Informatica

Informatica® Cloud Data Integration

Google Sheets Connector

Informatica Cloud Data Integration Google Sheets Connector
June 2019

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Publication Date: 2020-03-09

Table of Contents

Preface

Use *Google Sheets Connector* to learn how to read from or write to Google Sheets by using Cloud Data Integration. Learn to create a connection, develop mappings, synchronization tasks, and run synchronization and mapping tasks in Cloud Data Integration.

Informatica Resources

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services Web Site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Data Integration editions and applications as well as information about other Informatica Cloud integration services.

Data Integration Communities

Use the Data Integration Community to discuss and resolve technical issues in Data Integration. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Data Integration Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

To find resources on using Cloud Application Integration (the Informatica Cloud Real Time service), access the community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-application-integration/content

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Data Integration Connector Documentation

You can access documentation for Data Integration Connectors at the Data Integration Community: https://network.informatica.com/cloud/index.htm

You can also download individual connector guides: https://network.informatica.com/docs/DOC-15333.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or online.

For online support, click Submit Support Request in the Data Integration application. You can also use Online Support to log a case. Online Support requires a login. You can request a login at https://network.informatica.com/welcome.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Google Sheets Connector

This chapter includes the following topics:

Google Sheets Connector Overview

You can use Google Sheets Connector to connect to Google Sheets from Data Integration. Use Google Sheets Connector to read data from and write data to Google Sheets.

You can use a Google Sheets object as a source and a target in synchronization tasks, mapping tasks, and mappings.

When you run a task or mapping, the Secure Agent uses the JAVA client libraries of the Google Sheets APIs to integrate with Google Sheets.

Google Sheets Connector Supported Objects and Task Operations

You can perform insert, update, and delete operations on a Google Sheets target. You cannot perform upsert operations on a Google Sheets target.

The following table lists the Google Sheets Connector object types that you can include in Data Integration tasks:

Task Type	Source	Target	Lookup	Data Preview
Synchronization	Yes	Yes	Yes	Yes
Mapping	Yes	Yes	Yes	Yes

The standard objects are the Sheet Names that are present in the spreadsheet. The Sheet Names are dynamic and fetched automatically during data preview.

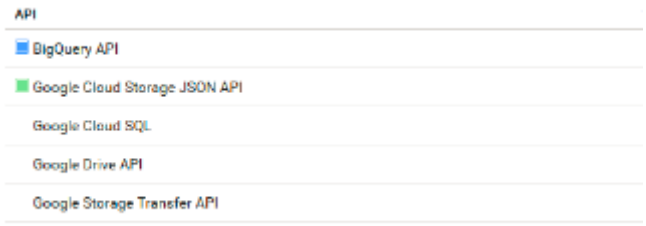
The following table lists the objects and task operations that Google Sheets Connector supports for Sheet Names object:

Read	Insert	Update	Upsert	Delete
Yes	Yes	Yes	NA	Yes

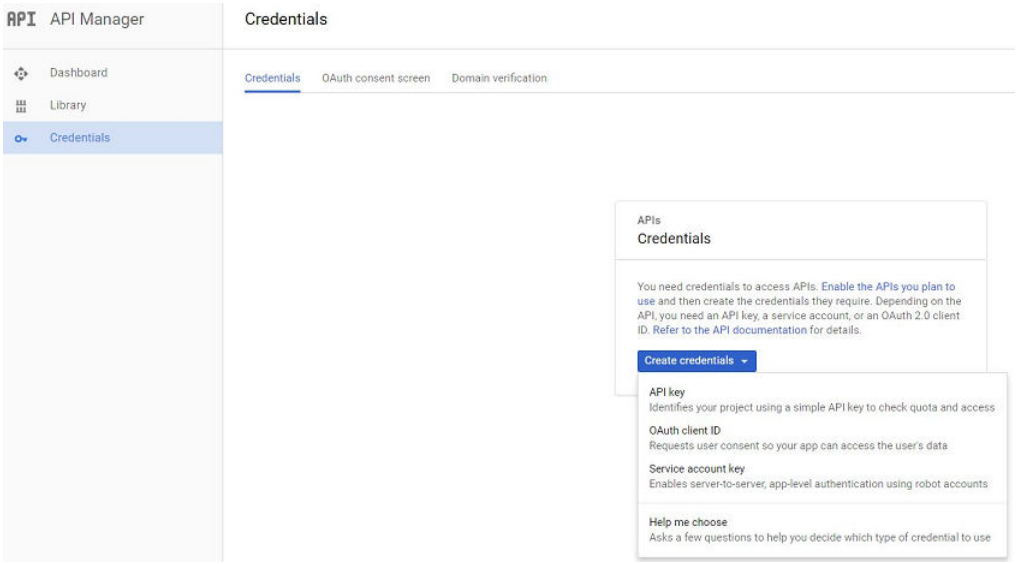
Administration of Google Sheets Connector

Before you use Google Sheets Connector, you must complete the following prerequisite tasks:

1. Create a Google account to access Google Sheets.
2. On the **Dashboards** page of the Google API Console, https://console.developers.google.com/, enable the **Google Sheets API** for your project. Google Sheets Connector uses the Google APIs to integrate with Google Sheets.
   The following image shows the **Dashboard** page where you can enable the APIs:



3. On the **Credentials** page of the Google API console, click on **Create Credentials** > **OAuth client ID**.
   The following image shows the **Credentials** page where you can create the credentials for your project:



4. Fill the form on the **OAuth consent screen** tab.

5. Click **Save**.
6. Select **Application type** as **Other**.
7. Enter the **Name**.
8. Click **Create**.
   The Client ID and Client Secret appears on the screen. Copy the Client ID and Client secret values.
9. In the OAuth 2.0 Client IDs section, download the JSON file corresponding to your OAuth client ID that contains the client_id, project_id, auth_uri, token_uri, auth_provider_x509_cert_url, client_secret, and redirect_uris values.
10. Generate the OAuthu 2.0 access tokens. You will need to enter these details when you create a Google Sheets connection in Data Integration.
    For more information on generating the OAuth 2.0 access tokens, click the following URL:
    https://developers.google.com/identity/protocols/OAuth2WebServer#obtainingaccesstokens

Generating OAuth 2.0 access tokens

Google Sheets Connector uses OAuth 2.0 protocol to connect to Google Sheets. After you generate the client ID and client secret for your project, you must generate the OAuth 2.0 access tokens.

Step 1. Generate the authorization code

You must generate the authorization code to gain access to the current site and to generate a valid refresh token.

Perform the following steps to generate the authorization code:

1. Enter the following URL in the Google chrome browser:

   ```
   https://accounts.google.com/o/oauth2/auth?
   access_type=offline&approval_prompt=auto&client_id=<client_id>&response_type=code&scope=h
   ttps://www.googleapis.com/auth/spreadsheets&redirect_uri=<redirect_uri>
   ```

   For example, `https://accounts.google.com/o/oauth2/auth?acess_type=offline&approval_prompt=auto&client_id=1234-abc54dfa6.apps.googleusercontent.com&response_type=code&scope=https://www.googleapis.com/auth/spreadsheets&redirect_uri=http://localhost`
2. Choose the Google Account for which you want to approve the access request.
3.
4. Click **Allow** to grant the permission to see, edit, create, and delete your spreadsheets in Google Drive. Click **Allow** to confirm.

   The redirect URL page includes the authorization code as a query string in the following format:

   ```
   https://<redirect_url>/?code=<authcode>&scope=https://www.googleapis.com/auth/
   spreadsheets
   ```

   The following image shows the authorization code as a query string in the URI:

5. Copy the authorization code displayed in the URL.

Step 2. Generate the refresh token and access token

Google Sheets Connector uses a authorization code to generate an access token and a refresh token. Google Sheets Connector uses an access token to authorize and access Google Sheets. The access token expires

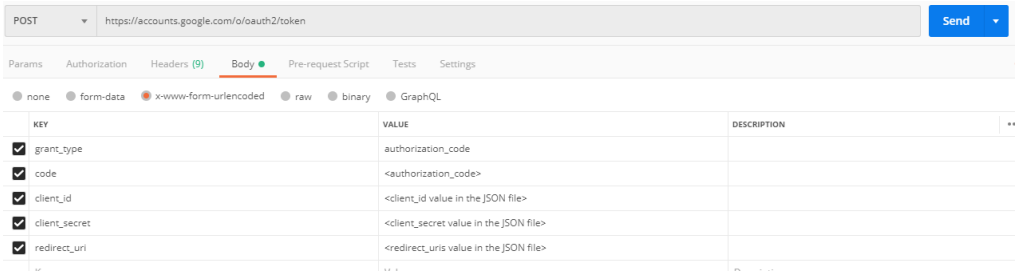
after a period of time. When the access token expires, you can use the refresh token to generate a new access token.

Perform the following steps to generate the refresh token:

1. Enter the following URL in the Postman application: `https://accounts.google.com/o/oauth2/token`
2. Select the **POST** method.
3. On the **Body** tab, select the **x-www-form-urlencoded** body type.
4. On the **Body** tab, add the following key-value pairs:

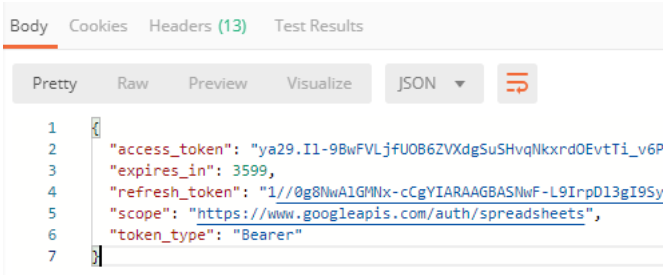
Key	Value
grant_type	Enter **authorization_code**.
code	Enter the authorization code that you generated in step 1.
client_id	The client_id value present in the JSON file.
client_secret	The client_secret value present in the JSON file.
redirect_uri	The redirect_uri value present in the JSON file. For example, `http://localhost`.

The following image shows the **Body** tab where you can add the key-value pairs to generate the refresh token:



5. Click **Send**.

   The refresh token and access token is generated in the **Response** tab. The following image shows the refresh_token and access_token in the **Response** tab:



   Note: Refresh token is generated only when you set the **access_type** parameter to **offline** in the initial request to obtain the authorization code. The **expires_in** field specifies the remaining time period of the access token in seconds.

6. Save the refresh token and access token in a secured location.

Step 3. Refreshing an access token

When the access token expires, you can use the refresh token to generate a new access token.

Perform the following steps to generate the access token:

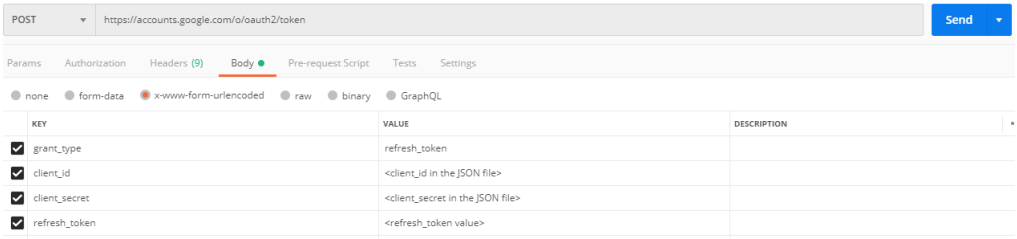
1. Enter the following URL in the Postman application:

   `https://accounts.google.com/o/oauth2/token`

2. Select the **POST** method.
3. On the **Body** tab, select the **x-www-form-urlencoded** body type.
4. On the **Body** tab, add the following key-value pairs:

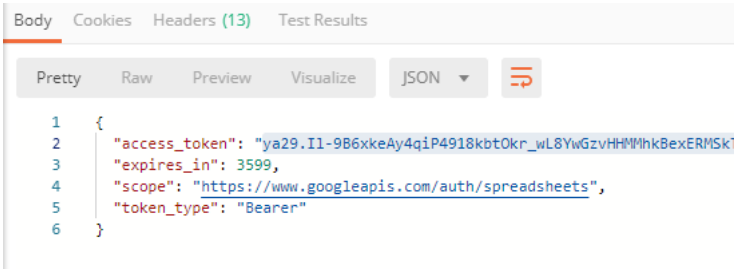
Key	Value
grant_type	Enter **authorization_code**.
client_id	The client_id value present in the JSON file.
client_secret	The client_secret value present in the JSON file.
refresh_token	The refresh token that you generated in step 2.

The following image shows the **Body** tab where you can add the key-value pairs to generate a new access token:



5. Click **Send**.

   The new access token is generated in the **Response** tab. The following image shows the new access token in the **Response** tab:



CHAPTER 2

Google Sheets Connections

This chapter includes the following topics:

Google Sheets Connections Overview

Create an Google Sheets connection to access Google Sheets data from Data Integration. You can create a connection on the Connections page or when you create a task. After you create a connection, it becomes available to all users who have access to the organization.

You can use Google Sheets connections in synchronization tasks, mapping tasks, and mappings.

Google Sheets connection properties

When you create a Google Sheets connection, you must configure the connection properties.

The following table describes the Google Sheets connection properties:

Property	Description
Runtime Environment	Name of the runtime environment where you want to run the tasks.
ClientId	Required. The Client ID from Google Developer Console.
ClientSecret	Required. The Client Secret from Google Developer Console.
RefreshTokenForSheet	Required. The Refresh Token received after exchanging authorization code for Google Sheets.
RefreshTokenForDrive	Optional. The Refresh Token received after exchanging authorization code for Google Drive. This option is required when you enter the spreadsheet name in the **SpreadSheetName** field.
SpreadSheetName	Name of the spreadsheet in Google Sheets.
SpreadSheetId	ID of the spreadsheet in Google Sheets.

Property	Description
InitialColumnRange	Initial column range of the spreadsheet in Google Sheets.
FinalColumnRange	Final column range of the spreadsheet in Google Sheets.
HeaderPresent	Select this option to indicate that the sheet contains a header. If you select this option and the sheet does not contain a header, the first row is treated as the header.
CreateNewSpreadsheet	Select this option to create a new spreadsheet in Google Sheets. The Google Sheets Connector creates an empty spreadsheet with the name that you specified in the **SpreadSheetName** field. Once you test the connection, disable this option. Otherwise, the Google Sheets Connector will create a new spreadsheet with the same name everytime

CHAPTER 3

Synchronization Tasks with Google Sheets Connector

This chapter includes the following topics:

Google Sheets sources in synchronization tasks

You can use Google sheets object as a source in a synchronization task.

You can configure the Google Sheets source properties on the **Source** page of the Synchronization Task wizard.

The following table describes the Google Sheets source properties:

Property	Description
Connection	Name of the active Google Sheets source connection.
Source Type	Type of the Google Sheets source objects available. You can read data from a single Google Sheets source object. You cannot read data from multiple objects or parameterize the object.
Source Object	Name of the Google Sheets source object.
Display technical names instead of labels	This property is not applicable for Google Sheets Connector because both the technical names and labels are the same for Google.
Display source fields in alphabetical order	Displays source fields in alphabetical order. By default, fields appear in the order returned by the source system.

Google Sheets Targets in the Synchronization Task

You can use Google Sheets object as a target in a synchronization task.

The following table describes the Google Sheets target properties:

Property	Description
Connection	Name of the active Google Sheets target connection that is associated with a dataset.
Target Object	You can select an existing object from the list or create a target at run time using the Create Target option.
Child Object	This property is not applicable for Google Sheets Connector.
Display technical names instead of labels	This property is not applicable for Google Sheets Connector because both the technical names and labels are the same for Google.
Display target fields in alphabetical order	Displays target fields in alphabetical order. By default, fields appear in the order returned by the target system.

Rules and guidelines for Google Sheets targets

Consider the following rules and guidelines when you write data to a Google Sheets target:

- When you perform an insert operation on a Google Sheets target, you must configure a data filter for the following Google Sheets source fields:

Field Name	Operator	Data Type	Example
range	=	String	range = A1:B2
majorDimension	=	String	majorDimension = ROWS
valueInputOption	=	String	valueInputOption = USER_ENTERED

- When you perform a delete operation on a Google Sheets target, you must configure a data filter for the **range** field in the Google Sheets source.

CHAPTER 4

Mappings and Mapping Tasks with Google Sheets Connector

This chapter includes the following topics:

Google Sheets Sources in Mappings and Mapping Tasks

To read data from Google Sheets, configure a Google Sheets object as the Source transformation in a mapping or mapping task.

Specify the name and description of the Google Sheets source. Configure the source, query options, and advanced properties for the source object.

The following table describes the source properties that you can configure for a Google Sheets source:

Property	Description
Connection	Name of the active Google Sheets source connection.
Source Type	Type of the Google Sheets source objects available. You can read data from a single Google Sheets source object or parameterize the object.
Object	Name of the Google Sheets source object based on the source type selected.
Filter	Configure a simple filter or an advanced filter to remove rows at the source. You can improve efficiency by filtering early in the data flow. A simple filter includes a field name, operator, and value. Use an advanced filter to define a more complex filter condition, which can include multiple conditions using the AND or OR logical operators.

You can set the tracing level in the advanced properties session to determine the amount of details that logs contain.

The following table describes the tracing levels that you can configure:

Property	Description
Terse	The Secure Agent logs initialization information, error messages, and notification of rejected data.
Normal	The Secure Agent logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes session results, but not at the level of individual rows.
Verbose Initialization	In addition to normal tracing, the Secure Agent logs additional initialization details, names of index and data files used, and detailed transformation statistics.
Verbose Data	In addition to verbose initialization tracing, the Secure Agent logs each row that passes into the mapping. Also notes where the Secure Agent truncates string data to fit the precision of a column and provides detailed transformation statistics. When you configure the tracing level to verbose data, the Secure Agent writes row data for all rows in a block when it processes a transformation.

Google Sheets Targets in Mappings and Mapping Tasks

To write data to a Google Sheets target, configure a Google Sheets object as the Target transformation in a mapping or mapping task.

Specify the name and description of Google Sheets target. Configure the target and advanced properties for the target object.

The following table describes the target properties that you can configure for a Google Sheets target:

Property	Description
Connection	Name of the active Google Sheets connection that is associated with a dataset.
Target Type	Type of the Google Sheets target objects available. You can write data to a single Google Sheets target object. You cannot write data to multiple objects or parameterize the object.
Object	Name of the Google Sheets target object based on the target type selected.
Operation	You can select one the following operations: - Insert - Update - Upsert - Delete - Data Driven

The following table describes the advanced properties that you can configure for a Google Sheets target:

Property	Description
Success File Directory	Not applicable for Google Sheets Connector.
Error File Directory	Not applicable for Google Sheets Connector.
Forward Rejected Rows	Not applicable for Google Sheets Connector.

Rules and guidelines for Google Sheets targets

Consider the following rules and guidelines when you write data to a Google Sheets target:

- When you perform an insert operation on a Google Sheets target, you must configure a data filter for the following Google Sheets source fields:

Field Name	Operator	Data Type	Example
range	=	String	range = A1:B2
majorDimension	=	String	majorDimension = ROWS
valueInputOption	=	String	valueInputOption = USER_ENTERED

- When you perform a delete operation on a Google Sheets target, you must configure a data filter for the **range** field in the Google Sheets source.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

Data Integration uses the following data types in mappings, synchronization tasks, and mapping tasks with Google Sheets:

Google Sheets native data types

Google Sheets data types appear in the **Fields** tab for Source and Target transformations when you choose to edit metadata for the fields.

Transformation data types

Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When Data Integration writes to a target, it converts the transformation data types to the comparable native data types..

Google Sheets and Transformation Data Types

The following table describes the data types that Data Integration supports for Google Sheets sources and targets:

Google Sheets Data Type	Transformation Data Type	Range and Description for the Transformation Data Type
STRING	String	1 to 104,857,600 characters

INDEX