Filip Skurniak
filip@skurniak.pl

TABLEAU & GSHEETS INTEGRATION ANALYSIS

This report contains analysis of problem of incorporating google sheets as data sources into Tableau Server and presents Tableau SDK as an alternative to live connections.

TABLEAU & GSHEET INTEGRATION ANALYSIS

Table of content

TABLEAU & GSHEET INTEGRATION ANALYSIS

Problem overview

Tableau has created a possibility for their customers to extend standard capabilities of Tableau Server by supporting custom data connectors. It was a response to growing need for creating reports from different data sources. As a consequence Tableau developer community created great amount of data connectors [1]- plugins which enabled connecting Tableau reports with different data sources. One of which - Google Sheet Connector - will be mentioned in this report.

Unfortunately, not all features of standard data sources were taken into consideration while creating web connectors as plugins for Tableau Server. The weakest point for the time being is lack of support for live connection to data source. User is unable to refresh report data even if the data provided by Google Sheet has changed. Live connection means that any changes in the data source are automatically visible to end user. In contrast to live connection it is possible to create local snapshot of a data source which means creating a specific file called data extract.

Google Sheets Tableau Connector

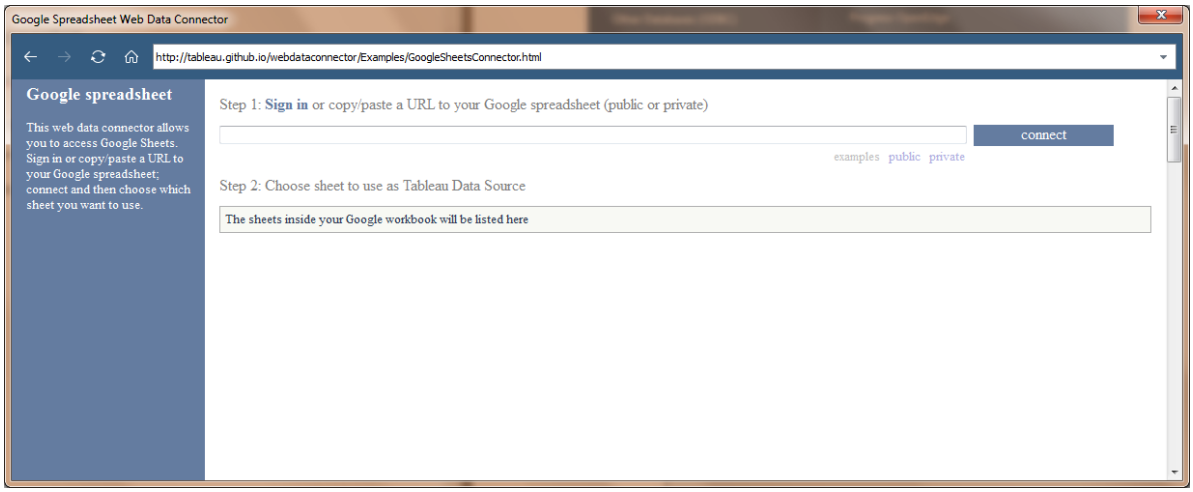
Google Sheet connector is part of open source project on GitHub. It consists of two parts: sources[2] and implementation[3].

SHORT TUTORIAL

To use Google Sheet Connector in Tableau Server or Tableau Desktop follow this short tutorial

1. Click 'Connect to Data' in main report creator and 'More Servers'
2. Choose 'Web Data Connector'
3. Enter your web data connector URL in the textbox

[1] https://github.com/tableau/webdataconnector
[2] https://github.com/tableau/webdataconnector/tree/gh-pages/Examples
[3] http://tableau.github.io/webdataconnector/Examples/GoogleSheetsConnector.html



4. Provide link to your spreadsheet and connect (note that spreadsheet can be private or public)
   a. Private spreadsheets needs further authentication[4] which means you will be prompted to enter user and password of the user that has rights to given spreadsheet
   b. Public spreadsheet also need authentication. Only spreadsheets published[5] (snapshotted) by user do not need authentication
5. Authentication is standard for Google services. As a result of properly submitted login form access token is generated and used to query data
6. Choose spreadsheet you want to work on

TABLEAU SERVER ISSUE

According to problem overview it is not possible to setup a live connection to google spreadsheet using existing Google Spreadsheet Connector. As a result of experiments it is assumed that access token gained in process of authentication is short lived (typically only valid for an hour).

What it means is that the token is only valid while publishing the workbook to server for the first time. When the report refresh is performed later the token is expired and needs to be refresh. Tableau Server probably does not have mechanism to refresh access token

[4] https://developers.google.com/identity/protocols/OAuth2
[5] https://support.google.com/docs/answer/183965?hl=en

and uses old expired token. As a result data is not refreshed. According to OAuth2.0 protocol refreshing access token would mean prompting user for their credentials once again.

Possible workarounds

ODBC DRIVER

Tutorial was prepared for creating a report with ODBC driver as a result of discussion on community support site[6]. You can also find pdf tutorial[7].

The solution is based on commercial solution of ODBC connector[8] and generating ClientID and SecretID for google services.

TABLEAU SDK - EXTRACT MANIPULATION

Tableau offers SDK which can be used to programmatically change extracts generated by Tableau Server or Desktop. This means that there is a way to create data source compatible with Tableau products from every data source that can be queried. Tableau SDK supports Python versions 2.6 and 2.7. (Python 3.3 is not supported), Java and C/C++.

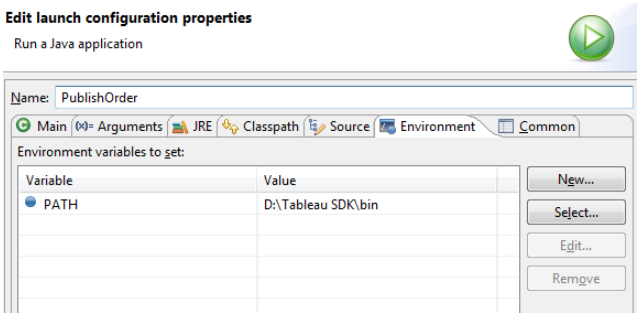
Extract API was tested with Java language but all methods and functions are implemented also in listed programming languages.

EXTRACT API

To setup the SDK with Eclipse and Java the documentation[9] can be followed. The tricky part is to remember to setup 'Run Configuration for project'. In Eclipse it can be found under Properties → Run/Debug Settings → Edit Configuration → Environment.

---

[6] https://community.tableau.com/message/310437
[7] https://community.tableau.com/servlet/JiveServlet/download/422466-74784/Tableau%20to%20Google%20Docs%20Connection.pdf
[8] http://www.cdata.com/drivers/gsheets/odbc
[9] http://onlinehelp.tableau.com/current/api/sdk/en-us/help.htm#SDK/tableau_sdk_using_java_eclipse.htm%3FTocPath%3D____5



According to documentation the variable PATH must be set up.

Using Extract API of steps listed below.

To start using Extract API can be initialized by calling *`initialize`* method. Initialization provides logging in the *`TableauSDKExtract.log`* file.

Extract object is initialized by calling Extract class constructor with file path to Tableau data extract. Extension '.tde' of the file must be provided.

In order to manipulate data in extract *`TableDefinition`* must be known. *`TableDefinition`* represents the schema for a table in a Tableau data extract. The schema consists of a collection of column definitions, or more specifically name/type pairs. There are two approaches regarding *`TableDefinitions`* in extracts. It can be acquired from the extract or created.

Creating new *`TableDefinition`* is similar to creating table in SQL. All columns and datatypes have to be provided before any data can be stored in extract. This stage can be considered as creating schema of table. Method *`addColumn`* of *`TableDefinition`* class is used to add new column to a table definition. It is possible to add column with collation by calling *`addColumnWithCollation`* method.

Next step is to create *`Table`* object with *`TableDefinition`* created in the previous step. New Table is created by calling *`addTable`* method where first parameter is the name of table and second a *`TableDefinition`*. First parameter of this function have to be 'Extract'. It is a requirement stated in the documentation. As a result there can only be one table called 'Extract' in every extract created by Extract API.

To insert data into *`Table`* first the *`Row`* to inset has to be created. *`Row`* can be added to table using *`insert`* method.

Summary

Tableau offers integration with many data sources but openness to community connectors and other tools is what distinguish their products from other visualization tools. Tableau SDK is powerful engine for manipulating data. Applications of Tableau SDK are much wider than these described in this report. It can be use in ETL processing, integrating data from many sources or simply as a part of connector to not supported data source.