Working with Stratum Cloud Import API

Stratum.Viewer 7

Getting Started

- Access to Stratum Cloud Import API
- Introduction to Stratum Cloud Import API

Format & Parameters

- Required Format Details For API
- URL GET Method Parameters & Examples
- URL POST Method Parameters & Examples

Additional Information

Advanced Concepts

- Email Notifications & Processing Logs
- Processing Details For API-Initiated Data Imports

Definitions

- Data Steward
- Enhanced Category: Data Steward Or User Controlled
- Target Category For Data Import

Getting Started

Access to Stratum Cloud Import API

The Stratum Cloud Import API can be used with Stratum.Viewer implementations that have Data Import functionality set up for them. Those implementations must meet specific requirements related to import functionality, which includes having a Windows SQL Server implementation and Stratum Analyst Hub license. Additionally, a Stratum Cloud Import license is required in those implementations in order to access API features. Stratum.Viewer administrators who are designated as Data Stewards can use the API.

Note: See topics about Data Import in Viewer help for more details about Data Import requirements and features.

Copyright 2024 Silvon Software, Inc. ***All rights reserved.*** *(Last Revised: October 2024)*

Introduction to Stratum Cloud Import API

The API provides another option for handling data imports that bring user-curated data directly into your implementations Stratum Data hub. Your imported data is available immediately in Viewer to build out Stratum views, charts, and dashboards. The other option is to use the Data Import interface within Stratum.Viewer, which has windows specific to setting up and processing imports.

The API can be used to process imports where the source of data is a local SQL Server database table or Azure cloud sources such as an Azure Blob Container, an Azure File Share, or an Azure SQL Database. Create, process, and check the status of those imports interactively from Viewer or by using the Stratum Cloud API's URL calls.

- Like Viewer, the API offers flexible approaches and helpful efficiencies for imports. The API can automatically map the selected source data to applicable targets (dimensions and measures) in your data hub. Alternatively, you can tell the API to use a Data Import template from Viewer for mapping.
- The Stratum Cloud API also can be used by an app, website, or services – for example with an Azure portal's Azure Data Factory. In that scenario, the API could be called upon to import data from an Azure Data Pipeline. Features specific to your implementation, in this case your Azure Data Factory, would be used either to run the API-driven import right away or to schedule it for processing later according to the event criteria within your app.

***Note:** Existing knowledge and experience with API's and Azure are required when working with the features noted above. For example, you'll need to use your knowledge of your Azure implementation and API calls that use the URL POST method when using the Stratum Cloud Import API with an app, website or services, such as Azure Data Factory.

Format & Parameters For API Calls

Required Format Details For API

The structure of API calls for Stratum Cloud Import API begins with the URL for your Stratum.Viewer implementation followed by a forward slash then the text api/dataimport/ as shown in the example below. See the bold text in the following example of an API call using the URL GET method.

https://abcstratum.silvon.com/api/dataimport/process/?SourceType=1&SourceName=PointofSalesData.CSV

Minimum required details that you'll need to supply when setting up the API call for an import are:

1. **SourceType** – The type of source where the import data file or table resides. This is either Azure File Share, Azure Blob, Azure SQL Table, and On-Premise SQL Table (also known as a local SQL Server table). In the example above, source type is 1 which indicates a type of Azure File Share.
2. **SourceName** – The name of the file or SQL table that contains your import data. In the above example, the specified name indicates it's a CSV file.
3. If you're processing an existing data import, supply the ID of the import using a **DataImportId** parameter.
4. If you're using a Data Import Template for data mapping, supply the ID of the template using a **TemplateId** parameter. The ID can be looked up in Stratum.Viewer by opening the template from the Data Import Templates list window.

Additional parts of the calls vary depending on whether you are using the URL GET method or URL POST method. For example, the GET method has "**process**", "**create**", or "**status**" text in the API call that tells the API what to do such as process an import or check the status of a data import that's already been submitted for processing. The GET method also can be used with the ***DataImportId=*** parameter to check the status of an import that was already submitted for processing. The example at the start of this topic uses "process" text followed by the source type and name parameters – in that case, the import will be created and processed right after it's created.

See these topics for a complete list of parameters for either type of Stratum Cloud Import API calls.

- URL GET Method Parameters & Examples
- URL POST Method Parameters & Examples

URL GET Method Parameters & Examples

Here are examples of Stratum Cloud Import API calls that use the URL GET method. The table at the end of this topic shows the available parameters for the calls. Remember, these always begin with the URL for your Stratum.Viewer implementation followed by a forward slash then the text api/dataimport/. After that, there is process, create, or status text and then /? text that precedes parameters for telling the API what to import or what status to check on for existing in-process imports. Use the "&" character to separate multiple parameters.

After you copy the call into a browser tab, simple status information about the completed call displays in the browser window.

Example 1: Create & Process A Data Import

Create and process a data import using the source file available from "Azure File Share" where the file name is "PointofSalesData.CSV".

https://abcstratum.silvon.com/api/dataimport/**process**/?SourceType=1&SourceName=PointofSalesData.CSV

Example 2: Only Create A Data Import

Only create a new data import (don't process it) using the source file available from "Azure File Share" where the file name is "Budget Units.CSV".

https:/abcstratum.silvon.com/api/dataimport/**create**/?SourceType=1&SourceName=Budget%20Units.CSV

Example 3: Process A Previously Created Data Import

Process an existing data import that hasn't been processed yet, by passing the Data Import ID.

https://abcstratum.silvon.com/api/dataimport/**process**/?DataImportId=34

Example 4: Check The Processing Status Of A Data Import

Check the processing status of a specific Data Import by passing the Data Import ID.

https://abcstratum.silvon.com/api/dataimport/**status**/?DataImportId=34

You can also check the status by looking up the data import in the Data Import list window of Stratum.Viewer.

Example 5: Use A Template For Import Mapping

Create and process a data import using the source file available from "Azure Blob" where the file name is "BudgetNetChangeData.txt". Also, use the import template with ID "3" to handle the import's data mapping.

https://abcstratum.silvon.com/api/dataimport/**process**/?SourceType=2&SourceName=BudgetNetChangeData.CSV&TemplateId=3

Summary of URL GET Method Information

Main Parts or Parameters	**Purpose**
Environment URL followed by forward slash then the text api/dataimport/	Specifies what environment to use and to use the Stratum Cloud Import API. Example: https://abcstratum.silvon.com/api/dataimport/
create	Tells the API to create a data import, but not process it.
process	Tells the API to either: 1. Create then process a data import. The call must include the /SourceType= and SourceName= parameters. 2. Process an already-created data import. The call must include the /DataImportId= parameter.

Copyright 2024 Silvon Software, Inc. ***All rights reserved.*** *(Last Revised: October 2024)*

status	Tells the API to check the status of a data import that's been submitted for processing. The browser tab will display the status. That call also must include the /DataImportId= parameter.
SourceType=	Specifies the type of source where the data file or table resides. Valid values for SourceType= are: 1 – Azure File Share 2 – Azure Blob 3 – On-premise SQL Server Table 4 – Azure SQL Table
SourceName=	The name of the file or SQL Table. Files can be Excel, CSV, or Text.
DataImportId=	The ID of the existing data import that you want to process or check its status.
TemplateId=	The ID of the Data Import Template that you want to use for mapping the data import. The ID can be looked up in Stratum.Viewer by opening the template from the Data Import Templates list window.

URL POST Method Parameters & Examples

Here are examples of Stratum Cloud Import API calls that use the URL POST method – for use with an app, website, or services. The table at the end of this topic shows the different parameters for the calls. Remember, these calls always consist of the URL for your Stratum.Viewer implementation followed by a forward slash then text api/dataimport/. The JSON payload sends related parameters for the API that you specify in the BODY of the call.

Example 1: Create & Process A Data Import

Create and process a data import using the source file available from "Azure File Share" where the file name is "PointofSalesData.CSV".

https://abcstratum.silvon.com/api/dataimport/

The JSON payload sends these parameters within the BODY of the call.

{"SourceType":1,"SourceName":"PointofSalesData.CSV"}

Example 2: Use A Template For Import Mapping

Create and process a data import using the source file available from "Azure Blob" where the file name is "BudgetNetChangeData.txt". Also, use the import template with ID "3" to handle the import's data mapping.

https://abcstratum.silvon.com/api/dataimport/

The JSON payload sends these parameters within the BODY of the call.

{"SourceType":2,"SourceName":"BudgetNetChangeData.txt","TemplateId":3}

Example 3: Use A Template For Import Mapping & Give The Import A Description

Create and process a data import using the source file available from "Azure Blob" where the file name is "Budget Adjusted Data.CSV". Use the import template with ID "15" to handle the import's data mapping. Also, give the import a description of "Budget Refresh Data".

https://abcstratum.silvon.com/api/dataimport/

The JSON payload sends these parameters within the BODY of the call.

{"SourceType":2,"SourceName":"Budget Adjusted Data.CSV","TemplateId":15,"description":"Budget Refresh Data"}

Example 4: Control Mapping & Other Optional Parameters

The URL POST method has options to manually control mapping for a data import. This example uses parameters to designate a description for the import, set import configuration options like those in the Import Configuration window that's available when doing imports from Stratum.Viewer (including a choice to delete all category data before importing data from this import), and telling the API which dimensions and measures to use when mapping import data to target data in Stratum. The example will create and process a data import using the source file available from "Azure File Share" where the file name is "Budget Adjusted Data.CSV". Also, it will give the import a description of "Budget Refresh Data" and follow the import configuration and mapping specification defined by other parameters in the BODY of the call.

https://abcstratum.silvon.com/api/dataimport/

The JSON payload sends these parameters within the BODY of the call.

```
{
"sourceType": 1,
"sourceName": "Budget Adjusted Data.csv",
"description": "Budget Refresh Data",
"targetCategory": "Budget",
"deletedCategoryDataBeforeImport": 1,
"transactionDateFormat": 0,
"delimiter": 0,
"fieldQuote": 0,
"allowSchemaDrift": 0,
"dataMapping": [{
   "columnCaption": "Product",
   "targetType": "0",
   "targetName": "Product"
  },{
   "columnCaption": "Transaction Date",
   "targetType": "2",
   "targetName": ""
  },{
   "columnCaption": "Budget Amt",
   "targetType": "1",
   "targetName": "Budget Amount"
  },{
   "columnCaption": "Budget Unit",
   "targetType": "1",
   "targetName": "Budget Unit"
  }]
}
```

Summary of URL POST Method Information

Main Parts or Parameters	**Purpose**
Environment URL followed by forward slash then the text api/dataimport/	Specifies what environment to use and to use the Stratum Cloud Import API. Example: https://abcstratum.silvon.com/api/dataimport/
"SourceType":	Specifies the type of source where the data file or table resides. Valid values for SourceType= are: 1 – Azure File Share 2 – Azure Blob 3 – On-premise SQL Server Table 4 – Azure SQL Table

Copyright 2024 Silvon Software, Inc. ***All rights reserved.*** *(Last Revised: October 2024)*

"SourceName":	The name of the file or SQL Table. Files can be Excel, CSV, or Text.
"DataImportId":	The ID of the existing data import that you want to process.
"TemplateId":	The ID of the Data Import Template that you want to use for mapping the data import. The ID can be looked up in Stratum.Viewer by opening the template from the Data Import Templates list window.
Optional Parameters	**Purpose** (Use to specify a description or manually configure & map an import)
"description":	Optionally specify a description to associate with the import. Example: "description": "Budget Refresh Data"
"targetCategory": *"deletedCategoryDataBeforeImport":* *"transactionDateFormat":* *"delimiter":* *"fieldQuote":* *"allowSchemaDrift":*	Optionally use to manually specify import configuration properties. These properties are similar to what you would see if using the Import Configuration window in Stratum.Viewer. They include the Target category to import to, whether or not to delete all target category data before the import, the format of transaction dates, the delimiter and field quote type, and whether or not to allow schema drift. See the Import Configuration window help topic in Viewer for more details. **Example:** "targetCategory": "Budget", "deletedCategoryDataBeforeImport": 1, "transactionDateFormat": 0, "delimiter": 0, "fieldQuote": 0, "allowSchemaDrift": 0 **Options:** "deletedCategoryDataBeforeImport": Use 0 for No, 1 for Yes "transactionDateFormat": Use 0 for yyyy-mm-dd (2012-03-01), 1 for m/d/yy (3/1/12), 2 for mm/dd/yy (03/01/12), 3 for m/d/yyyy (3/1/2012), 4 for d-mmm-yy (1-Mar-12), 5 for dd-mmm-yy (01-Mar-12), 6 for mmmm d, yyyy (March 1, 2012), 7 for d-mmm-yyyy (1-Mar-2012) "delimiter": Use 0 for comma, 1 for colon, 2 for equal sign, 3 for semicolon, 4 for space, 5 for tab, 6 for pipeline "fieldQuote": Use 0 for double quotes, 1 for single quotes "allowSchemaDrift": Use 0 for No, 1 for Yes
"dataMapping": *"columnCaption":* *"targetType":* *"targetName"*	Optionally use this set of parameters to manually map imported data to data in the Stratum Data hub. These properties are similar to what you would see if using the Data Mapping window in Stratum.Viewer. They indicate heading captions for columns of data in the source file or table, the type of data (transaction date, dimension, or measure), and the specific date format, Stratum dimension, or Stratum measure to map data to during the import. See the Data Mapping – Data Import help topic in Viewer for more details.

	Example:
	
	"dataMapping": [{
	"columnCaption": "Product",
	"targetType": "0",
	"targetName": "Product"
	},{
	"columnCaption": "Transaction Date",
	"targetType": "2",
	"targetName": ""
	},{
	"columnCaption": "Budget Amt",
	"targetType": "1",
	"targetName": "Budget Amount"
	},{
	"columnCaption": "Budget Unit",
	"targetType": "1",
	"targetName": "Budget Unit"
	}]
	
	Options:
	
	"columnCaption": Enter the name of what shows in the heading for the column being mapped.
	
	"targetType": Use 0 for dimensions, 1 for measures, 2 for transaction dates.
	
	"targetName": For dimension and measure columns, enter the name of the target Stratum dimension or measure where data needs to be imported. For transaction date columns, leave blank in the format of double quotes “”.

Additional Information

Email Notifications & Processing Logs For Stratum Cloud Import API

Administrative settings in Stratum.Viewer can be used to send email notifications to administrators about data import processes done by the API or through the Stratum.Viewer interface. Emails can be sent when imports complete successfully or fail. The settings also can be used to attach processing logs to those emails.* See the Data Import section of Stratum.Viewer help for more details.

***Note:** In hosted Stratum environments, processing logs will only be sent to recipients with emails from @silvoncloud.com or @silvon.com.

Processing Details For Stratum Cloud Import API

Processing details about data imports can be emailed to specified recipients. Administrators can also view import and processing details from these other places:

1. Open the Data Import list window in Stratum.Viewer. Imports submitted for processing are listed there with basic details including the processing status such as Completed. To lookup more details, select an import then click the “See Processing Details For A Data Import” icon in the window toolbar.
2. Look for the log files saved in Stratum.Connector’s Log sub-folder. Administrators with the authority to access the Log folder can open log files to see their details. Log files for API-driven data imports begin with the prefix text “Data Import API…”.

Copyright 2024 Silvon Software, Inc. ***All rights reserved.*** *(Last Revised: October 2024)*

Definitions

Data Steward

Security Administrators can be granted Data Steward rights, which expands their capabilities within Stratum Data Import and data management functionality.

- Data Stewards have full access to Data Import features — they can import data* from all cloud and local sources of data that have been defined for their environment.
- Also, Data Stewards can use either the Stratum Cloud API or the Data Import windows in Stratum Viewer to initiate their data imports. Both tools are available to them. Stratum users who aren't Data Stewards can import data only from local sources and only from the Data Import windows in Viewer.

***Note:** Data can only be imported into measures from categories that have been assigned the "Enhanced" Stratum data model architecture, which is handled via Category maintenance.

Enhanced Category: Data Steward Or User Controlled

Categories with an 'Enhanced' data architecture designation can be used with Data Import. These categories contain data that is controlled by the user community, either general users or Data Steward administrators. The type of user who can import into the category is determined by User vs. Data Steward designations controlled from the Category window. For these categories, data can be imported from corporate business systems as well as external data sources such as demographics, housing trends, or other unique data that will complement your Stratum data.

Target Category For Data Import

Data Imports have data mapping properties that define where to allocate the imported data to when a Data Import gets processed. Data can be imported into measures that belong to the category that's been selected as the Target Category for the Data Import.