



Multiple Application Platform Server

MAPS 6.6 Release Guide

Product version 6.6

Last updated 9/24/2020

Trademark, Publishing Statement, and Copyright Notice

© 1998-2020 Evisions, Inc. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. No part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Evisions, Inc.

The information contained herein is subject to change without notice and is not warranted to be error-free. Product features referenced herein for a period of time may not match product contents. Evisions, Inc. does not warrant that the functions contained in the software will meet your requirements or that the operation of the software will be uninterrupted or error free. Evisions, Inc. reserves the right to make changes and/or improvements in the software without notice at any time.

This software and documentation may provide access to or information on content, products, and services from third parties. Evisions, Inc. and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Evisions, Inc. and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services. Evisions, Inc. does not endorse the content or developer of any products or web sites mentioned.

All information in this guide is designed for instructional purposes only. Evisions, Inc. makes no guarantees regarding the accuracy or performance of any techniques used in this guide. Software configurations and environments may vary, and some techniques used in this guide may not operate efficiently under all configurations. This guide may contain examples of various technologies or products, which are the sole property and responsibility of their creators.

Trademarks are the property of the respective owners for any products mentioned herein.

Table of Contents

What's New in MAPS 6.6?	4
MAPS Updates	4
Installation	4
1. Prepare test environment and ensure current backup	4
2. Check for updates	4
3. Allow update process to complete	5
4. Verify the latest version is installed	5
REST Connector	6
REST Connector Configuration	6
Driver Properties	8
Custom Headers	10
Endpoint Configuration	10
Bearer Token Configuration for a REST Connector	20
Sharing a REST Connector Data Connection	26
Using the REST Connector in Argos	27
Query Construction when Using Mapped Query Parameters	29
XPath Format	29
XML Endpoints	29
JSON Endpoints	31
Example of Creating a REST Connector for Ellucian Ethos	32
MAPS 6.6 Release Notes	40
Getting Help	41

What's New in MAPS 6.6?

Evisions is pleased to announce the release of MAPS version 6.6, which includes the following changes:

- Support for custom API server-side filters (also known as URI Query filters) in the REST Connector configuration. This provides support for endpoint connections such as those in Ellucian Ethos.
- Support for customizable pagination styles. This allows the REST Connector to return all available sets of data using four different styles of pagination.
- Support for inclusion of dynamic values within the URL configured for the REST endpoint.
- Support for Bearer Token Authentication for a REST Connector.
- Ability to export and import REST Connectors.
- Automatic daily refresh of all client licenses.
- Additional enhancements and resolved issues

We appreciate the feedback received from all of our users. Our products would not be what they are today without your continued support. If you have any comments or suggestions, please do not hesitate to [open a case with the HelpDesk](#) and let us know.

MAPS Updates

MAPS 6.6 is an update to the MAPS service (which includes the eLauncher) and the MAPS Config. You should upgrade both components concurrently.

The latest versions of MAPS included in this release are:

- MAPS Service 6.6.0.2386
- MAPS Config 6.6.0.1326

After you install the MAPS components, you will also install the updated REST Connector driver:

- REST Connector XQP 1.0.1.38

Installation

1. Prepare test environment and ensure current backup

We highly recommend installing updates in a test environment before applying them to your production environment. You should make sure that a current backup is available in case of any unforeseen issues. To create a full backup of your MAPS environment, go to the **Server** -> **Backups** screen in MAPS and click **Backup Now**.

2. Check for updates

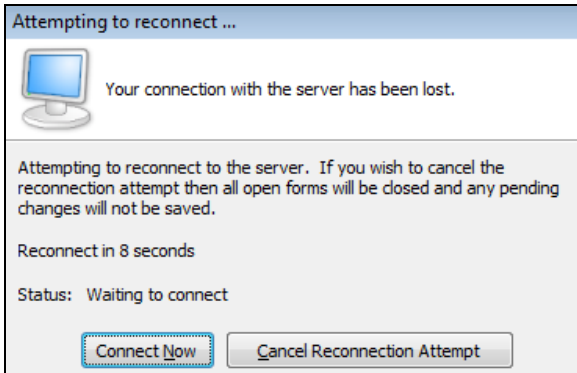
Click the **Check for Updates** button in the MAPS Configuration tool to view available updates.

If you need to apply earlier updates prior to the 6.6 update, please refer to the appropriate [release guide\(s\)](#) before proceeding.

It is possible to apply updates when users are on the system; however, to avoid the possibility of losing unsaved work we recommend applying updates during off hours.

[3. Allow update process to complete](#)

When applying updates to the MAPS service you will be temporarily disconnected from the server:



The update process may take a few minutes to complete. **Do NOT cancel the reconnection attempt or manually restart the server.** You will automatically be reconnected to the server once the update has been applied.

[4. Verify the latest version is installed](#)

To ensure that you are on the most current version, continue clicking the **Check for Updates** button and applying the updates until no new updates are available.

[Please Provide Us with Your Feedback!](#)

As always, we welcome any [feedback or suggestions](#) you may have. We very much appreciate your thoughts and suggestions, so please keep the great ideas coming!

REST Connector

The REST Connector is a data connection that can be used to connect to third-party data sources which provide one or more RESTful API endpoints.

The REST Connector:

- Supports XML or JSON data.
- Can be configured to map query filters (WHERE clauses) into API parameters.
- Does not require use of a script like the Cloud Connector.

Once you have created a REST Connector data connection, you can use it as a data source in the MAPS applications; for example, in an Argos DataBlock or FormFusion template. The API is then called whenever a query using the connection runs.

See also:

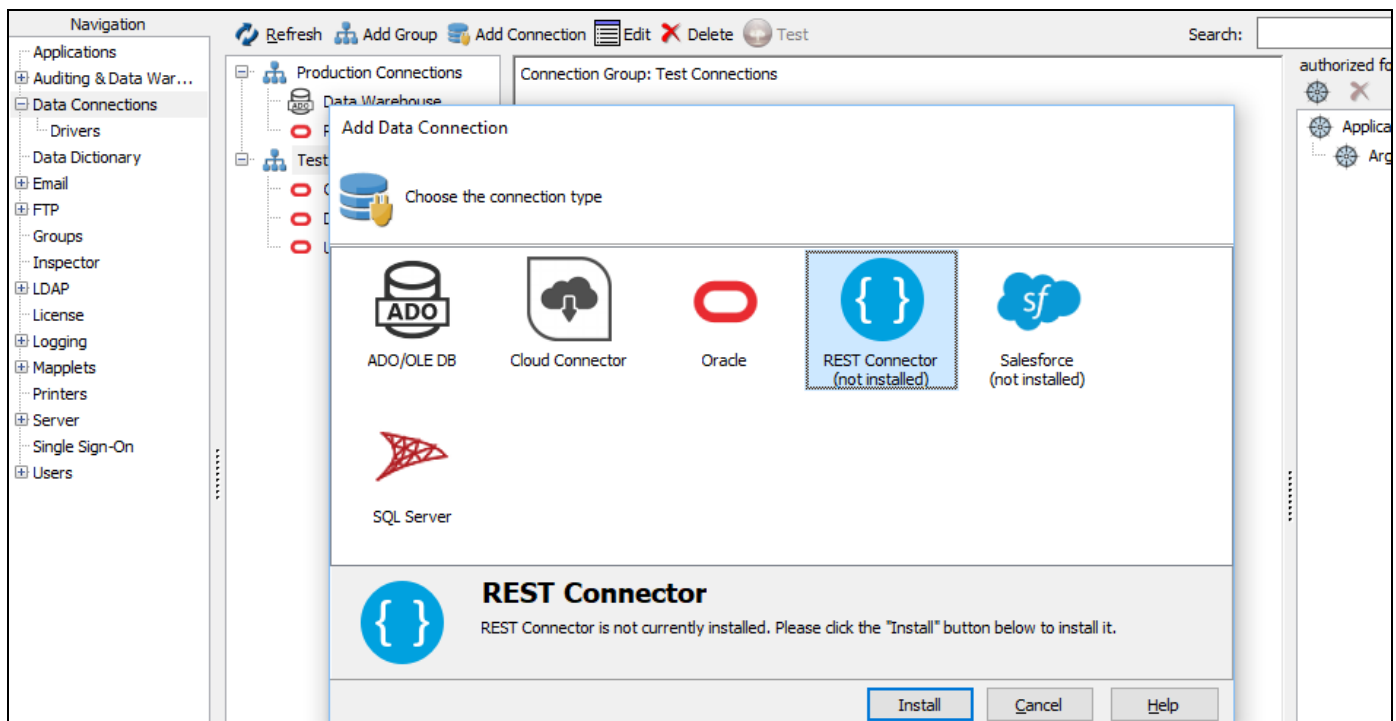
"REST Connector Configuration" below

"Using the REST Connector in Argos" on page 27

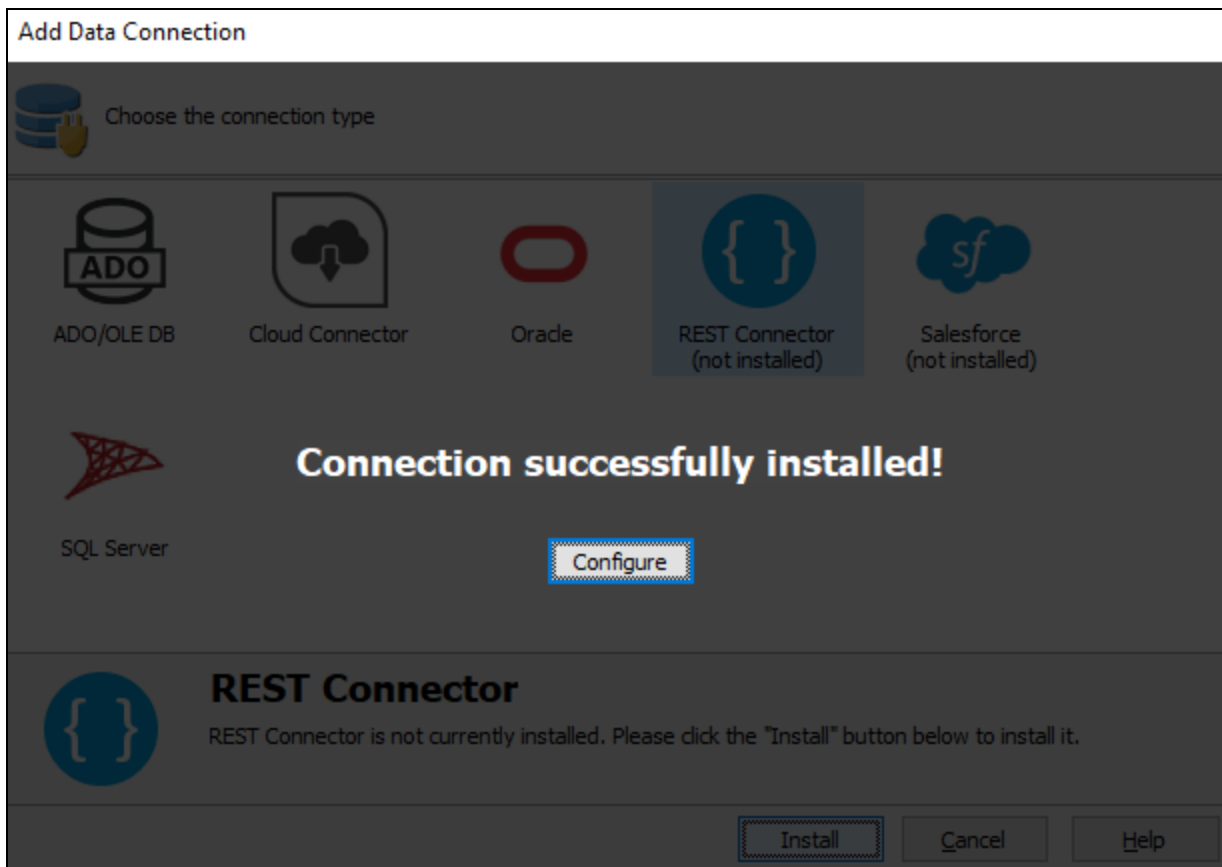
"XPath Format" on page 29

REST Connector Configuration

On the **Data Connections** screen in MAPS Config, select **Add Connection**. Choose **REST Connector** from the list of data connection types.

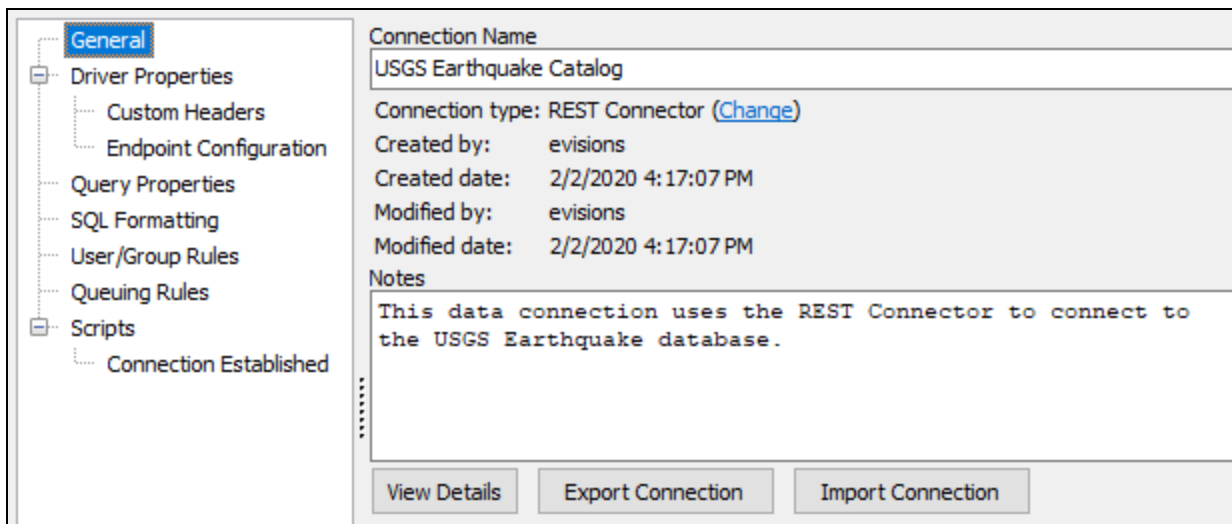


Select the **Install** button to download and install the driver.



When installation is complete, select **Configure** to configure the data connection properties.

On the **General** screen, enter a descriptive name for the data connection, and any desired notes.



Driver Properties

The screenshot shows the 'Driver Properties' configuration window. On the left is a tree view with the following items: General, Driver Properties (selected), Custom Headers, Endpoint Configuration, Query Properties, SQL Formatting, User/Group Rules, Queuing Rules, Scripts, and Connection Established. The main configuration area on the right contains the following fields:

- Base URL:** A text input field containing the URL `https://earthquake.usgs.gov/fdsnws/event/1/`.
- Response Type:** A dropdown menu with 'XML' selected.
- Logging:** A dropdown menu with 'Basic' selected.
- Authentication:** A dropdown menu with 'None' selected.

On the **Driver Properties** screen, enter the following parameters:

- **Base URL** - The base URL for the API to which you are trying to connect. A base URL does not typically return data itself, but is the root path of all endpoints provided by the API. Do not include endpoint names in the base URL-- these will be configured in the next step.
- **Response Type** - The format of the data returned by the API. Choose from JSON or XML. Note: if the API requires you to specify the desired format as a parameter, you will also need to include this parameter when configuring the endpoint(s) in the next step.
- **Logging** - The level of logging for this connection. Choose from Off (no logging), Basic (queries, numbers of rows returned, start/end time, and errors), or Full (adds cache queries, HTTP headers, request/response bodies, SSL negotiation, and additional details that may be helpful in troubleshooting). Logs are saved on the MAPS server, and can be viewed on the Logging > Logs screen in MAPS Config.

- **Authentication** - Choose from None, HTTP Basic, OAuth2, or [Bearer Token](#).
 - **None** - No authentication required, or authentication is being handled via a [custom header](#) in the HTTP request.
 - **HTTP Basic** - Specify a username and password.

Authentication:	
HTTP Basic	
Property	Value
User Name	myusername
Password	●●●●●●●●

- **OAuth2** - Authenticate using OAuth2. Requires registering MAPS with the application that is using OAuth2. You will be prompted to log in to the application the first time the connector runs, in order to obtain an access token.

Authentication:	
OAuth2	
Property	Value
Access Token URL	https://myapi/access-token-url/
Authorization URL	https://myapi/authorization-url/
Client ID	evisionstest
Client Secret	●●●●●●●●
Refresh Token URL	https://myapi/refresh-token-url/
Scope	profile
Use HTTPS On Callback	True

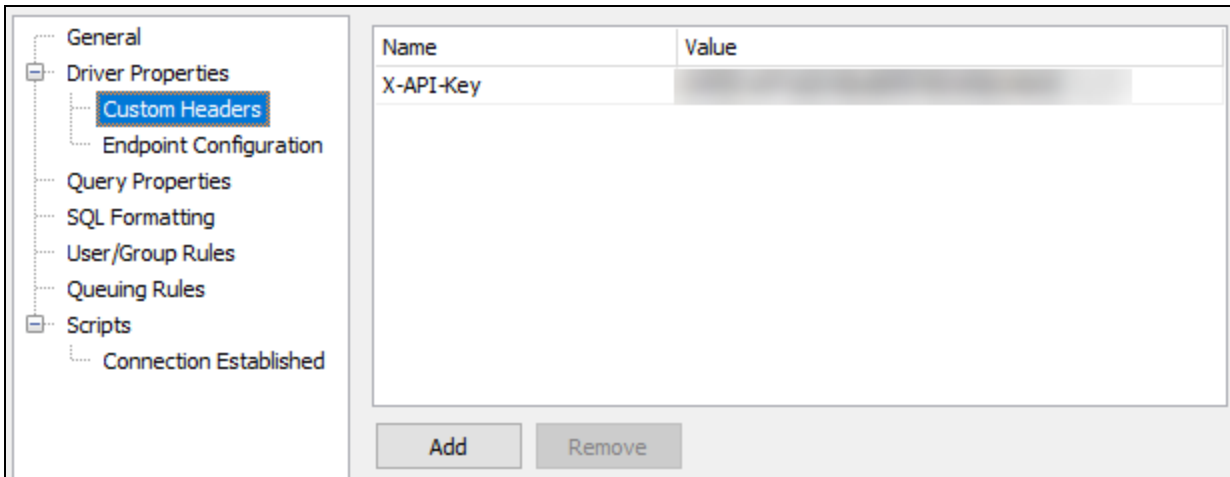
- **Client ID** - A unique identifier that is generated during the registration process.
- **Client Secret** - A password-like value specific to MAPS that is generated during the registration process. This value needs to remain secure.
- **Authorization URL** - The URL used to authenticate MAPS to the application that is using OAuth. This URL should be provided in the API documentation for the application (look for a "for developers" section).
- **Access Token URL** - The URL used to generate an access token after the user is authenticated. Access tokens usually only last for a brief time if not used in conjunction with a refresh token.
- **Refresh Token URL** - The URL used to receive a refresh token to extend authentication. Refresh tokens may be good for up to 30 days, depending on the API. This means that the user will not need to authenticate or receive a new access token for 30 days once a valid refresh token is received. Note: depending on the API, the refresh token URL may be the same as the access token URL.
- **Scope** - A list of the resources (permissions) that need to be included in the access token in order to use an endpoint, if required. Refer to your API documentation to determine whether a scope is required.
- **Use HTTPS On Callback** - Specifies whether HTTP or HTTPS should be used for the callback URL. True (HTTPS) or False (HTTP). A self-signed certificate will be automatically generated when using HTTPS.

OAuth Application Configuration

If you are configuring MAPS as the client in your OAuth application (the typical setup), you will need to specify **http://localhost:3333** or **https://localhost:3333** (depending on the **Use HTTPS On Callback** setting) as the callback URL in the OAuth application. This is the port MAPS uses for the HTTP listener that waits for the response from the OAuth application.

- **Bearer Token** - Authenticate using Bearer Tokens (also known as access Tokens). See "Bearer Token Configuration for a REST Connector" on page 20 for details.

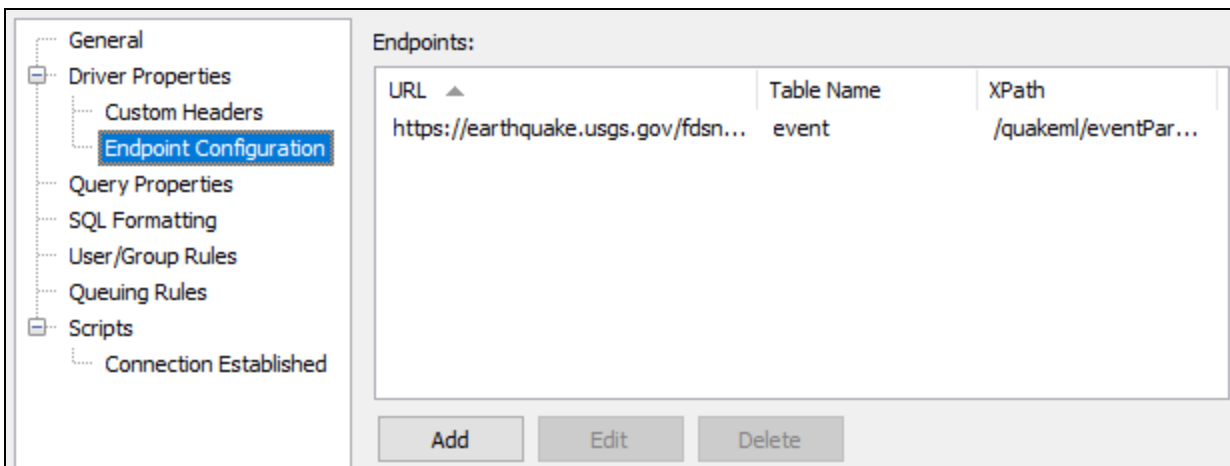
Custom Headers



The Custom Headers screen allows you to include optional custom headers in the HTTP request, if the API supports or requires it. Custom headers may be used for authentication or to specify an environment (such as a sandbox account) of the REST source. Type in the name of the header field, followed by its value. Use the Add and Remove buttons underneath the list of custom headers to add or remove additional custom headers as needed.

Endpoint Configuration

Next, select **Endpoint Configuration** under the Driver Properties tab.



Select **Add** at the bottom of the dialog to add an endpoint.

The Edit REST Endpoint dialog has three fields:

- **Endpoint** - The path to the endpoint. The path will be appended to the Base URL from the Driver Properties screen (shown above the field) to form the complete URL to the endpoint.
- **XPath** - An optional path to a repeating element within the XML or JSON response. This can be used to split the response into multiple rows. The [XPath format](#) differs depending on whether the response is XML or JSON. MAPS will attempt to detect the XPath for you when you generate the schema, but it may need to be modified to meet your needs.
- **Table Name** - The name for the table for this endpoint as it will appear in Argos. The table name is derived from the schema name, and is similar to a table name in a relational database. For example, if you set the table name to "earthquake", you could query it as `select * from earthquake`. If you leave the table name blank, MAPS will attempt to populate it from the data when generating the schema. You can then modify the name as desired.

Your Endpoints may require a variable value to be placed in the URL path. To use a variable in the Endpoint URL, use the format `[input.<variable_name>]`. For Example, if searching for students by ID, at the Endpoint: `students/{id}`, you would type: `student/[input.id]` in the Endpoint field. Whatever field is used for `[input.<variable_name>]` must be in the schema.

Note: This format will only work when the WHERE clause contains "=" for the variable. In addition, only a single value can be assigned to the variable.

Generate Schema

Select the **Generate Schema** button to generate the schema file by calling the API using the parameters and paths specified. You will be prompted to (temporarily) edit the constructed URL to include query parameters that limit the data. This will allow the API call to complete faster so that you can quickly retrieve a list of fields without downloading a lot of data.

As an example, for the earthquake API above we might add a start and end time to the query to limit it to a few results:

<https://earthquake.usgs.gov/fdsnws/event/1/query?format=xml&starttime=2014-01-01&endtime=2014-01-02>

Select **OK** to proceed. Depending on the amount of data being returned, it may take a moment to complete the request. If the request was successful, you should see an updated XPath, the schema name, and a list of fields in the Fields table below.

Ma Edit REST Endpoint

Configure the settings for the current REST endpoint.

Base URL: <http://earthquake.usgs.gov/fdsnws/event/1/>

Use this format to include a variable in the Endpoint URL: `[_input.<variable_name>]` [\(More Info\)](#)

Endpoint:

XPath:

Table Name:

To use query parameters in the URL via SQL WHERE Clauses, click "Map Query Parameters".

Fields:

Field Name ▲	Field Type	Query Parameters
creationInfo_agencyID	string	
creationInfo_creationTime	datetime	
creationInfo_version	integer	
description_text	string	
description_type	string	
event_id	string	
magnitude_creationInfo_agencyID	string	
magnitude_creationInfo_creationTime	datetime	
magnitude_evaluationMode	string	
magnitude_mag_uncertainty	double	
magnitude_mag_value	double	
magnitude_originID	string	
magnitude_stationCount	integer	

- **Add** - This will allow you to add fields to the schema that were not detected by the REST Connector.
- **Edit** - This will allow you to edit fields that are user-added, as well as fields added by the REST Connector.

- **Delete** - This will allow you to delete both the REST Connector and user-added fields.

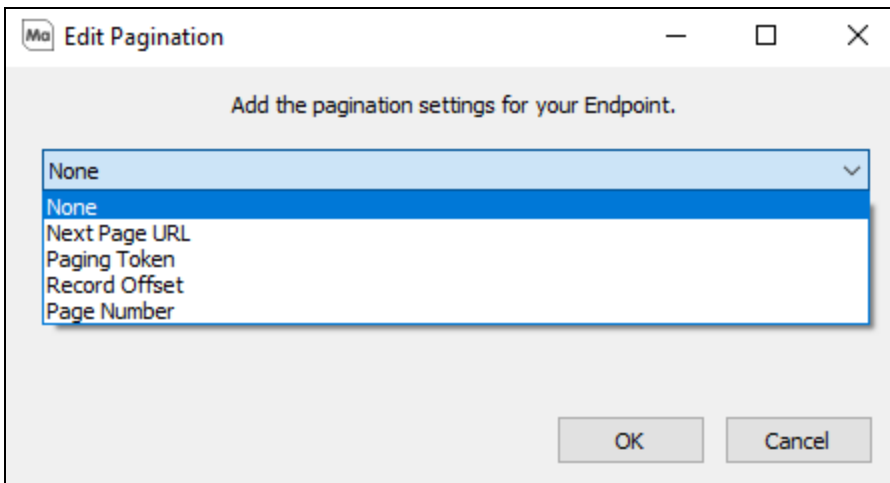
Edit Schema

The Edit Schema button opens the definition file for the schema. You may need to modify this file if a field is not detected as the correct data type. Otherwise, you should not need to make changes to the schema definition.

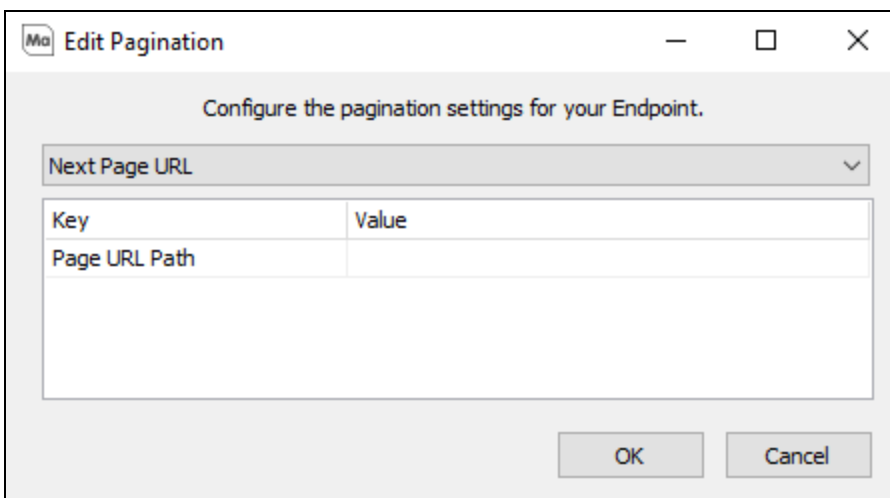
Pagination

The REST Connector supports customization of four different pagination styles, each of which allows all available sets of data to be returned. Once configured, the REST connection endpoint will automatically page the results when a query is executed and the full result will be returned.

Selecting the **Pagination** button will open the Pagination Options window.

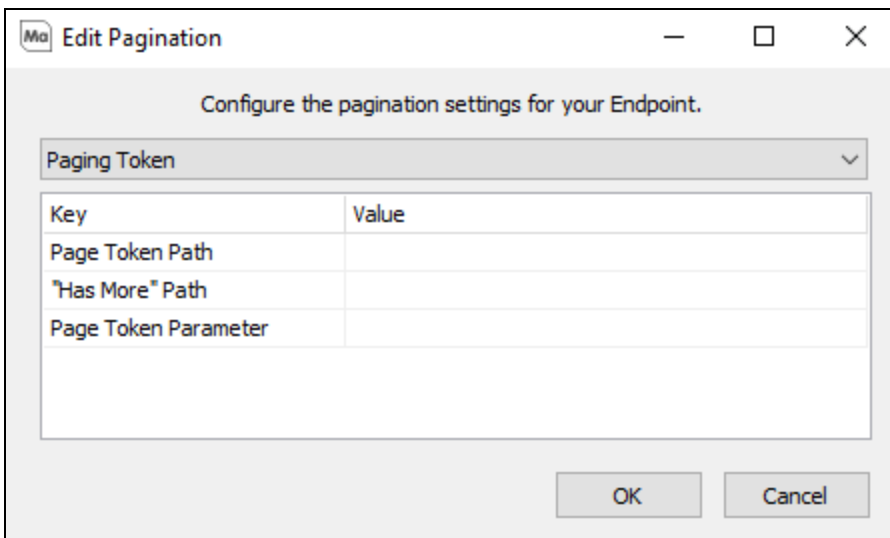


The REST Connector supports four types of paging:



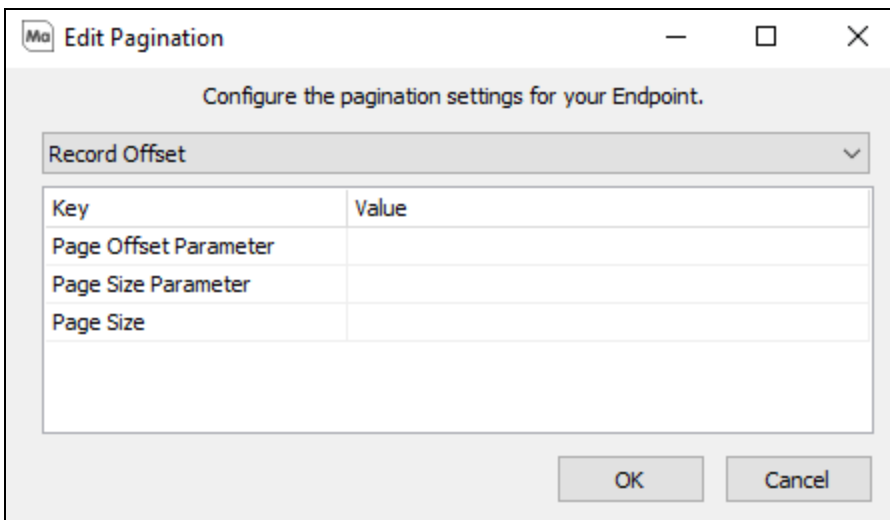
- **Next Page URL**

- **Page URL Path** - If the URL is passed in the response body, you must input the XPath to the element with the URL. A common configuration is to have the URL in the response header with a "Link" header. If this is the case, you will specify "header:Link" for this value.



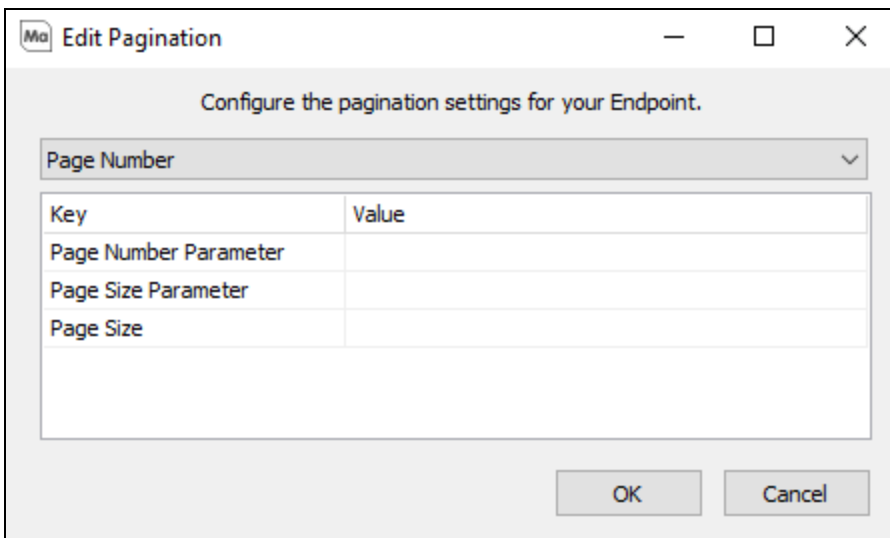
■ Paging Token

- **Page Token Path** - The path to the token that is passed from the API. It should be set to the XPath of the element.
- **"Has More Path"** - If a source includes this option, you will need to put the XPath to this variable in the response body.
- **Page Token Parameter** - The name of the parameter that is used to pass the token to the API. If the paging token is passed to the API via a query parameter, set this to the name of the parameter. If the paging token is passed to the API via the request body, set this to its XPath.



■ Record Offset

- **Page Offset Parameter** - The name of the query parameter that does the offsetting. A common name APIs use for this parameter is "offset".
- **Page Size Parameter** - This should be left blank when working with Record Offset pagination.
- **Page Size** - The Page Size should be set to the default size as listed on the API documentation for the endpoint that's being configured.



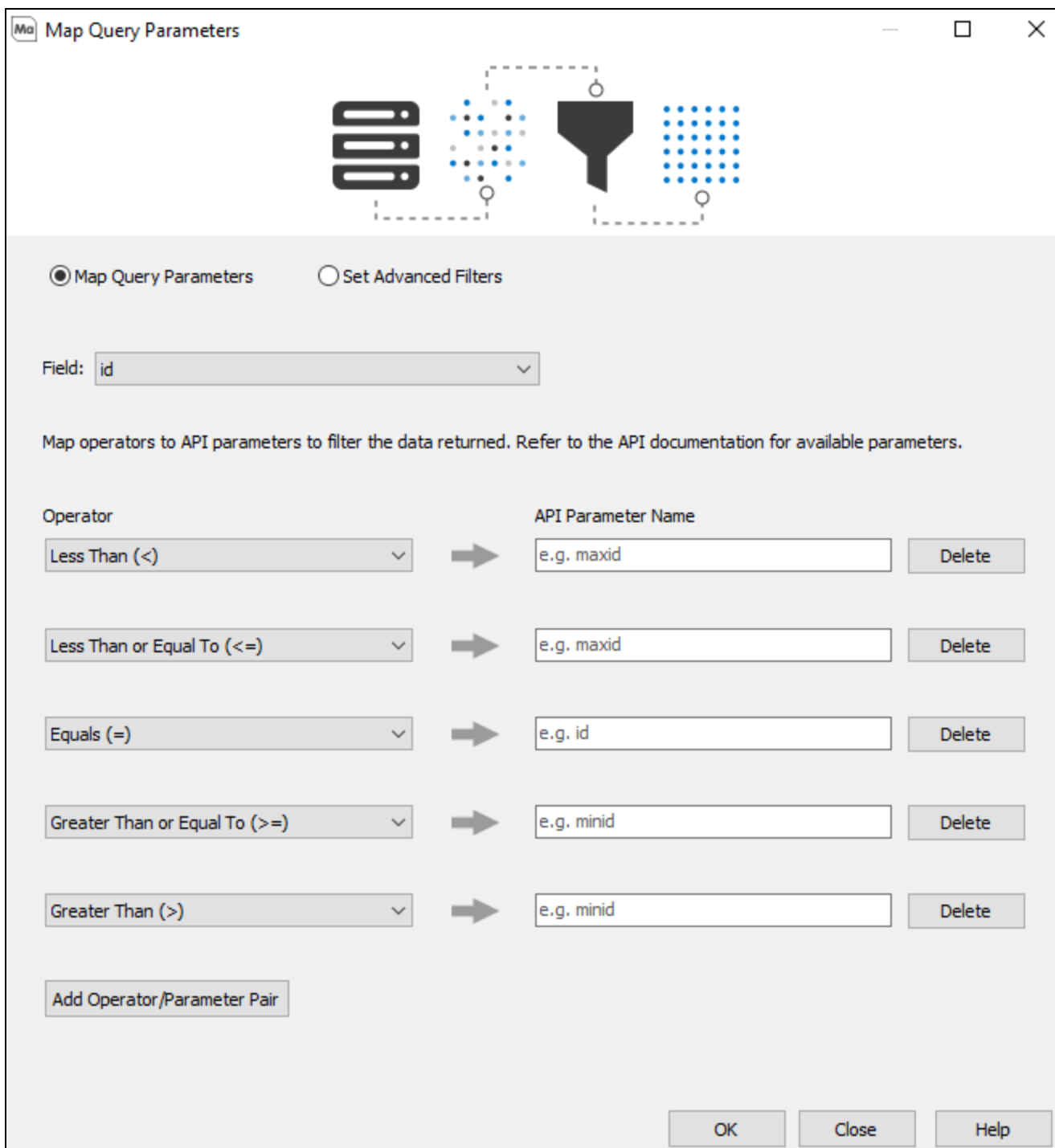
■ Page Number

- **Page Number Parameter** - The name of the query parameter that specifies the page number. A common name APIs use for this parameter is "page".
- **Page Size Parameter** - The name of the query parameter that specifies the size of the page. A common name APIs use for this parameter is "pagesize". This parameter is not included in all sources that use this pagination style. Sources that do not include this parameter do not allow you to control the number of records on each page. For this case, the Page Size should be set to this fixed number. When a source does include this parameter, the Page Size can be set to any number within the acceptable range for this parameter.
- **Page Size** - The number of records to be returned on each page.

Map Query Parameters

The next step is to map query parameters available in the API to the different comparison operators for each field that DataBlock Designers may want to use in their queries. This allows MAPS to intelligently convert (or 'map') the query written by the user into an API call that makes use of the available URL parameters provided by the API. Without this, the API call will request all records, and then MAPS will filter the results after they come back.

Select a field that relates to a parameter available in the API, then click the **Map Query Parameters** button.



You will default to the **Map Query Parameters** page, however you will also have the option to navigate to the **Set Advanced Filters** using the radio buttons near the top.

Use the **Add Operator/Parameter Pair** button to add the operator(s) you want to configure, then enter the parameters that should be associated with those operators.

In the example above, we are specifying the parameters that will be used if a query uses the less than or greater than comparison operators against the `magnitude_mag_value` field in its WHERE clause. If the query contains `WHERE magnitude_mag_value > 5` (i.e., limiting results to earthquakes with magnitudes greater than 5), the API call will now contain `&minmagnitude=5` in the parameter list.

Query Construction

Queries must be kept relatively simple in order to take advantage of any query parameter mappings you have configured. See the Query Construction section at the end of the [Using the REST Connector in Argos](#) page for examples of supported SQL.

Set Advanced API Filters

Ma Set Advanced API Filters

Map Query Parameters Set Advanced Filters

Filter Name:
criteria

API Filter Values:

Filter Type	Used Fields
Object	subject_id

Add Edit Delete

SQL Sample:
`select *
from courses
where courses."subject_id" = 'value1'`

GET Preview:
`https://integrate.elluciancloud.com/api/courses
?criteria=
{\"subject\":{\"id\":\"value1\"}}`

OK Close Help

Selecting the **Set Advanced Filters** radio button will allow you to configure JSON objects/arrays that will be used in the URL of an endpoint.

- **Filter Name** - The specified parameter name of the filter you are adding.
- **API Filter Values** - The list of filters already configured.
- **Add, Edit** and **Delete** - Allows you to add, edit, or delete an API filter.
- **SQL Sample** - This displays an example of the simplest SQL statement required to trigger the JSON Object Filter.
- **GET Preview** - This displays an example of a GET request being sent, including the JSON query parameters.

Selecting **Add**, **Edit**, or **Delete** will open the Edit Advanced API Filter screen.

Edit Advanced API Filter

JSON Type: Object

Object Key: subject

Item Key: id

Source: subject_id

SQL Sample:

```
select *
from courses
where courses."subject_id" =
'value1'
```

GET Preview:

```
https://integrate.elluciancloud.com/api/courses
?criteria=
{"subject":{"id":"value1"}}
```

OK Close Help

- **JSON Type** - A drop-down with the option of selecting **Object** or **Array**.

When **Object** is selected as the JSON type:

- **Object Key** - The input to specify the object key.
- **Item Key** - The input to specify the item key.
- **Source** - A drop-down used to select the schema field that will trigger the filter.

When **Array** is selected as the JSON type:

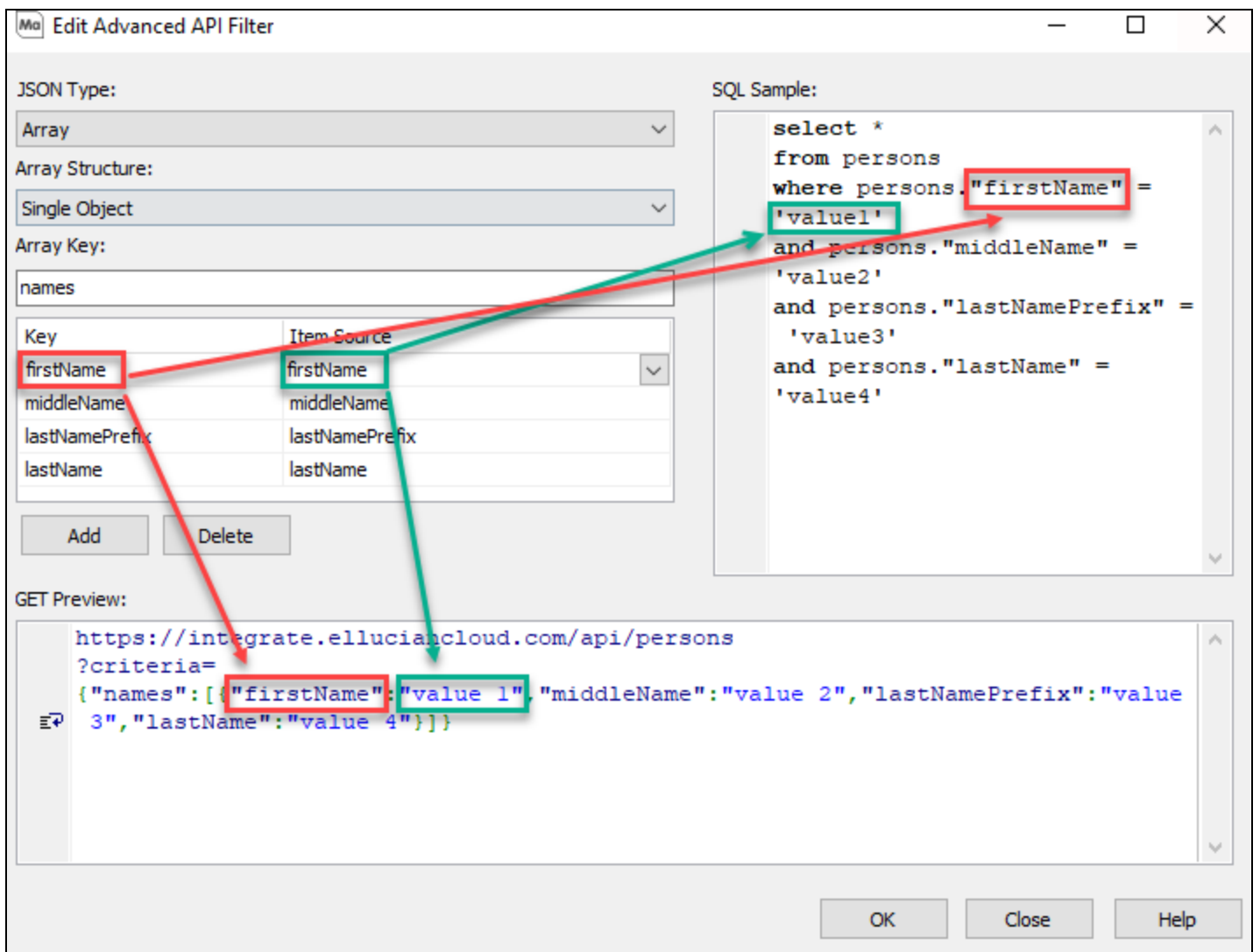
When **Array** is selected as the JSON type:

- **Array Structure** - A drop-down with the option of selecting **Comma Separated Objects** or **Single Objects**.
- **Array Key** - The name of the Array.

When **Comma Separated Objects** is selected as your Array Structure, you will see the following options:

- **Object Key Required** - If checked, you will be presented with an **Object Key** input.
- **Item Key** - The item identifier.
- **Object Key** - The object identifier.

When **Single Object** is selected as the Array Structure, you will see the following options:



- **Key/Item Source Multicolumn** - The Key/Item Source pairs within the JSON object. The Key represents the identifier for each item within the object. The Item Source is the field from the schema that triggers the inclusion of the value in the object when used in a SQL statement.
- **Add and Delete** - These buttons allow you to create a new Key/Item Source pair, or remove the selected Key/Item Source pair.

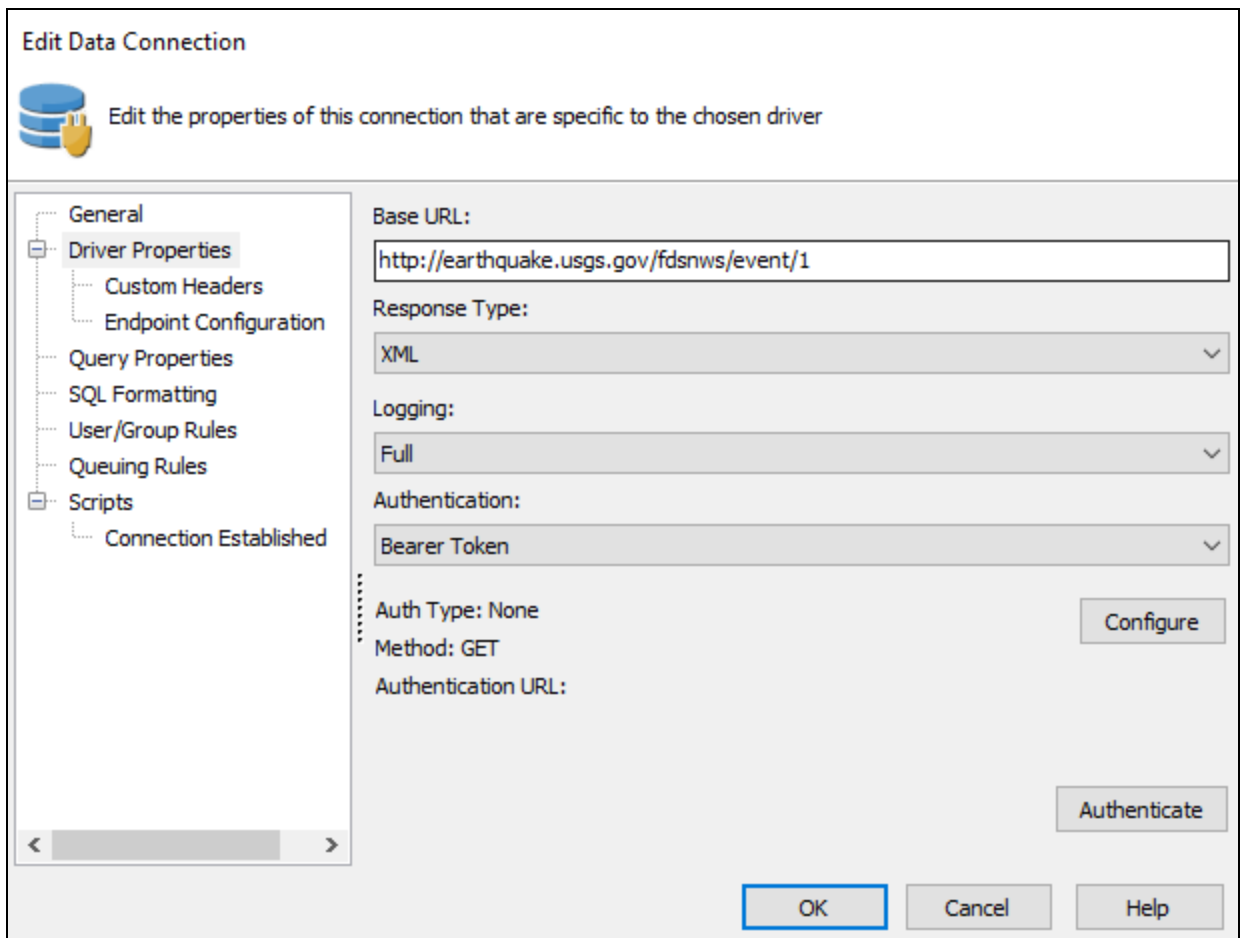
The REST data connection configuration is now complete. You do not need to modify any values on the Query Properties or SQL Formatting screens. If you want to set up specific users or groups on the User/Group Rules screen, each group that you want to be able to use this connection should be set to the default of "Use connection user name and password".

Lastly, authorize the connection for use with the desired application(s).

[Bearer Token Configuration for a REST Connector](#)

Bearer Token authentication is necessary for setting up connections to some REST sources. Occasionally REST APIs may refer to Bearer Tokens as Access Tokens in API documentation.

From the Edit Data Connection screen, select **Bearer Token** from the Authentication drop-down. The **Configure** button will appear below.



Selecting **Configure** will bring up the Bearer Token Authentication screen.

Bearer Token Authentication

Method: GET

Authorization URL:

Authentication Type: None

Key	Value

Custom Headers (Optional) Data/Parameters (Optional):

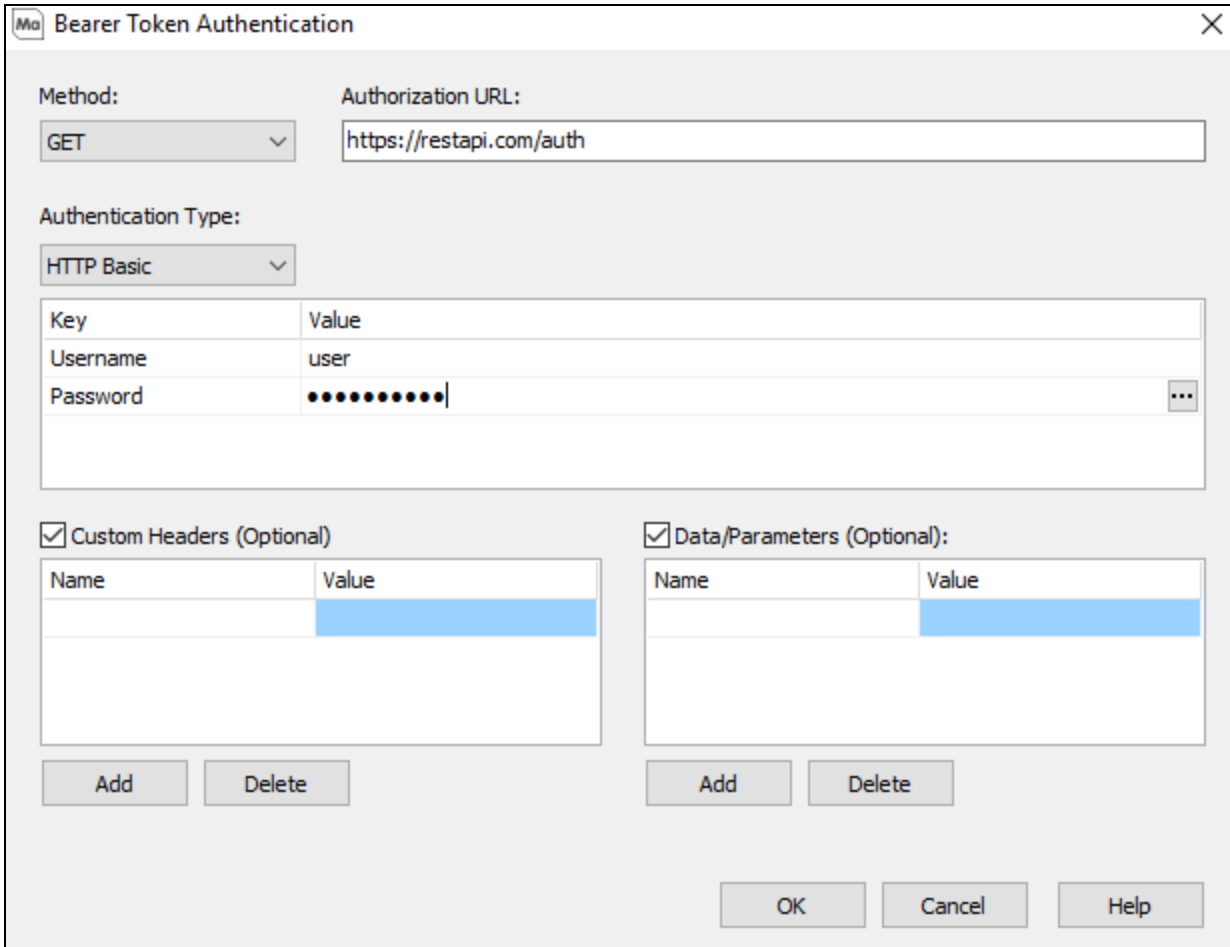
OK Close Help

- **Method** - This allows you to select between the POST or GET methods, which describe the request method that the API expects when authenticating, as outlined in your API documentation.
- **Authorization URL** - This describes the URL to which MAPS will send a request to receive a Bearer Token. This will typically be found in the Authentication section of your API documentation.
- **Authentication Type** - This describes the authentication scheme that is used to acquire the Bearer/Access token. The two supported Authentication types are **HTTP Basic**, and **Bearer Token**.
- **Custom Header (Optional)** - This describes any additional header values that can be passed during authentication.
- **Data/Parameters (Optional)** - This describes any additional data that can be passed during authentication. This section should only be used if the POST method is selected. The data will not be sent if using the GET method. The data will be sent in the message body in the format application/x-www-form-urlencoded.

Note: The purpose of the Authorization URL, which is specific to Bearer Token and OAuth2 authorization, is to allow MAPS to access data from the API. This is not to be confused with the Base URL, which is the root path of all endpoints provided by the API.

HTTP Basic

Selecting **HTTP Basic** as your Authentication Type requires you to provide a username and password in order for your API to supply a Bearer Token.

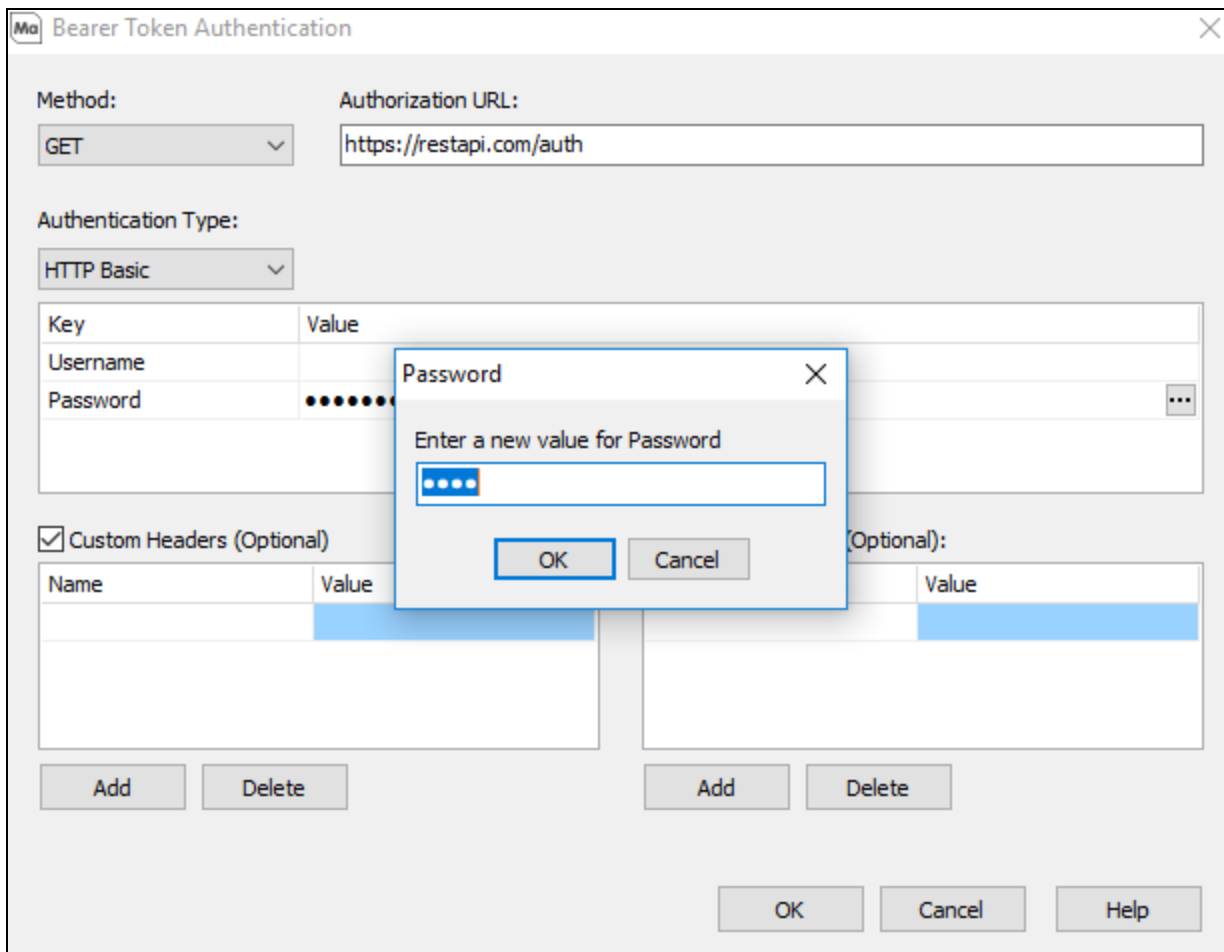


The screenshot shows a dialog box titled "Bearer Token Authentication" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Method:** A dropdown menu set to "GET".
- Authorization URL:** A text input field containing "https://restapi.com/auth".
- Authentication Type:** A dropdown menu set to "HTTP Basic".
- Key/Value Table:** A table with two columns: "Key" and "Value". It contains three rows:
 - Key: Username, Value: user
 - Key: Password, Value: a series of 12 black dots (masked password), with an ellipsis button (three dots) to the right of the input field.
 - Key: (empty), Value: (empty)
- Custom Headers (Optional):** A checked checkbox followed by a table with "Name" and "Value" columns. The table is currently empty.
- Data/Parameters (Optional):** A checked checkbox followed by a table with "Name" and "Value" columns. The table is currently empty.

At the bottom of the dialog, there are two pairs of "Add" and "Delete" buttons, one pair for each of the optional sections. At the very bottom, there are three buttons: "OK", "Cancel", and "Help".

Selecting the ellipsis on the **Password** Authentication Value will open the **Password** dialog.



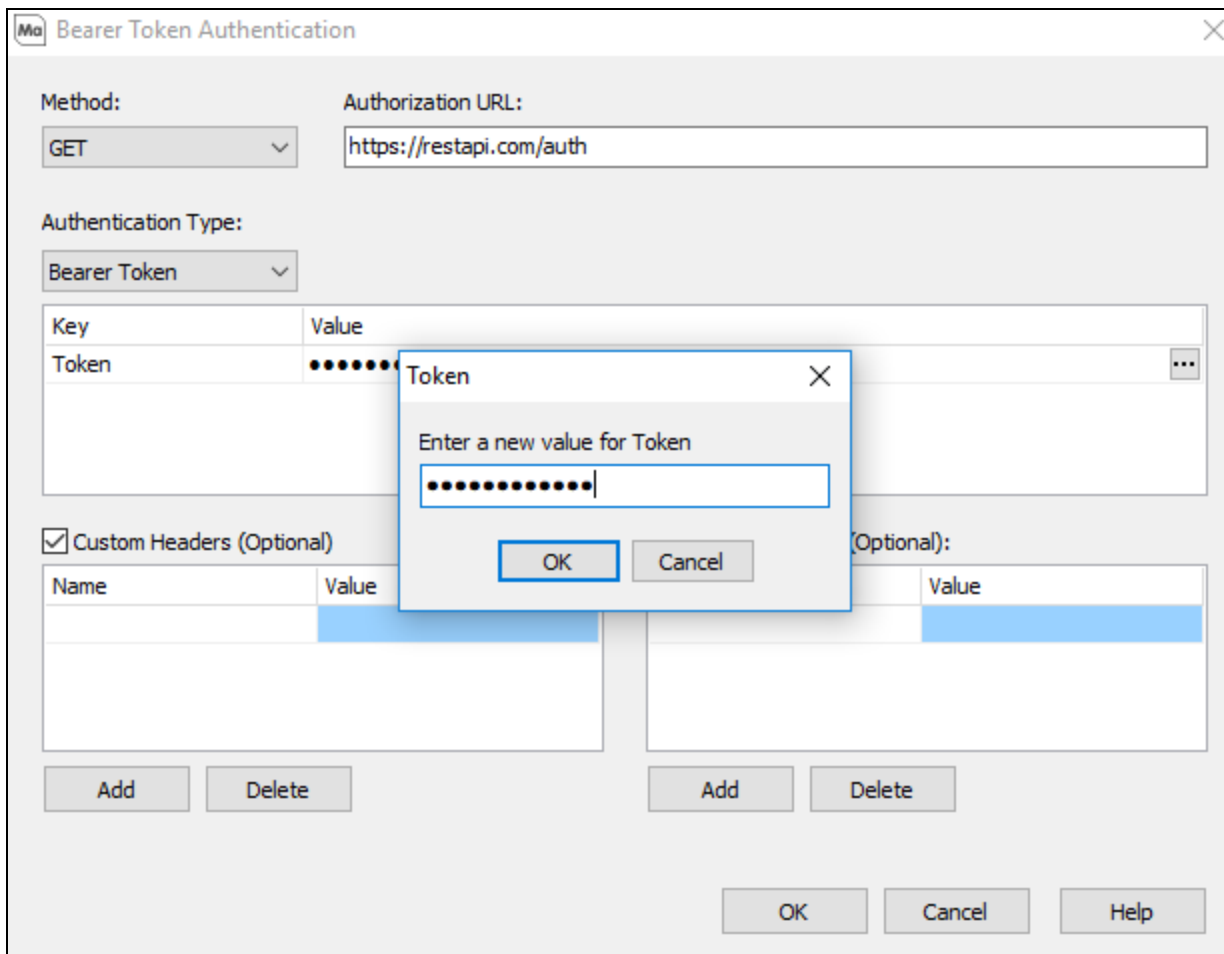
Bearer Token

Selecting **Bearer Token** as your Authentication Type will require you to provide a Token that will be sent to the Authorization URL.

The screenshot shows a dialog box titled "Bearer Token Authentication" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Method:** A dropdown menu set to "GET".
- Authorization URL:** A text input field containing "https://restapi.com/auth".
- Authentication Type:** A dropdown menu set to "Bearer Token".
- Token Table:** A table with two columns: "Key" and "Value". The first row has "Token" in the "Key" column and a series of dots in the "Value" column. An ellipsis button (...) is located at the end of the "Value" cell.
- Custom Headers (Optional):** A checked checkbox followed by a table with "Name" and "Value" columns. The "Value" cell is highlighted in blue.
- Data/Parameters (Optional):** A checked checkbox followed by a table with "Name" and "Value" columns. The "Value" cell is highlighted in blue.
- Buttons:** "Add" and "Delete" buttons are located below each of the optional sections. At the bottom of the dialog are "OK", "Cancel", and "Help" buttons.

Selecting the ellipsis on the Token Authentication Value will open the **Token** dialog. Please refer to your API documentation for your **Token Value**, or instructions on how to receive your **Token Value**.



See also:

"REST Connector Configuration" on page 6

"Using the REST Connector in Argos" on the next page

"XPath Format" on page 29

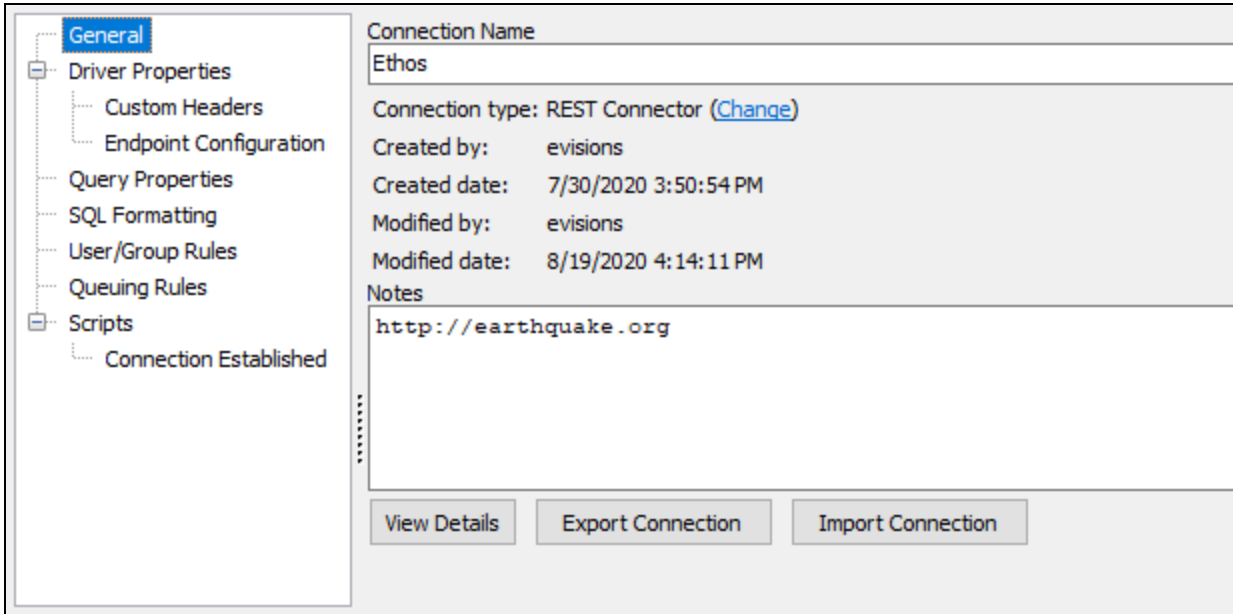
[Sharing a REST Connector Data Connection](#)

Once you have configured a REST Connector, you can share the configuration with other MAPS users, both inside and outside your organization.

When you export a REST Connector, authentication information **is not** exported. Authentication will need to be reconfigured after importing the file.

Exporting a Connection

The first step is to export the connection. Select **Export Connection** from the main configuration dialog, followed by the **Save** button..



The screenshot shows the 'General' tab of a configuration dialog. On the left is a tree view with 'General' selected. The main area contains the following fields and information:

- Connection Name:** Ethos
- Connection type:** REST Connector (Change)
- Created by:** evisions
- Created date:** 7/30/2020 3:50:54 PM
- Modified by:** evisions
- Modified date:** 8/19/2020 4:14:11 PM
- Notes:** http://earthquake.org

At the bottom are three buttons: 'View Details', 'Export Connection', and 'Import Connection'.

The file is exported as a `.mapsconnection` file to a location that you choose. Once you save the file, you can then send it to another user just as you would any other file.

Importing a Connection

To import a REST Connector data connection file, first determine whether are creating a new connection or if you want to overwrite an existing connection:

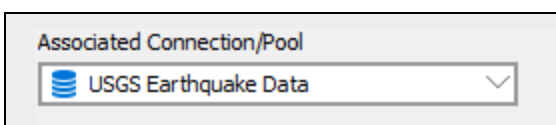
- To create a new REST Connector data connection, select **Add Connection** from the MAPS Data Connections page, then select the **REST Connector** option.
- To overwrite an existing REST Connector data connection, select the connection and then select **Edit** from the MAPS Data Connections page

Select **Import Connection** from the Add Data Connection page.

Navigate to the location of the `.mapsconnection` file, select the file, and select **Open**.

Using the REST Connector in Argos

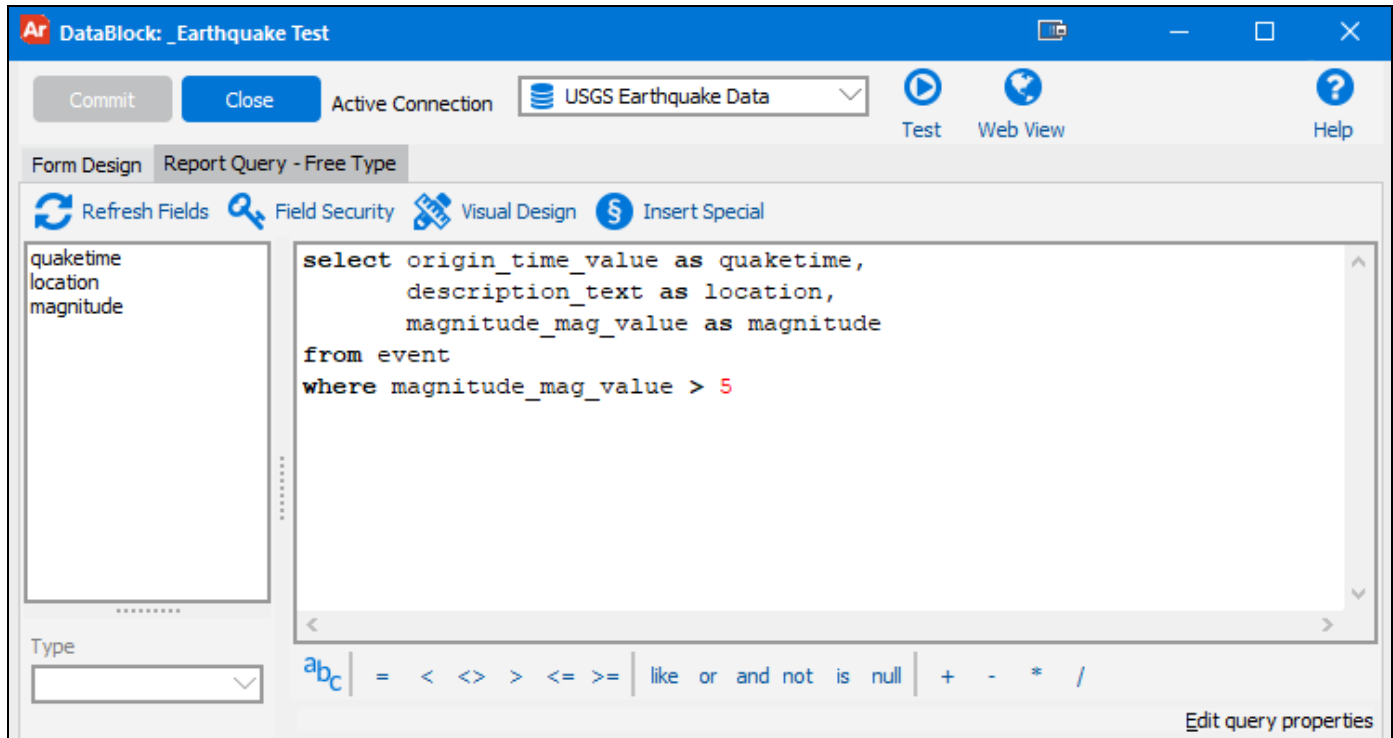
To use the REST Connector in Argos, create a DataBlock and then select your REST Connector data connection in the **Associated Connection/Pool** dropdown.



The screenshot shows a dropdown menu with the label 'Associated Connection/Pool'. The selected option is 'USGS Earthquake Data'.

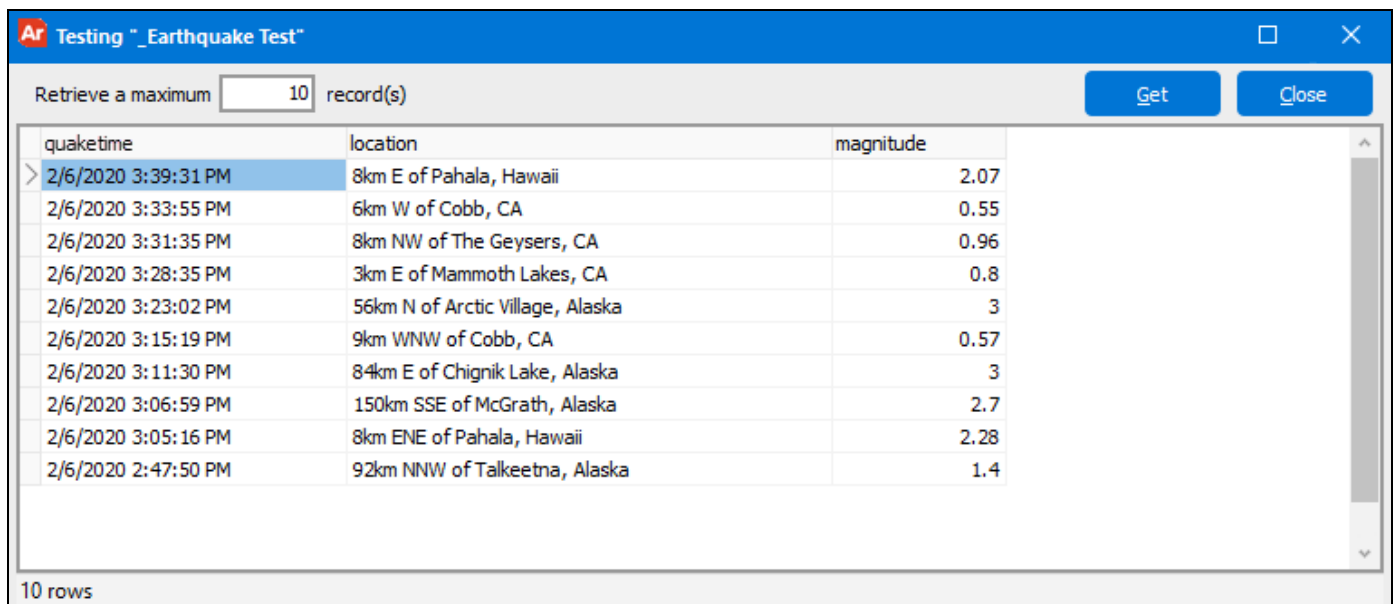
You can now use the REST Connector in your queries as you would any other data connection. This could be in the Visual Designer, free type report query, multicolumn list boxes, or in any other location where you can use SQL in Argos.

The following example shows how a query might look in the Free Type report query editor.



Note: You can avoid or reduce the use of underscores in your field names by specifying the XPath to the field when configuring the endpoint.

Validate and test the query to ensure that you are getting back the expected results:



You can now use the connector to create reports using this report query.

[Query Construction when Using Mapped Query Parameters](#)

If your REST Connector is configured to [map query parameters](#) to API parameters, you will need to know which fields and operators have been mapped in order to have the filtering done as part of the API call and not by Argos after the data has been returned. If Argos does the filtering, you will still get the expected results; however, the query may take longer to run and you may be requesting more data than intended from the API.

Supported SQL clauses include:

- SELECT
- INTO
- FROM
- JOIN
- WHERE
- GROUP BY
- HAVING
- UNION
- ORDER BY
- LIMIT

Note that your WHERE clauses must be kept very simple. You can use multiple mapped fields separated by AND keywords. For example:

- `WHERE magnitude_mag_value > 8 AND description_text is not null`

More complex WHERE clauses involving OR clauses, nested queries, or subqueries are not supported. For example, all of the following WHERE clauses will retrieve all data from the API and then apply the filtering in the WHERE clause after the data has been returned to Argos:

- `WHERE magnitude_mag_value > 8 OR description_text is null`
- `WHERE magnitude_mag_value > 8 OR (magnitude_mag_value > 5 AND description_text is not null)`
- `WHERE description_text in (select description_text from event)`

[XPath Format](#)

The XPath is the path(s) to the repeating fields in the API response that correspond to the table(s) that you want to query.

[XML Endpoints](#)

Let's look at the following example document that might be returned for an XML endpoint:

```
<Contributors>
  <Contributor>admin</Contributor>
  <Contributor>ak</Contributor>
  <Contributor>at</Contributor>
  <Contributor>atlas</Contributor>
  <Contributor>ok</Contributor>
  <Contributor>us</Contributor>
  <Contributor>uu</Contributor>
  <Contributor>uw</Contributor>
</Contributors>
```

In this case, we want a table called `contributor` that contains all `<Contributor>` records. Therefore, the XPath needs to be the path to the repeating `Contributor` elements:

```
/Contributors/Contributor
```

You can create more than one table from a single XML response. Consider the following:

```
<Contributors>
  <Contributor>
    <name>admin</name>
    <description>
      <value1>type</value1>
      <value2>1</value2>
    </description>
  </Contributor>
  <Contributor>
    <name>ak</name>
    <description>
      <value1>type</value1>
      <value2>1</value2>
    </description>
  </Contributor>
  <Contributor>
    <name>at</name>
    <description>
      <value1>type</value1>
      <value2>1</value2>
    </description>
  </Contributor>
  <Contributor>
    <name>atlas</name>
    <description>
      <value1>type</value1>
      <value2>1</value2>
    </description>
  </Contributor>
</Contributors>
```

The XPath to the `contributor` table is still `/Contributors/Contributor`. We could also decide we want to create a second table for `<description>`. Its XPath would be `/Contributors/Contributor/description`. To include both tables in the XPath field, separate the two XPaths using a semicolon:

```
/Contributors/Contributor;/Contributors/Contributor/description
```

JSON Endpoints

In the case of JSON, we want to use JSONPath syntax to specify the XPath for the tables. Consider the following JSON example:

```
{
  "catalogs": [
    "38457511",
    "=c",
    "ak",
    "at",
    "atlas",
    "av",
    "cdmg",
    "cgs",
    "choy",
    "ci",
    "sc",
    "se",
    "unknown",
    "us",
    "ushis",
    "uu",
    "uw"
  ],
}
```

The start of the JSON object is represented as '\$'. To specify that `catalogs` should be a table, we would construct the XPath as:

```
$.catalogs
```

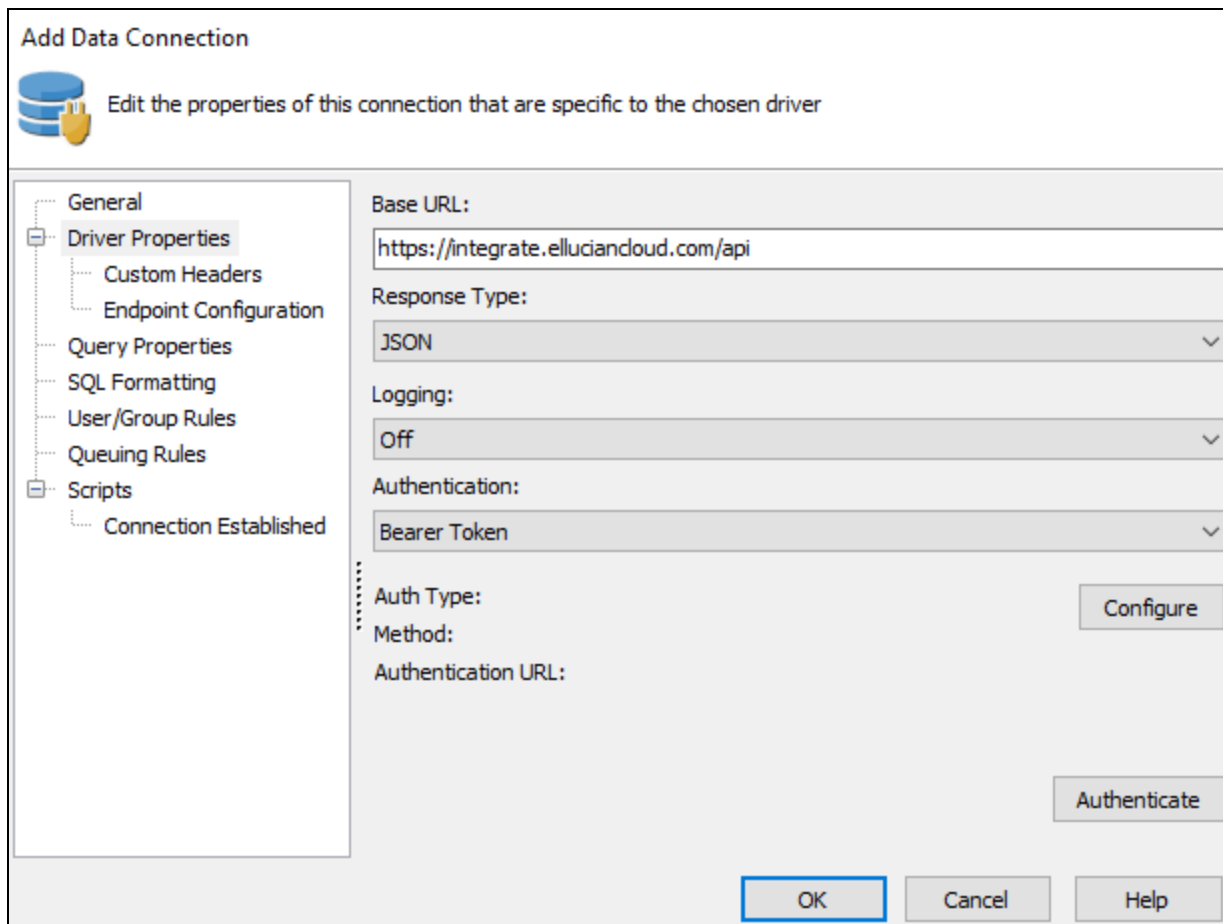
You can specify multiple tables from one JSON response by separating their JSONPaths with a semicolon, similar to the second XML example above.

Example of Creating a REST Connector for Ellucian Ethos

The REST Connector is a module that allows you to create a data connection to a wide variety of existing RESTful APIs. REST Data connections can be configured with multiple API endpoints, each of which functions similarly to relational database tables that can be queried by Argos and other MAPS applications.

Driver Properties

Navigate to the Add Data Connection screen to begin configuring your REST Connector for Ellucian Ethos.



Add Data Connection

Edit the properties of this connection that are specific to the chosen driver

- General
- Driver Properties**
 - Custom Headers
 - Endpoint Configuration
- Query Properties
- SQL Formatting
- User/Group Rules
- Queuing Rules
- Scripts
 - Connection Established

Base URL:

Response Type:
JSON

Logging:
Off

Authentication:
Bearer Token

Auth Type:

Method:

Authentication URL:

From the **Add Data Connection** Screen, navigate to **Driver Properties**, and enter your **Base URL**, and then input your **Response Type**, **Logging** if desired, and **Authentication** from the drop-down lists. Ellucian Ethos API returns data in the JSON format, so for this example select **JSON** as your Response Type.

For Authentication, select Bearer Token, as Ellucian Ethos will only provide access to its resources through Bearer Token authentication. When you have all the necessary information in your fields, select the **Configure** button.

Bearer Token Authentication

Ma Bearer Token Authentication

Method: POST Authorization URL: https://integrate.eluciancloud.com/auth

Authentication Type: Bearer Token

Key	Value
Token

Custom Headers (Optional) Data/Parameters (Optional)

OK Cancel Help

From the Bearer Token Authentication screen, select **POST** as your Method from the drop-down list. Then, type in your Authorization URL for Ellucian Ethos, and select **Bearer Token** as your Authentication Type from the drop-down list. Ellucian Ethos uses a two-step Bearer Token method of authenticating.

Next, select the ellipsis to open the **Token** dialog.

Ma Bearer Token Authentication

Method: POST Authorization URL: https://integrate.eluciancloud.com/auth

Authentication Type: Bearer Token

Key	Value
Token

Custom Headers (Optional) Data/Parameters (Optional)

OK Cancel Help

Token

Enter a new value for Token

.....

OK Cancel

Here you will input your API key. If you have not been provided an API key, please open a ticket with Ellucian. Select **OK** to close the Token dialog, and select **OK** once more from the Bearer Token Authentication screen to continue.

Edit Data Connection

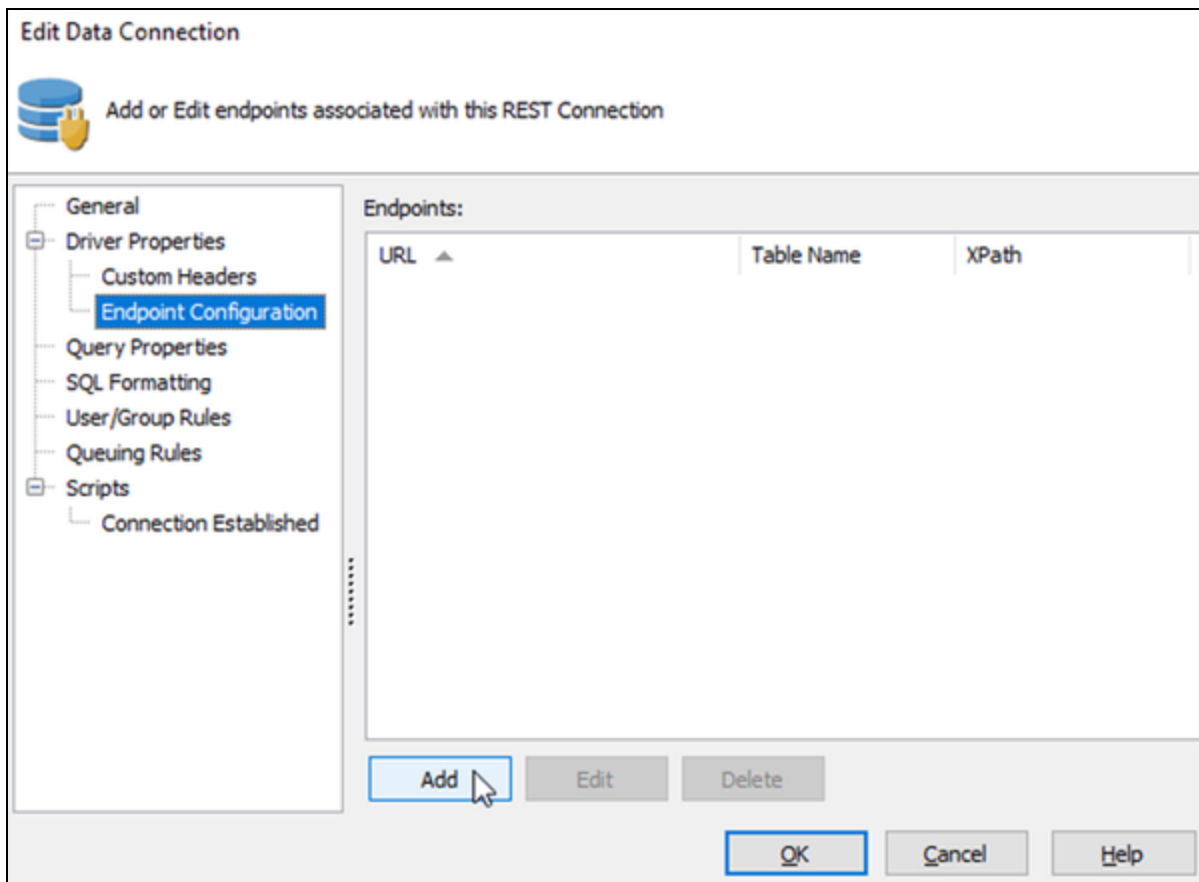
From the Edit Data Connections screen, navigate to **Custom Headers** tab.

Name	Value
Accept	application/json

Enter **Accept** into the Name field, and **application/json** for the Value field. This field will ensure that the data will be returned in a JSON format.

Note: If you'd like to use a previous version of the APIs, please refer to your API documentation.

Next, navigate to the Endpoint Configuration tab.



Select **Add**.

Edit REST Endpoint

Enter the endpoint URL and the XPath, then click 'Generate Schema' to continue.

Base URL: `https://integrate.elluciancloud.com/api/`

Use this format to include a variable in the Endpoint URL: `[input.<variable_name>]` ([More Info](#))

Endpoint:

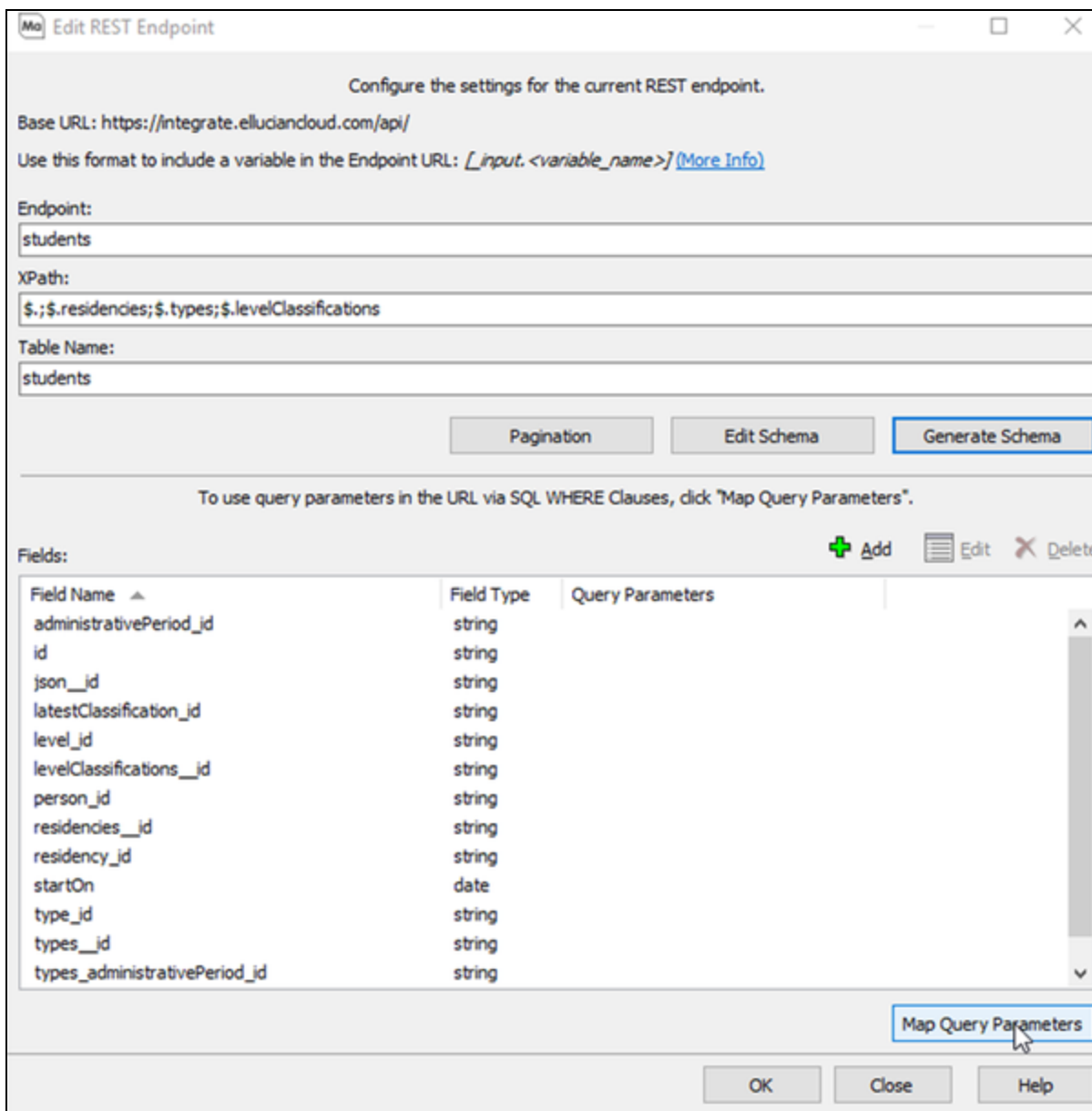
XPath:

Table Name:

In the **Endpoint** field, type the name of your endpoint. For this example we are using the **students** endpoint.

Next, specify your **Table Name**. It is recommended that your Table Name be similar to the name of your endpoint.

When you're finished, select **Generate Schema**.

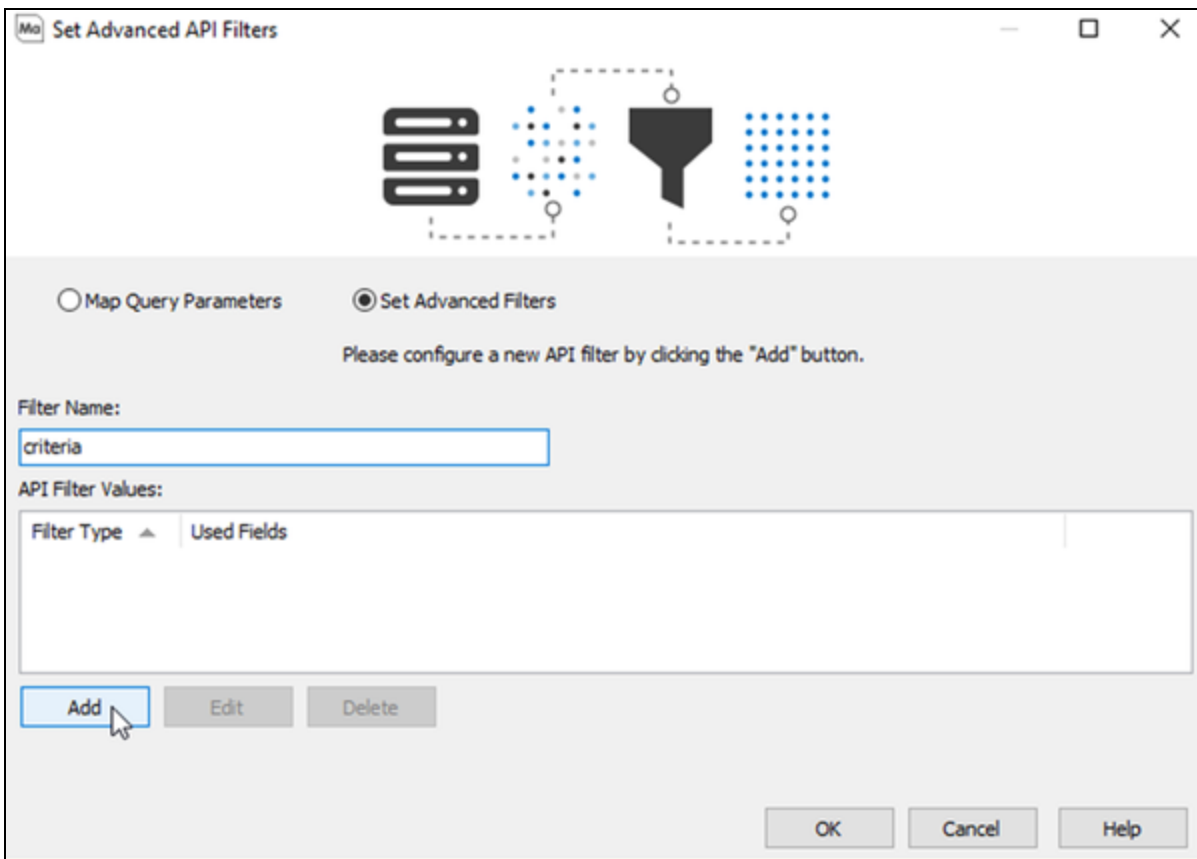


From here you can select **OK** to complete your connection, or you can select **Map Query Parameters** to filter your data from Ellucian Ethos by using the APIs.

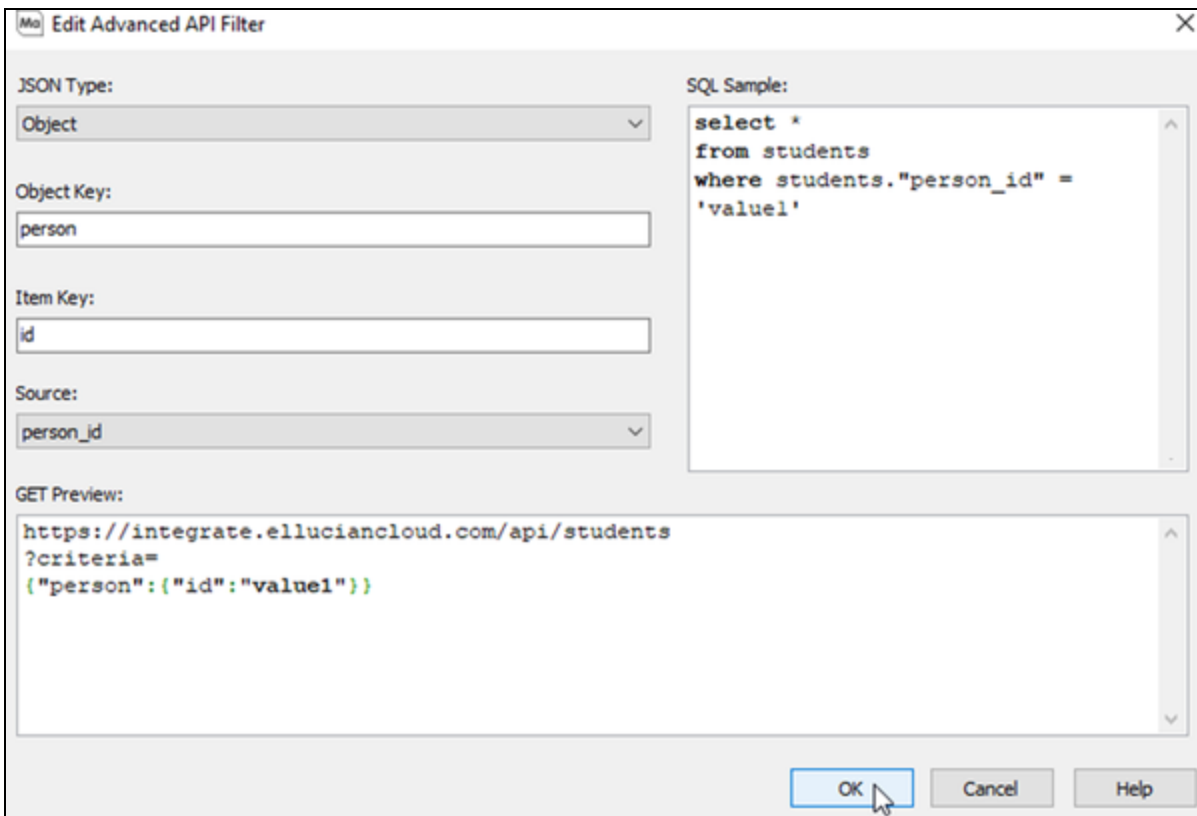
Set Advanced APIs Filters

Advanced API filters can be used to narrow down your data set from Ellucian Ethos. Filters used in MAPS queries may not provide Ellucian Ethos APIs with the proper information it needs to filter the data. Specifically, your queries may run longer than anticipated, or you may come across errors when the data set you are trying to return is too large. Because of this, we strongly recommend that you configure advanced API filters for any specific data that you may need.

From the Map Query Parameters screen, select the **Set Advanced Filters** radio button.



For **Filter Name**, enter the name of the parameter for your advanced filter. For our **students** example we are using **Criteria**. When finished, select **Add** to add a new API filter via the Edit Advanced API Filters screen.



For this example, we are setting up the **person_id** filter for the **students** endpoint. This will allow us to filter the data from the **students** endpoint by searching for a student by their **person_id**.

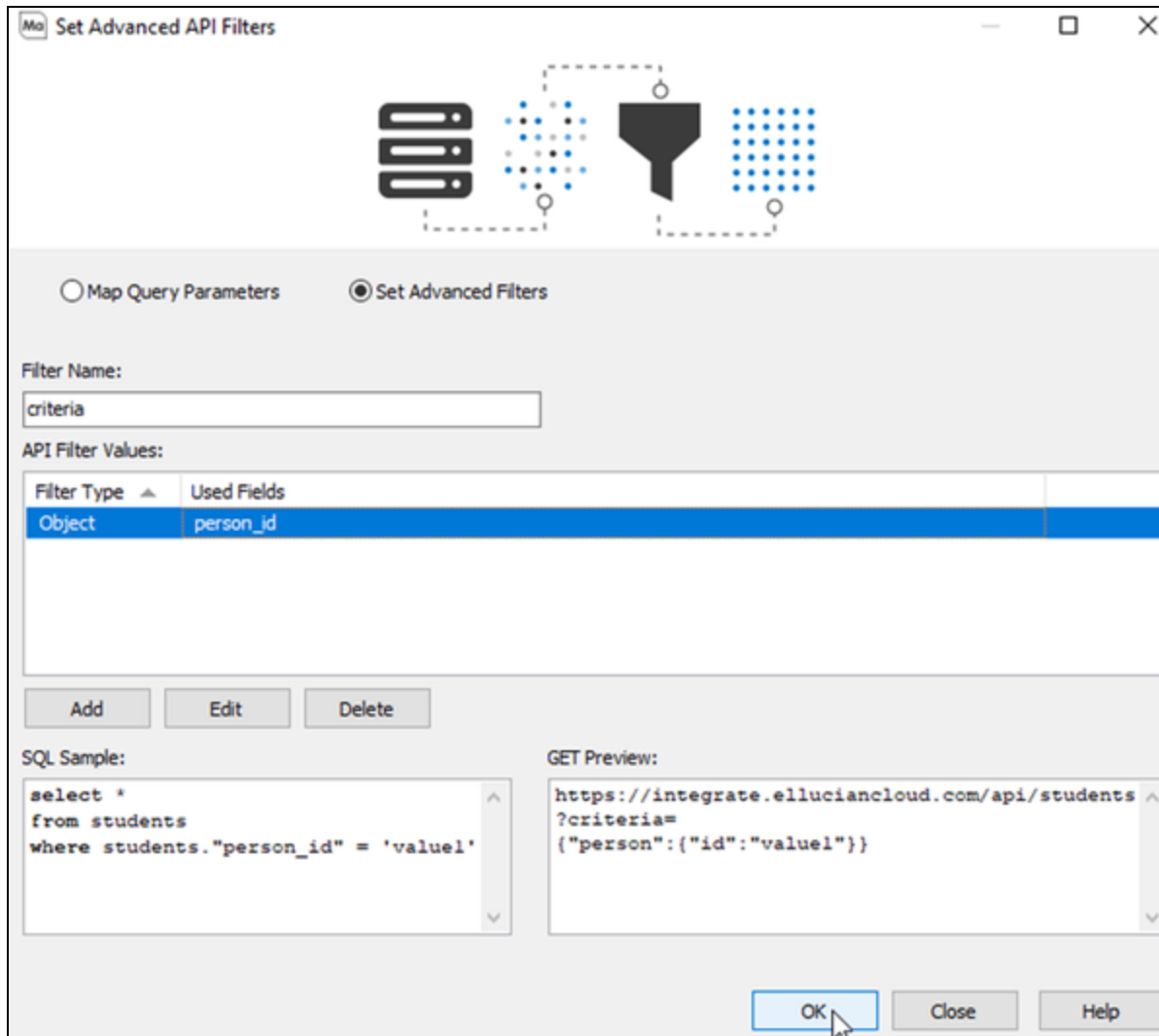
For JSON Type, select between **Array** or **Object**. We chose Object for this example because the **person_id** filter is formatted as a JSON object.

Input your **Object Key** and **Item Key** for the advanced filter in the corresponding fields.

In the **Source** drop-down list, select the schema field that you'd like to use as your source.

When finished, select **OK**.

You'll now see your JSON Type and schema fields under **API Filter Value**.



Select **OK** to save your filter to be used with future SQL queries.

MAPS 6.6 Release Notes

MAPS Service 6.6.0.2386 / MAPS Config 6.6.0.1326 / REST Connector XQP 1.0.1.38

MAPS

Enhancements

Area	Description	Issue number
Licensing	Client licenses are refreshed automatically each day.	MAPS-2548
REST Data Connections	REST Connectors can be exported and imported.	MAPS-2480
REST Data Connections	Added the Bearer Token authentication type to allow authentication of some REST sources using MAPS.	MAPS-2484
REST Data Connections	Added options to add, edit, and delete fields that are not present in the endpoint schema.	MAPS-2533
REST Data Connections	The REST Connector can now be used to connect to Ellucian Ethos data.	MAPS-2539
REST Data Connections	Added pagination options when configuring the REST Connector.	MAPS-2541
REST Data Connections	API server side filtering now has two options for configuration: MAPS Query Parameters, and API Filters in JSON format.	MAPS-2643
REST Data Connections	The Driver Properties details page for a REST data connection now includes the import and export status of the connector.	MAPS-2654
REST Data Connections	Added the ability to filter data from Endpoints using variables.	MAPS-2565
REST Data Connections	REST Connectors are now supported in FormFusion.	MAPS-2714

Resolved Issues

Area	Description	Issue Number
REST Data Connections	Custom Headers were being formatted incorrectly by the XQP, which prevented successful configuration of the REST Connector when using more than one custom header.	MAPS-2529
Security	Error messages in the MAPS log which contain supporting parameter information now conceal the values of all parameters that have names containing the text "password".	MAPS-2561

Getting Help

For information on using the software, please refer to the in-product Help, which contains detailed information on all aspects of the product.

If you are having problems with the installation or configuration, you can search our [support site](#), which includes a knowledge base of common issues. If you are unable to find the solution, submit a HelpDesk request with a detailed explanation of the problem you are experiencing.

Please do not hesitate to contact the Evisions HelpDesk if any questions or problems arise. We are here to help you and want to ensure your success.

If you find that areas of this documentation could benefit from additional detail or clarification, please let us know. We are constantly trying to improve the installation process to make it as easy as possible.