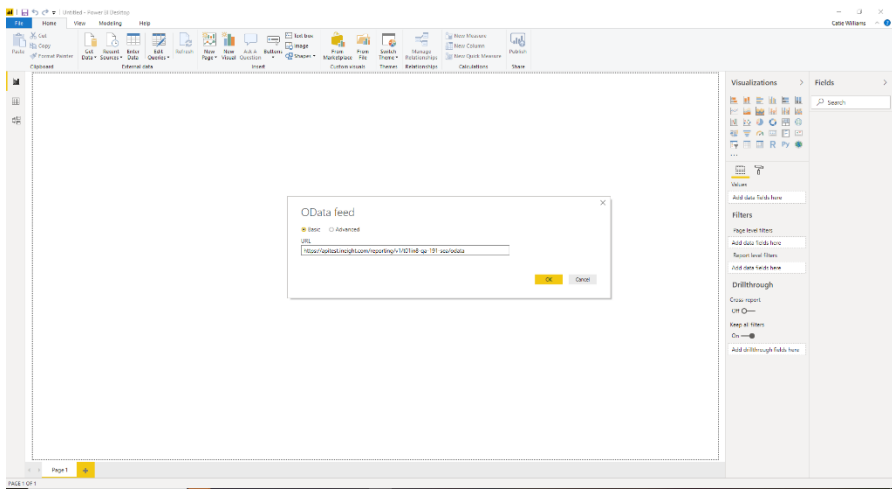
After the v1 – the URL needs to have the environment specified, followed by /odata

Example:

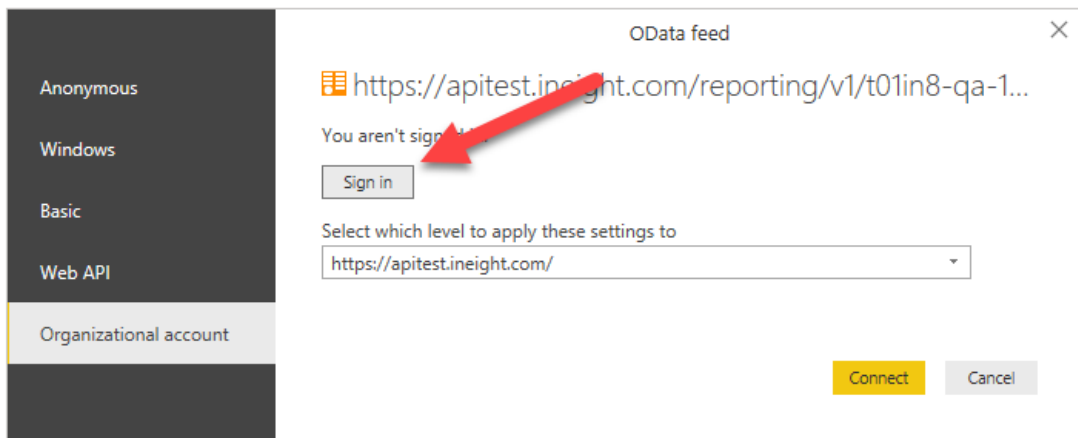
<https://api.ineight.com/reporting/v1/customerenvironmentname/odata>

After the URL has been entered, select “Ok”

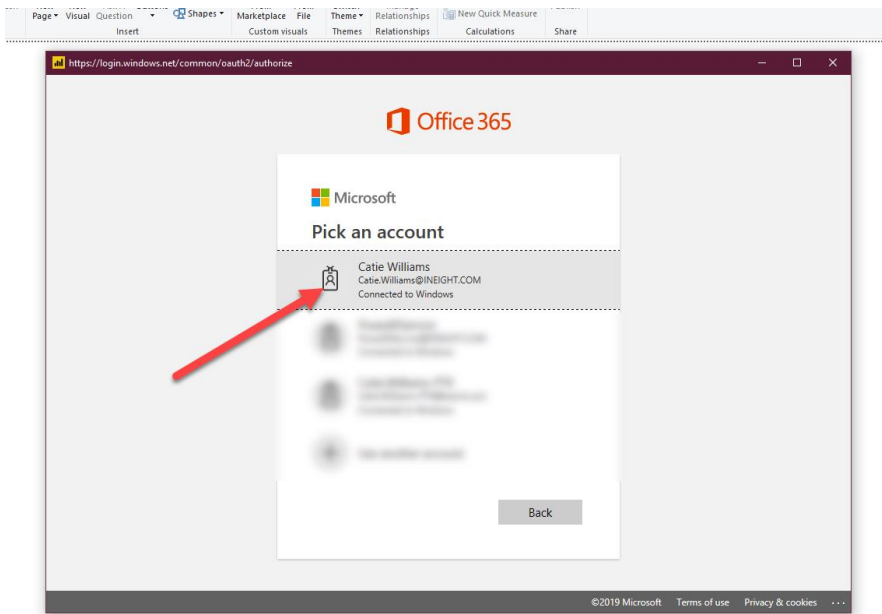
NOTE: “API.ineight.com” should be used with Production AND test/lower environments. The screenshots below are using APItest.ineight.com, this is not available for production or TST environments.



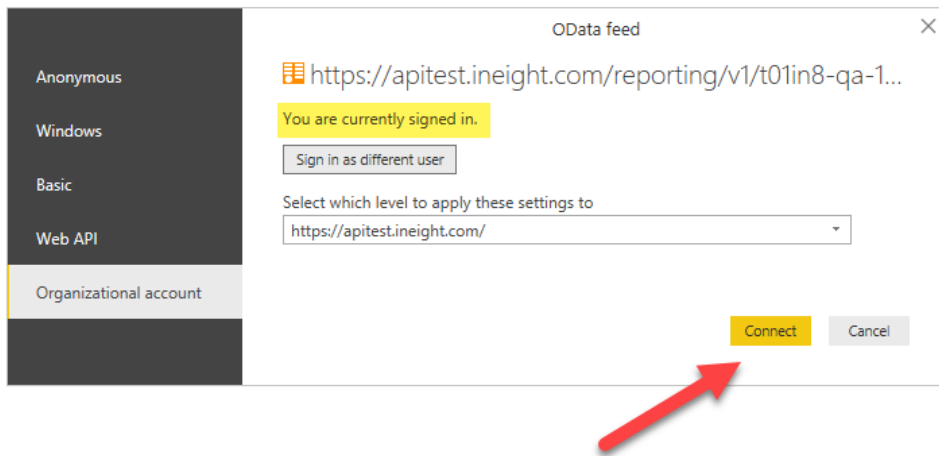
A dialog box will appear prompting for permissions – the user must select Organization Account; accessing these APIs requires permissions in the application. Select Organization Account > Sign In



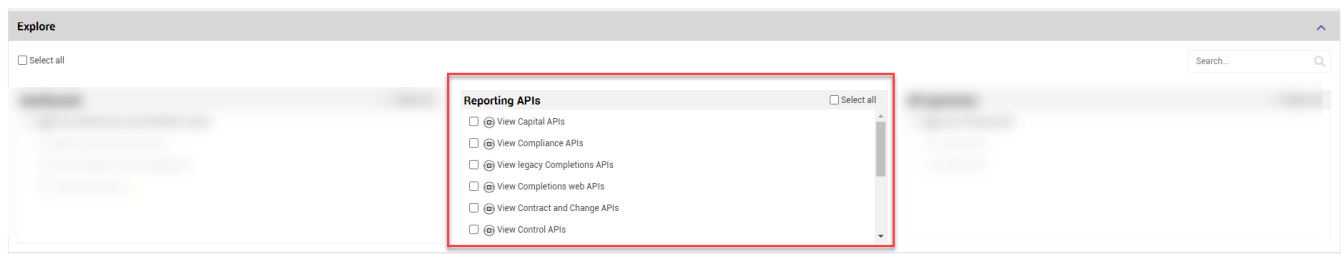
A dialog box appears for the user to select their account



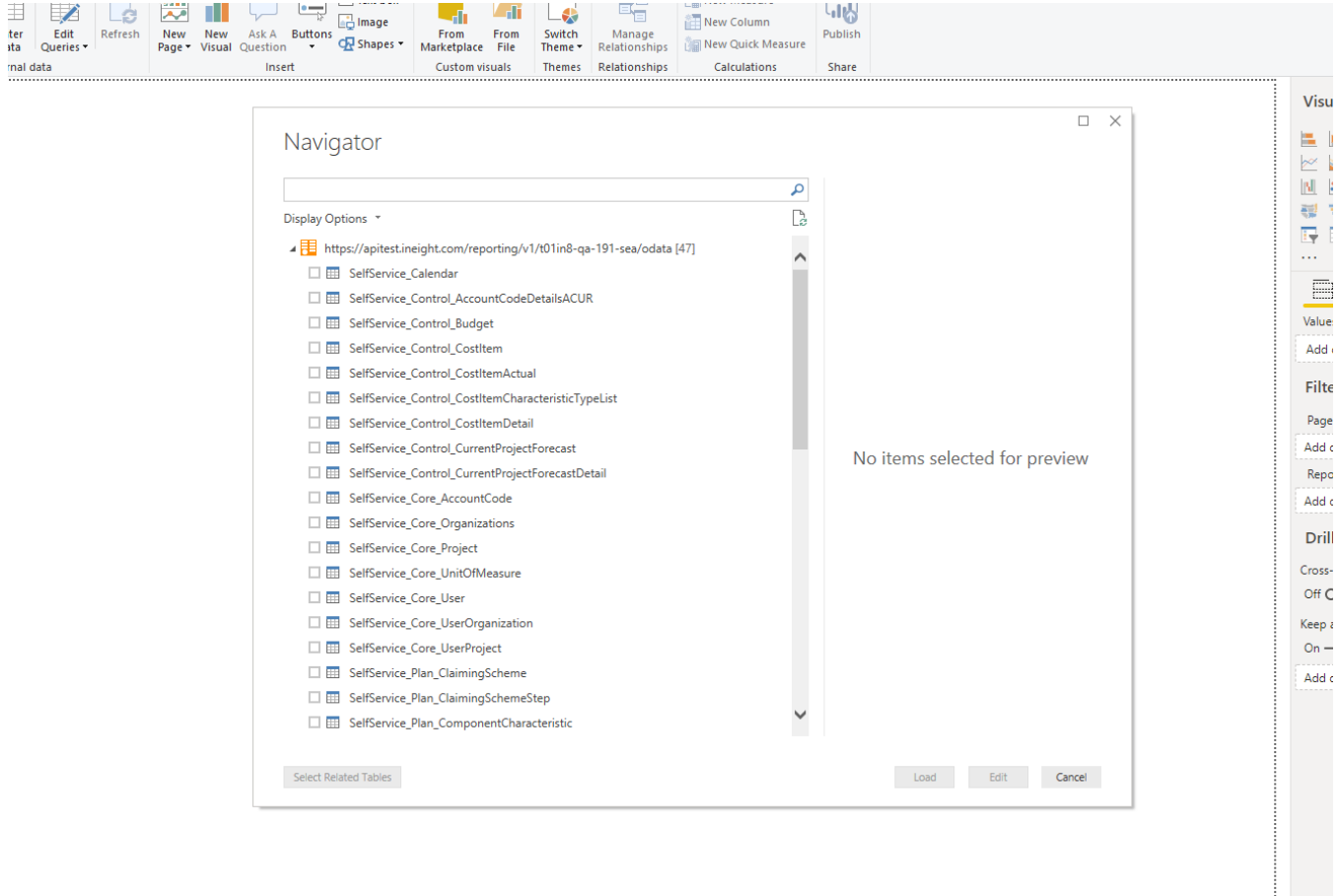
A message should indicate that the user is signed in, select Connect



NOTE: To access these APIs, you are required to have permissions assigned to your role in InEight Core Platform under Suite administration > Roles and Permissions > Explore >



A list of all the available APIs will appear, select either one or multiple APIs to load the data into your PowerBI file. After selecting, click load. It will take a little bit for all the data to load into the file, depending on the data volume in Project Suite.



Please note that we have one instance that spans multiple releases. If you are on an earlier version of Project Suite, you will see APIs that are offered with more recent releases as well. You will not have access to these APIs as your application infrastructure is not updated to support the newer APIs. Newer version's APIs will appear as they do in the black box in the screen shot below. The same will apply with connecting to Excel.

Navigator

Display Options ▾

- SelfService_Plan_DailyPlanIssueAttachment
- SelfService_Plan_DailyPlanIssueTag
- SelfService_Plan_DailyPlanNote
- SelfService_Plan_DailyPlanNoteAssociation
- SelfService_Plan_DailyPlanNoteAttachment
- SelfService_Plan_DailyPlanNotePayrollIndicator
- SelfService_Plan_DailyPlanNoteTag
- SelfService_Plan_DailyPlanQuantity
- SelfService_Plan_DailyPlanSignout
- SelfService_Plan_DailyPlanSummary
- SelfService_Plan_DailyPlanToolBoxTalk
- SelfService_Plan_Discipline
- SelfService_Plan_ProjectComponentPartialView
- SelfService_Plan_QuantityClaim
- SelfService_Plan_WBSClaimingSchemeAssociation
- SelfService_Plan_WorkPackage
- SelfService_Plan_WorkPlanDailyPlan

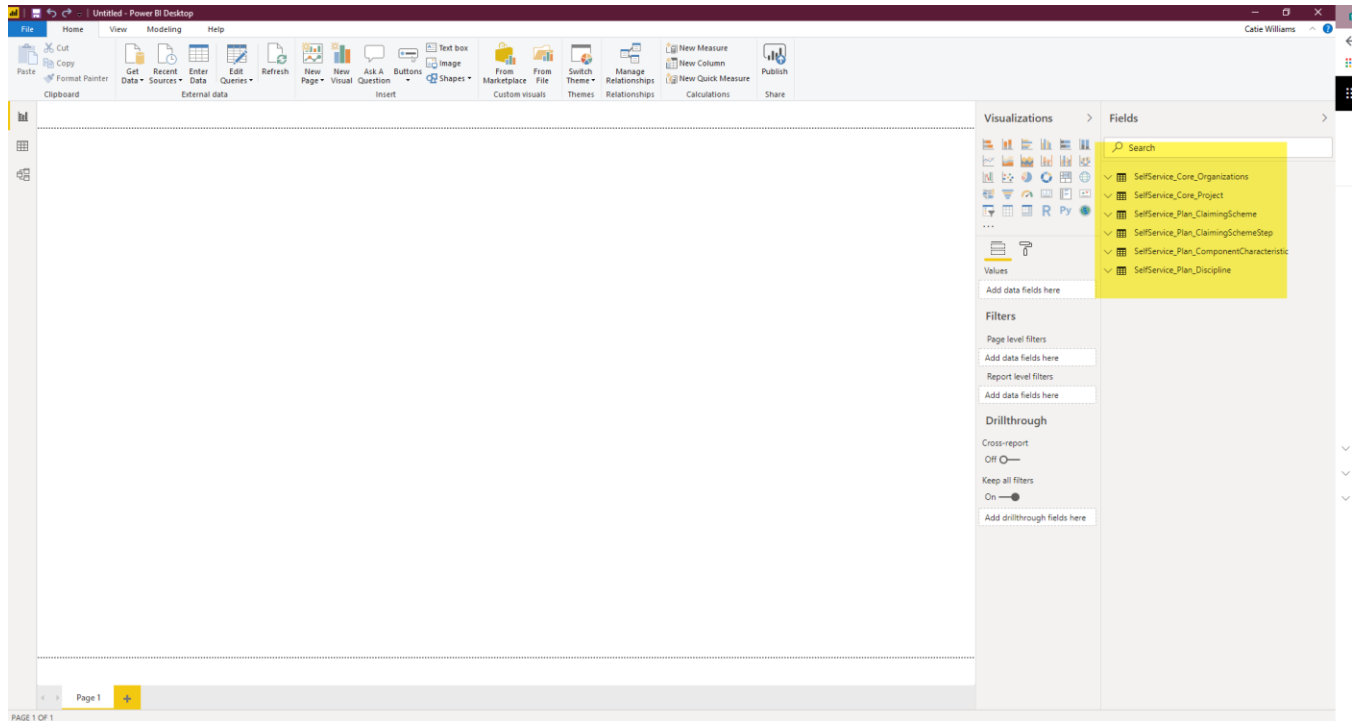
- Cannot get feed for 'SelfService_Core_ProjectEmployee',...
- Cannot get feed for 'SelfService_Core_ProjectEquipment',...
- Cannot get feed for 'SelfService_Plan_WorkPackageCom...

Select Related Tables

No items selected for preview

Load Transform Data Cancel

After loading the data, the APIs should appear on the right-hand side under the “Fields” pane

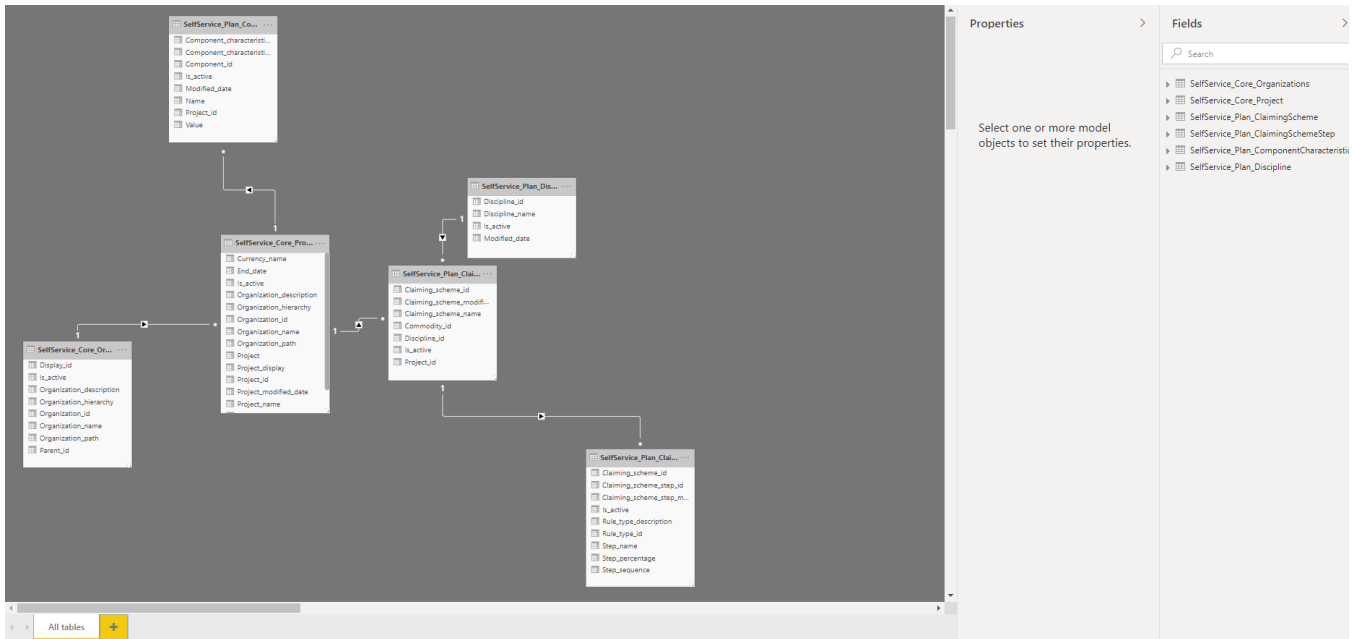


The screenshot shows the Power BI Desktop interface. The 'Fields' pane on the right side is expanded, showing a search bar and a list of data sources. The following APIs are listed under the 'Fields' pane:

- SelfService_Core_Organizations
- SelfService_Core_Project
- SelfService_Plan_ClaimingScheme
- SelfService_Plan_ClaimingSchemeStep
- SelfService_Plan_ComponentCharacteristic
- SelfService_Plan_Discipline

The 'Visualizations' pane is also visible on the left side of the Fields pane, showing various visualization options like Column chart, Line chart, etc.

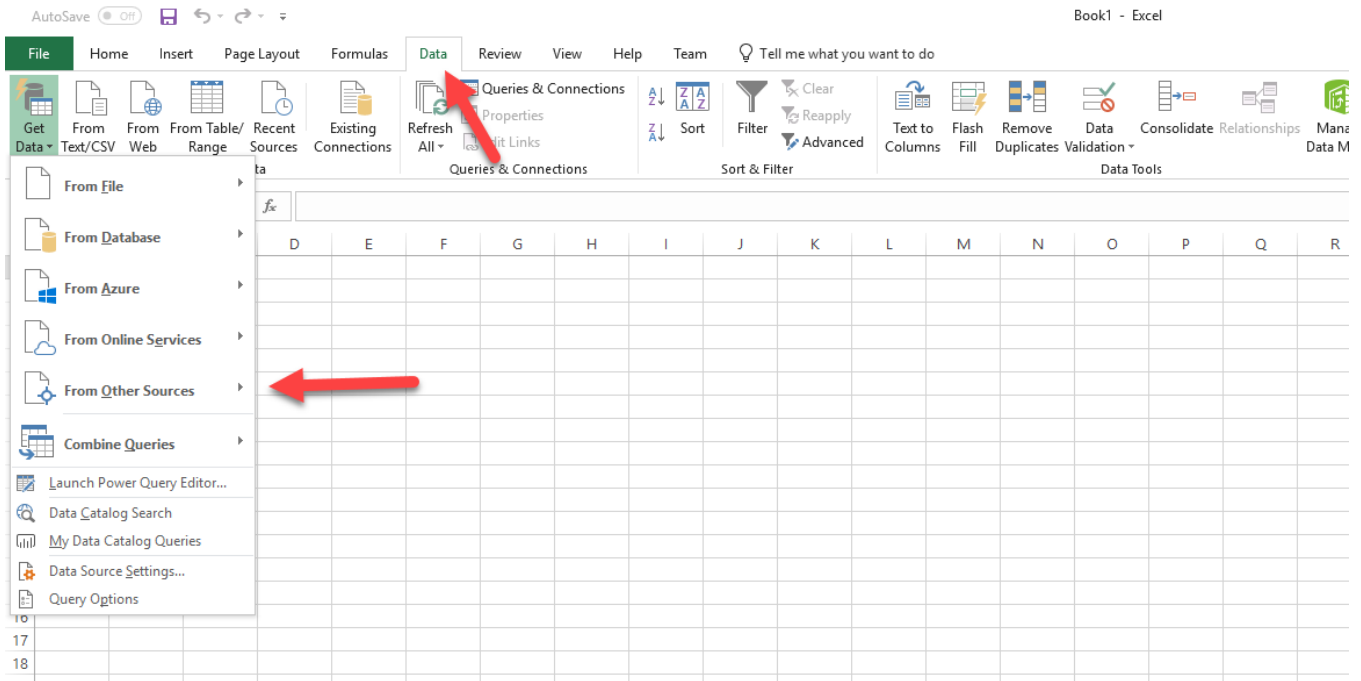
The APIs have also been automatically related to one another, which can be seen under the modeling tab



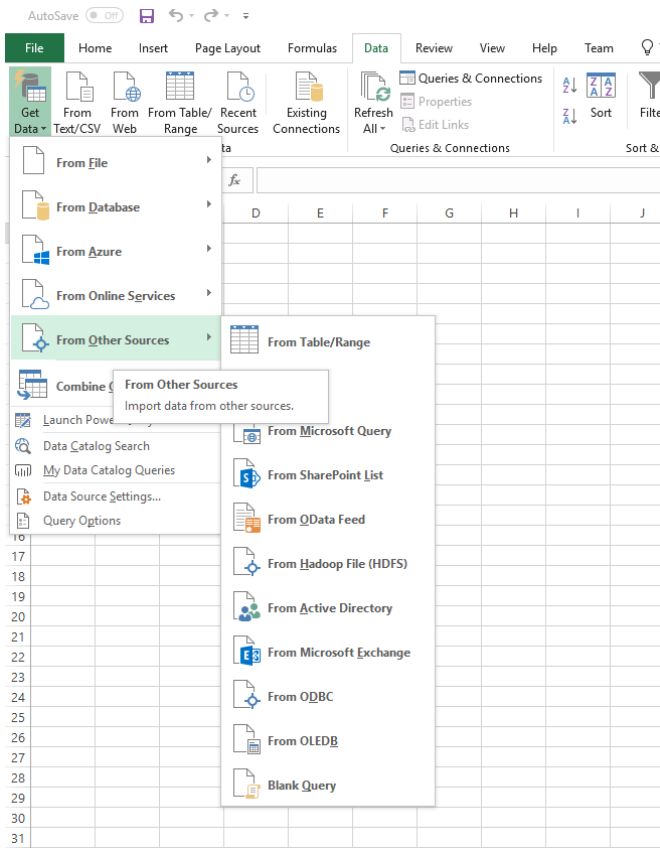
Excel

The reporting APIs can also be accessed via Excel, or any tool that supports OData APIs and Azure Active Directory (AD).

Within Excel, navigate to the “Data” tab, select “Get Data”, select “From Other Sources”



Select “From OData Feed”



Like the experience in PowerBI, you will let enter a modified URL for the environment you want to connect to.

At this time, the URL needs to be entered for the environment manually.

If the environment is a QA or TST the prefix URL is:

<https://apitest.ineight.com/reporting/v1/>

If the environment is Production, the prefix URL is:

<https://api.ineight.com/reporting/v1/>

After the v the release level needs to be specified then, the specific environment, followed by /odata

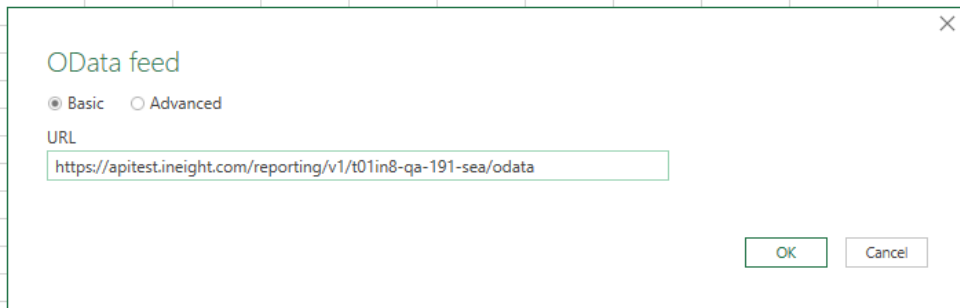
Example:

<https://apitest.ineight.com/reporting/v1/t01in8-qa-191-sea/odata>

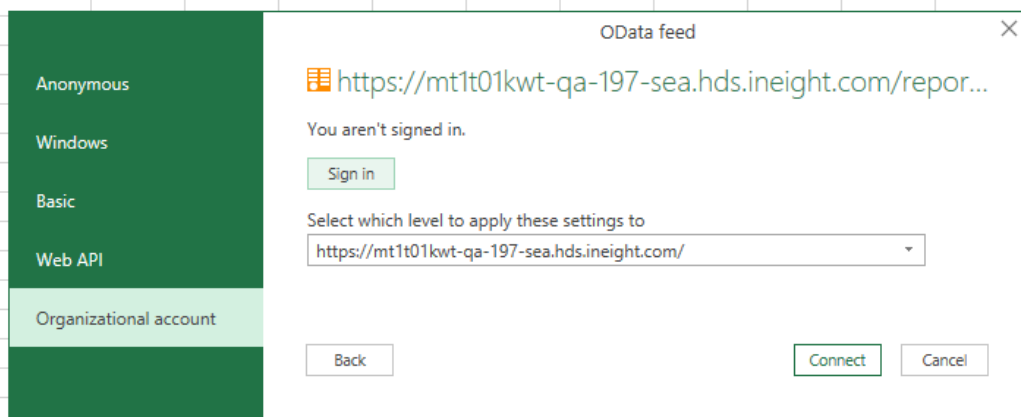
Or

<https://api.ineight.com/reporting/v1/customerenvironmentname/odata>

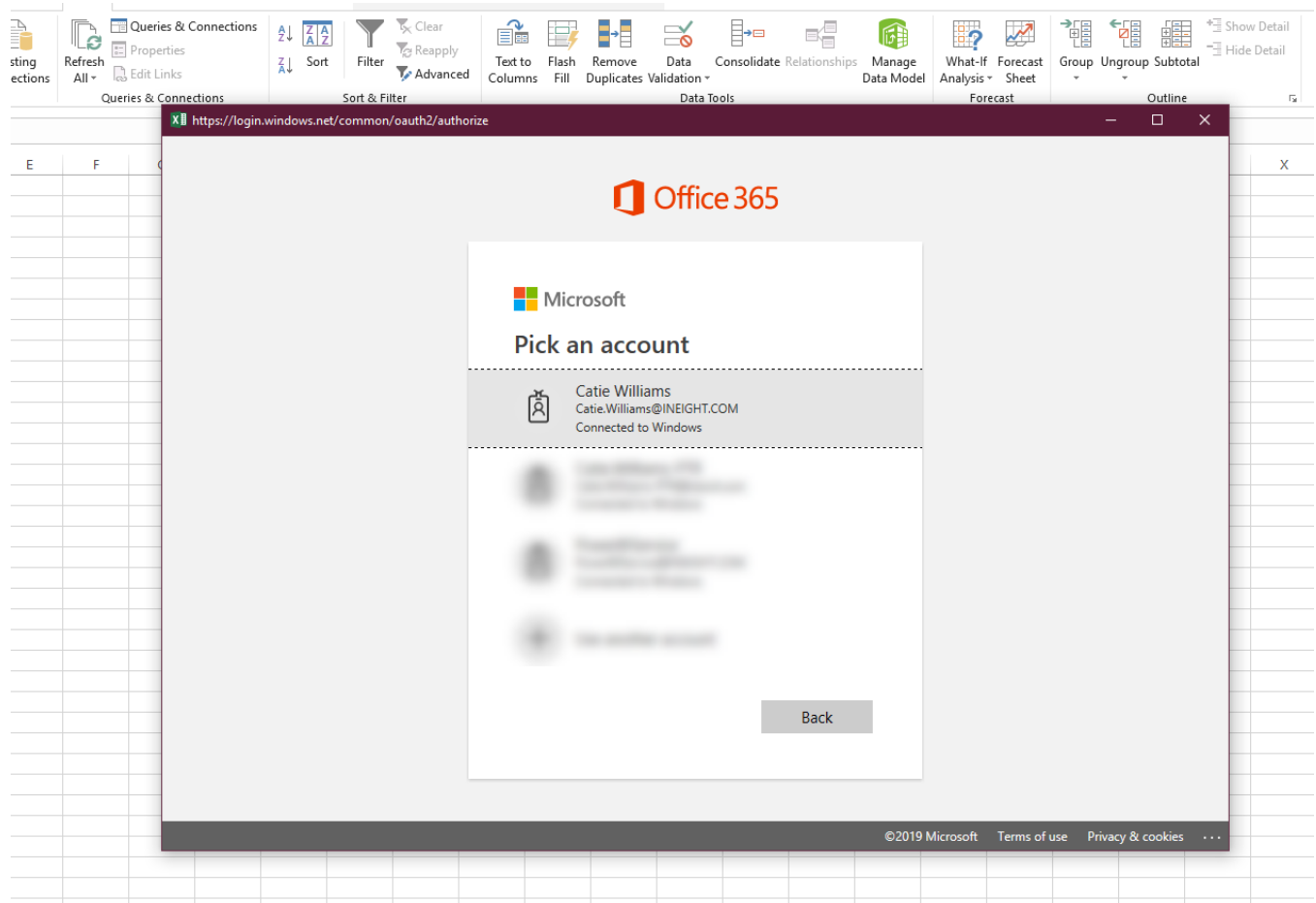
After the URL has been entered, select “Ok”



You will then be prompted to verify your credentials. Make sure to select "Organizational account"

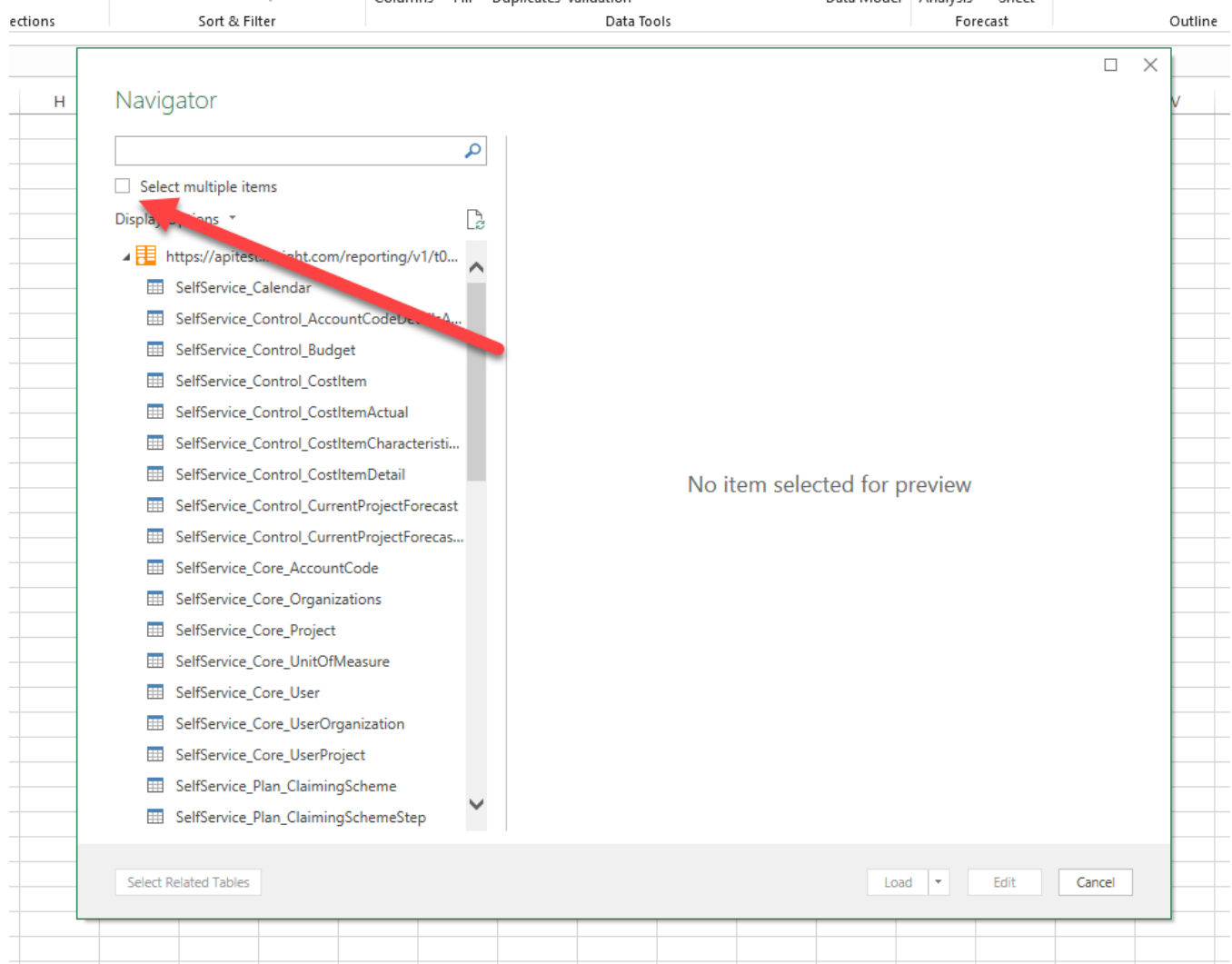


You will sign in with the account used to sign in to Project Suite

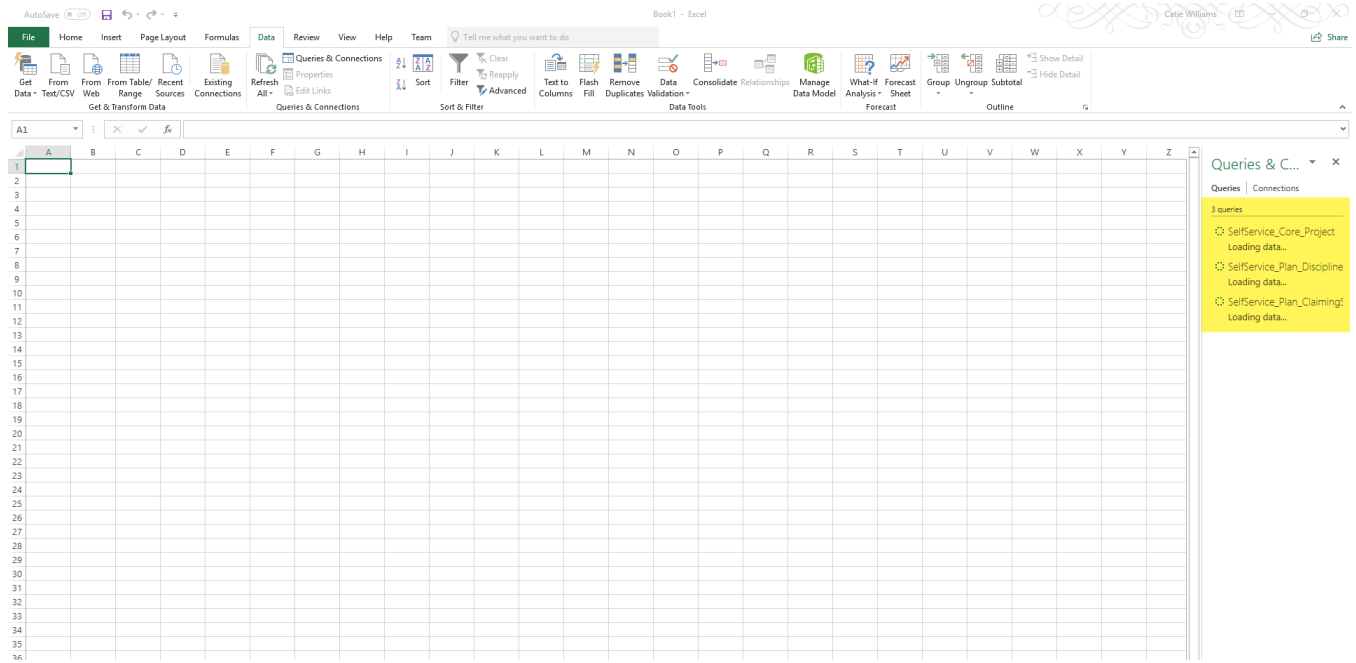


Then you will see a list of APIs that are available for querying.

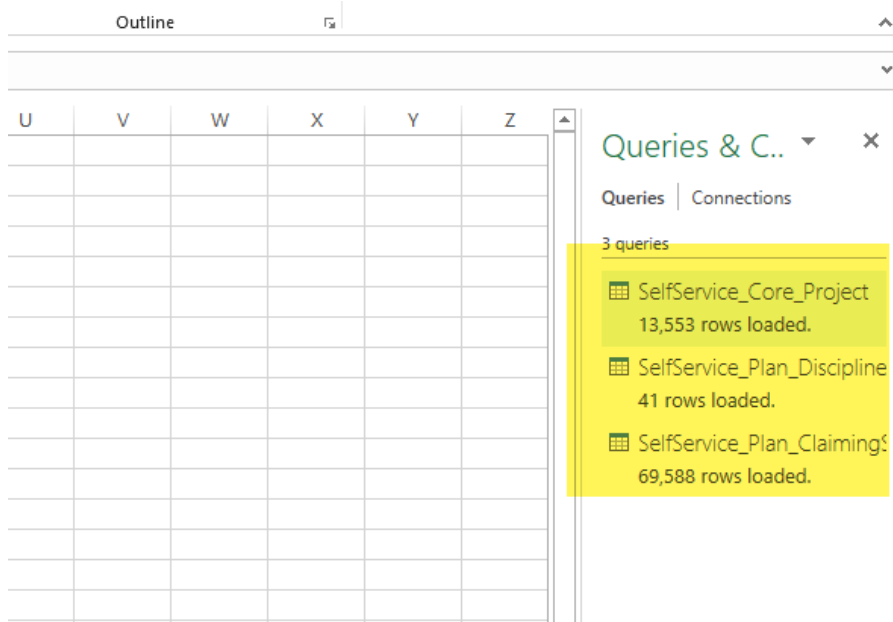
If you want to select Multiple in Excel, you must select "Select Multiple Items"



The tables will load into your excel file



When it is finished loading, it will show the number of records in each table, the data is then ready for your analysis. You can save this file and refresh the data vs reconnecting every time.



If you want to add more data to the Excel sheet, you simply go back to “Data” > “Get Data”

OData link filtering

You can limit the data you bring into your PowerBI or Excel files by applying a filter to your OData link.

For example, if you wanted to filter the results of the SelfService_Core_Project API to just Project Id 1, you would add the following:

Project_Id eq 1

OData link:

[https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?\\$filter=Project_Id eq 1](https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?$filter=Project_Id eq 1)

You will need to know the name of the field you would like to filter and the data type. This will help you format the syntax for the filter. Typically, text fields will require single quoted around the text you are filtering on, for example:

Project_Display eq 'Test Project'

OData link:

[https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?\\$filter=Project_Display eq 'Test Project'](https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?$filter=Project_Display eq 'Test Project')

Whereas, like above, the Project_Id field is a numeric value field, so no quotes are required.

NOTE: The Power BI server does not support scheduled refreshes on OData links that have been filtered. You cannot utilize a filtered OData link to access data in a Power BI file, publish it to the Power BI Server, and set a refresh schedule. Power BI will only allow you to create a refresh schedule on dashboards that are using the “base” link. Ex:

Filtered: [https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?\\$filter=Project_Display eq 'Test Project'](https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?$filter=Project_Display eq 'Test Project')

VS

Not filtered (base): <https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata>

You can also use this methodology to select specific columns from the APIs to load into your PowerBI file rather than pulling all columns into the file. The values you will enter here should be a comma separated list of the fields you wish to pull in, the text from the \$select parameter is included below:

Project_Id, Project_Display

OData link:

[https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?\\$select=Project_Id, Project_Display](https://apitest.ineight.com/reporting/v1/mt1t01kwt-qa-211-sea/odata/SelfService_Core_Project?$select=Project_Id, Project_Display)

NOTE: this is a method to create and test your OData filters or select list, you can also add the filters to your OData link manually if you are familiar with the syntax.

Postman

APIs can also be viewed via Postman

Before this process can be started, you will need to request the following from your InEight account representative:

- Tenant ID
- Resource ID

- ClientID
- ClientSecret

This information should be kept private as it will provide access to data in the application.

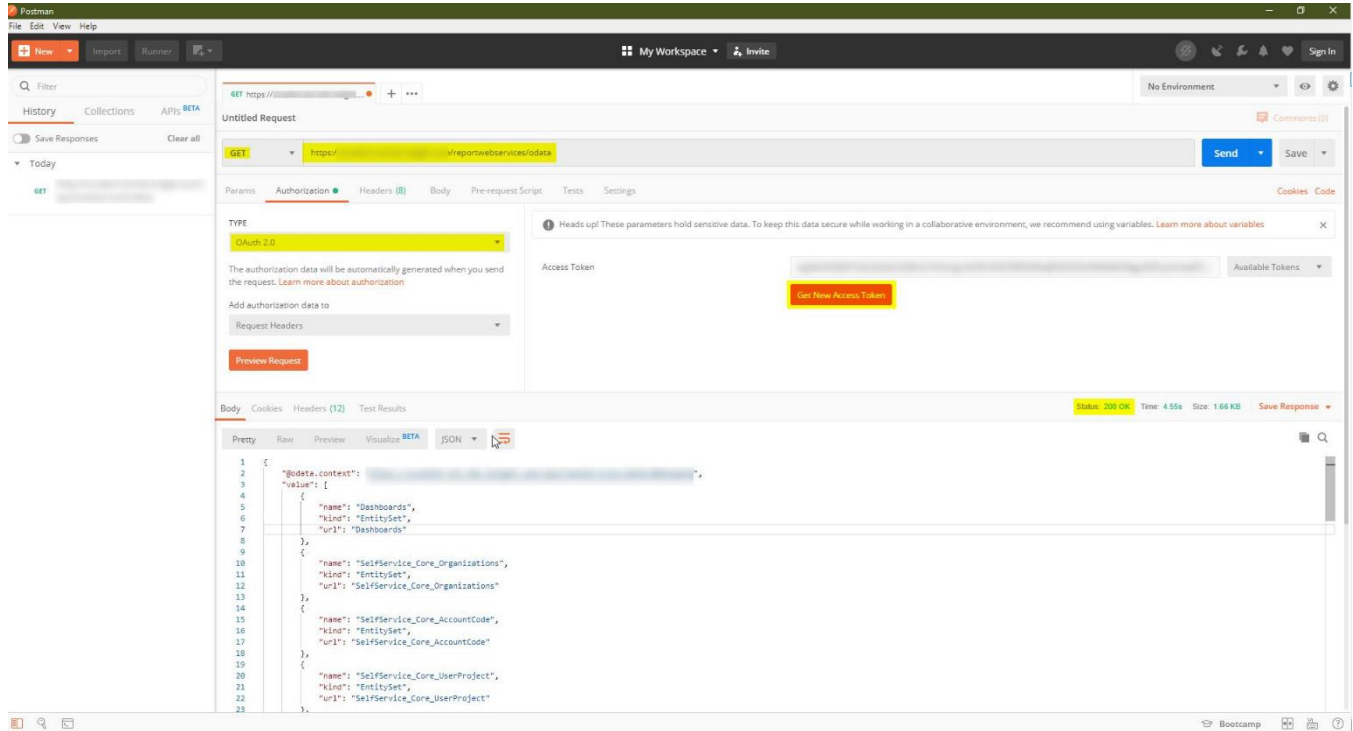
Get

Under Get enter the application URL followed by /reportwebservices/odata

The full URL should look like this:

<https://TenantInfo.hds.ineight.com/reportwebservices/odata>

Select Get New Access Token



Token Name

Select Token Name

Grant Type

Select Client Credentials

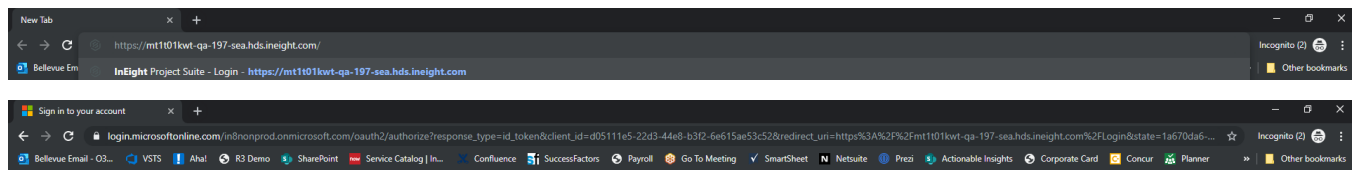
Access Token URL:

This is a URL that will retrieve the token – it will look similar to this:

<https://login.microsoftonline.com/in8nonprod.onmicrosoft.com>

To determine what this URL is, open an incognito browser, enter your application URL and select enter – this will show you what your login URL is

NOTE: When connecting to production or a test environment – your URL will not indicate in8nonprod – it will likely indicate your tenant name. This is just an example.



Copy the URL and add the following to the end of the URL: /oauth/2/token

The URL should look like this:

https://login.microsoftonline.com/in8nonprod.onmicrosoft.com/oauth/2/token

Client ID

Enter the provided client ID

Client Secret

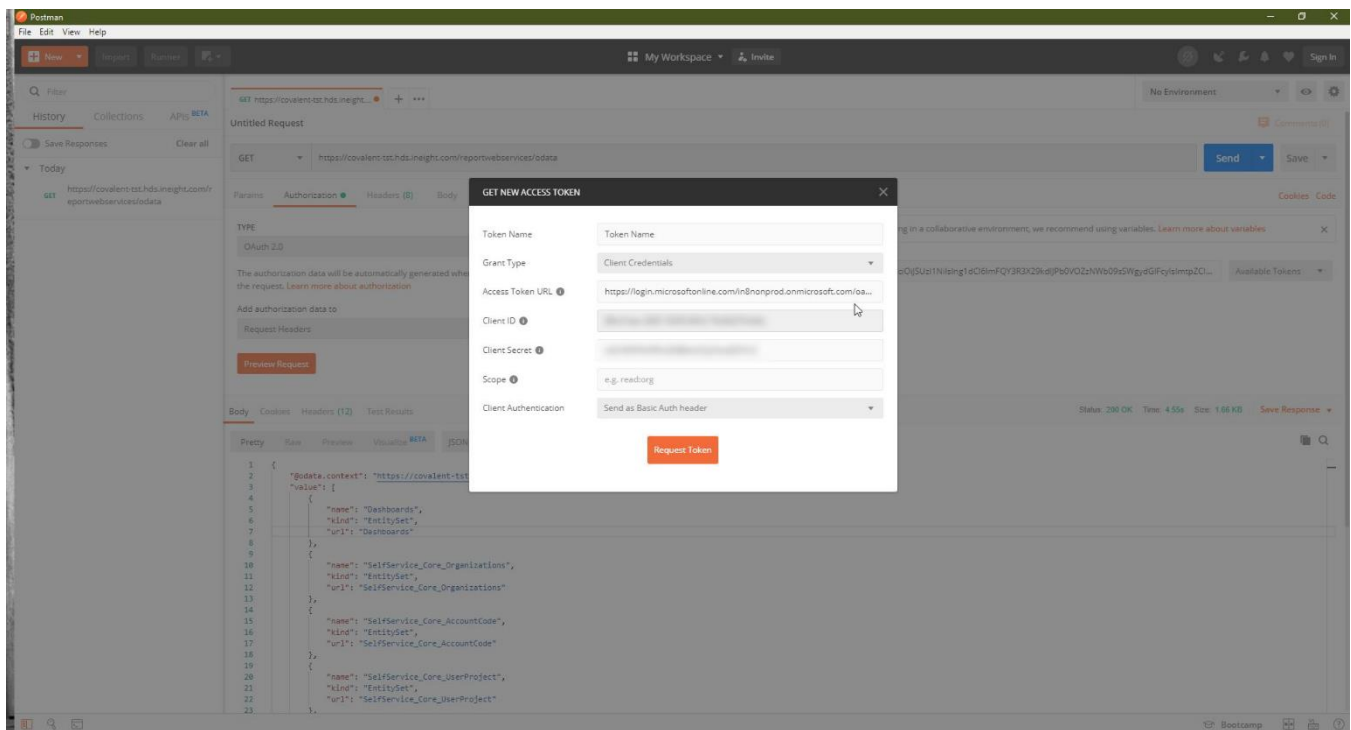
Enter the provided client secret

Client Authentication

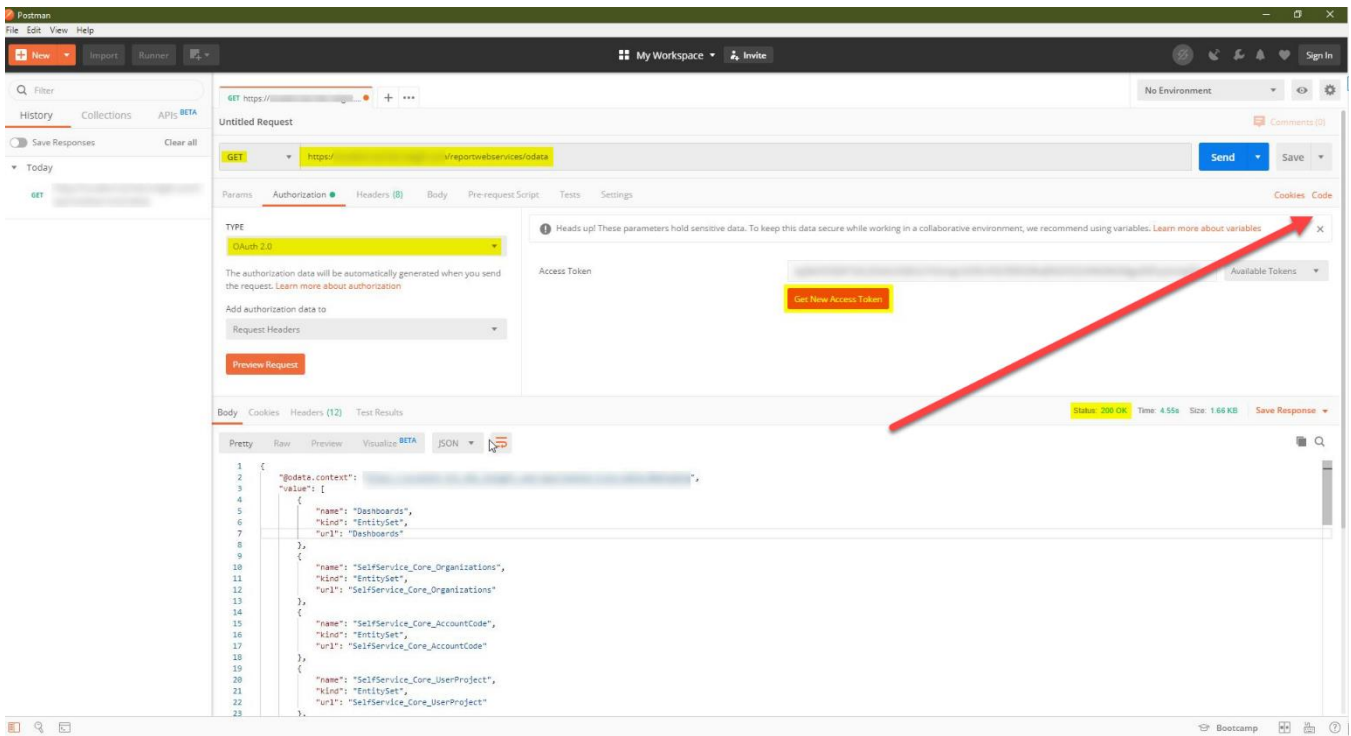
Select Send as Basic Auth Header

Request Token

Select Request Token



You will then be taken back to the main postman screen, select Code



This will show you the code snippet, based on whichever method you want to use (C#, etc)

