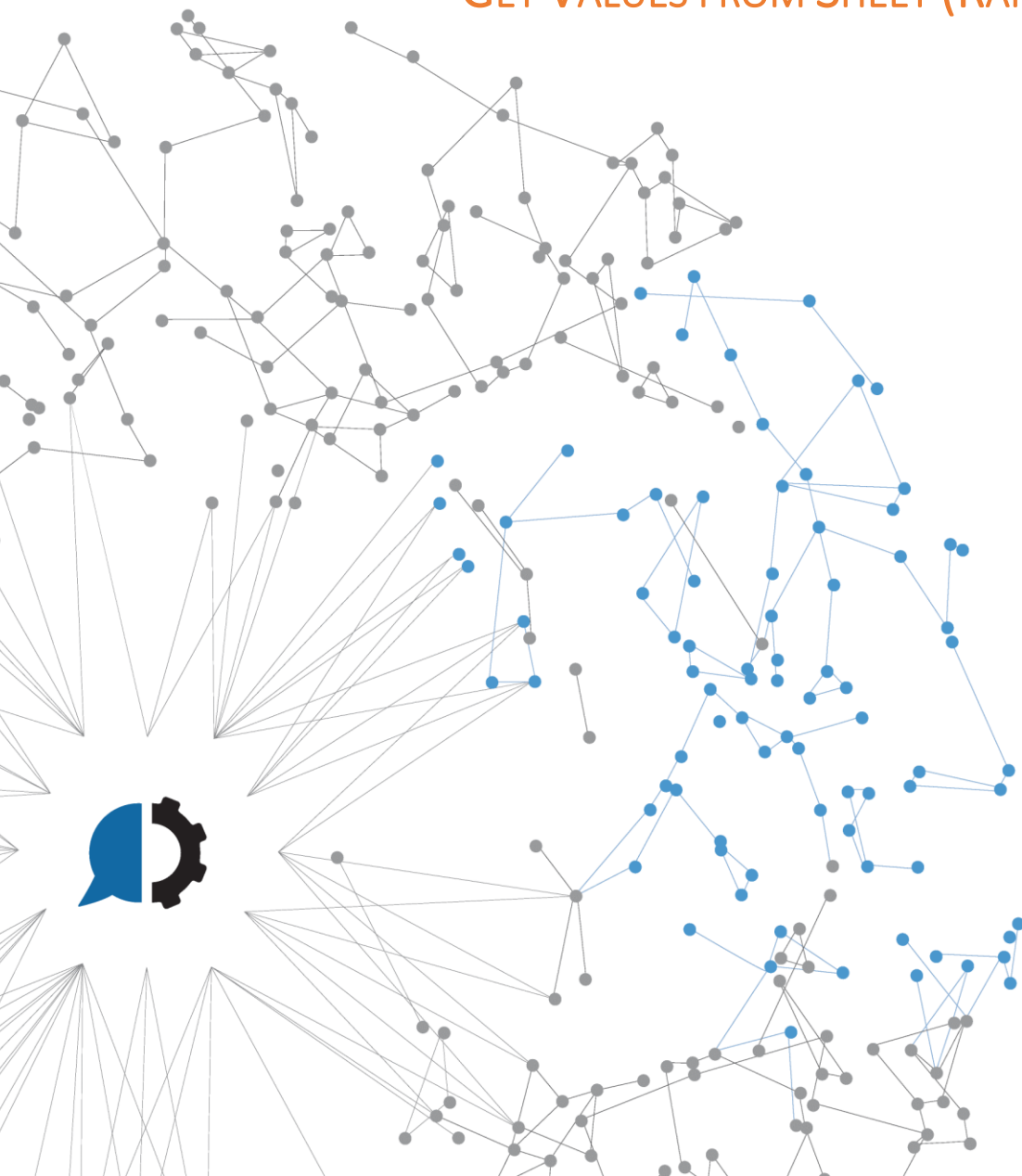




# GOOGLE SHEETS BOT

GET VALUES FROM SHEET (RANGE) TO DATASET





## Content

|   |   |
|---|---|
| OVERVIEW .....  | 3 |
| PREREQUISITES .....   | 4 |
| HOW TO IMPORT AN AUTOMATE TASK .....  | 6 |
| HOW TO EXECUTE THE GOOGLE SHEETS BOT - GET VALUES FROM SHEET (RANGE) TO DATASET ..... | 7 |
| APPENDIX A - TROUBLESHOOTING .....  | 9 |



## OVERVIEW

This how to guide introduces the *Google Sheets Bot - Get Values from Sheet (Range) to Dataset*. Instead of spending precious time and resources on repetitive, manual tasks, let Automate go to work and streamline your IT and business processes with robotic process automation. Automate can take over the manual steps which greatly reduces repetitive activities and improves the quality and consistency of the work. The top processes being automated are report generation, file movement, data import and export, and scheduling batch processing.

The ***Google Sheets Bot - Get Values from Sheet (Range) to Dataset*** is based in the **Google Sheets API**. It requests the data in a specified Google sheet **"Range"** by executing the **"*spreadsheets.values.get*"** HTTP GET method and ***creates an Automate Dataset object (Name: ds\_SheetValues)*** with the values that were received. Then you can use this dataset to loop the data and perform any desired action that you might need. Within the bot configuration, you will be able to set the Google Spreadsheet, the specific sheet name in case of multiple sheets, and the range for the values to be received. This task also provides an example on how to use the dataset and creates a ".CSV" file with the data.

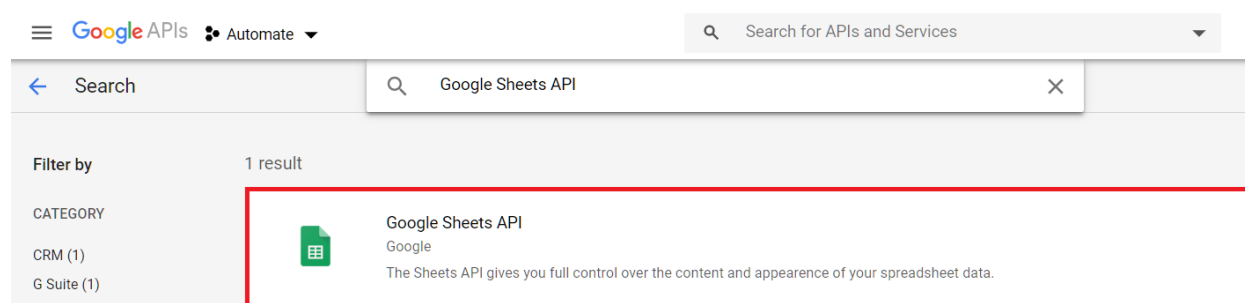
This bot uses an API Token authentication method to avoid using username and password. The token is associated to the account and scopes used during the Authorization Code generation. Carefully check the [PREREQUISITES](#) and the [HOW TO EXECUTE THE GOOGLE SHEETS BOT - GET VALUES FROM SHEET \(RANGE\) TO DATASET](#) sections to get detailed information.

There is also available a [GOOGLE BOT - GENERATE AND REFRESH TOKEN](#) that will provide an easy way for first creation and refreshing of the token for this bot to work in a daily basis. By default, the token expires after 1 hour. This bot is based in the OAuth Authorization Code Workflow.



## PREREQUISITES

- **Automate:** The Google Sheets Bot depends on Automate software in order to work. The minimal supported versions are:
  - [Automate Ultimate 11.2](#)
  - [Automate Plus 11.2](#)
  - [Automate Desktop 11.2](#)
- **Automate Markup Language file (.AML):** The primary file type used in Automate which contains the steps of our Google Sheets Bot task
- **Google Sheets API:** Our minimal requirements are
  - **Enable the Google Sheets API in your Google account:** In order to be able to connect through API to the Google Sheets, you need to enable it within your developer console account. To do so, enter the console with this [link](#). Once logged in for the first time, you will be asked to create a project, all your integrations with Google will be associated to a project. In next image you can see that we named the project “Automate”. Go to API [library](#) menu in the left and look for the Google Sheets API:



Once enabled the status will be shown as this:



### Google Sheets API

Google

The Sheets API gives you full control over the content and appearance of your spreadsheet data.

MANAGE

TRY THIS API [↗](#)

✓ API Enabled



- [Create a Token for connection with the Google Sheets API](#): Once the API was enabled, follow the procedure provided by Google within this [link](#), to obtain a **OAuth 2.0 Token** with one of the next Scopes to be able to read the Google Sheets Data:

***"https://www.googleapis.com/auth/spreadsheets"***

***"https://www.googleapis.com/auth/spreadsheets.readonly"***

The next Google Drive Scope, also give Authorization to perform this action:

***"https://www.googleapis.com/auth/drive"***

***"https://www.googleapis.com/auth/drive.readonly"***

Note that the scopes with *".readonly"* are just enough for this Bot.

Remember that we also provide in our [BotStore](#) the **"Google Sheets Bot - Generate and Refresh Token"** based on the OAuth2.0 Authorization Code Workflow. This bot will provide you an easy way to maintain the Token and Token-Refresh and be able to execute this bot on a daily basis after obtaining an **Authorization Code** with any of the scopes mentioned before.

- [Google Sheets API Version](#): This bot was created using the Google Sheets API v4.

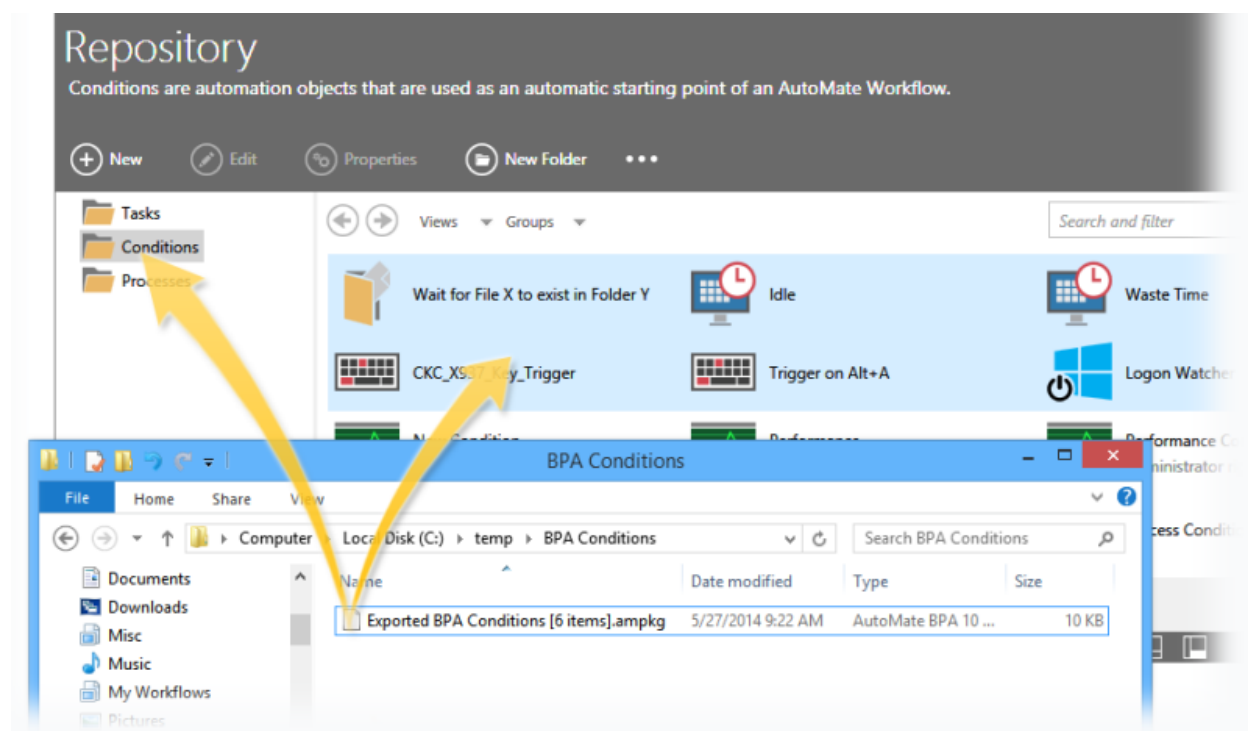


## HOW TO IMPORT AN AUTOMATE TASK

Compatible file types can be imported to the repository via drag-and-drop

From the [Server Management Console](#), navigate to the Repository section

Drag the desired file(s) from its original location and drop them into the folder in the [SMC](#). Files can be dropped into the folder icon or the main panel (as shown below). Imported object(s) are automatically placed into their corresponding repository location.



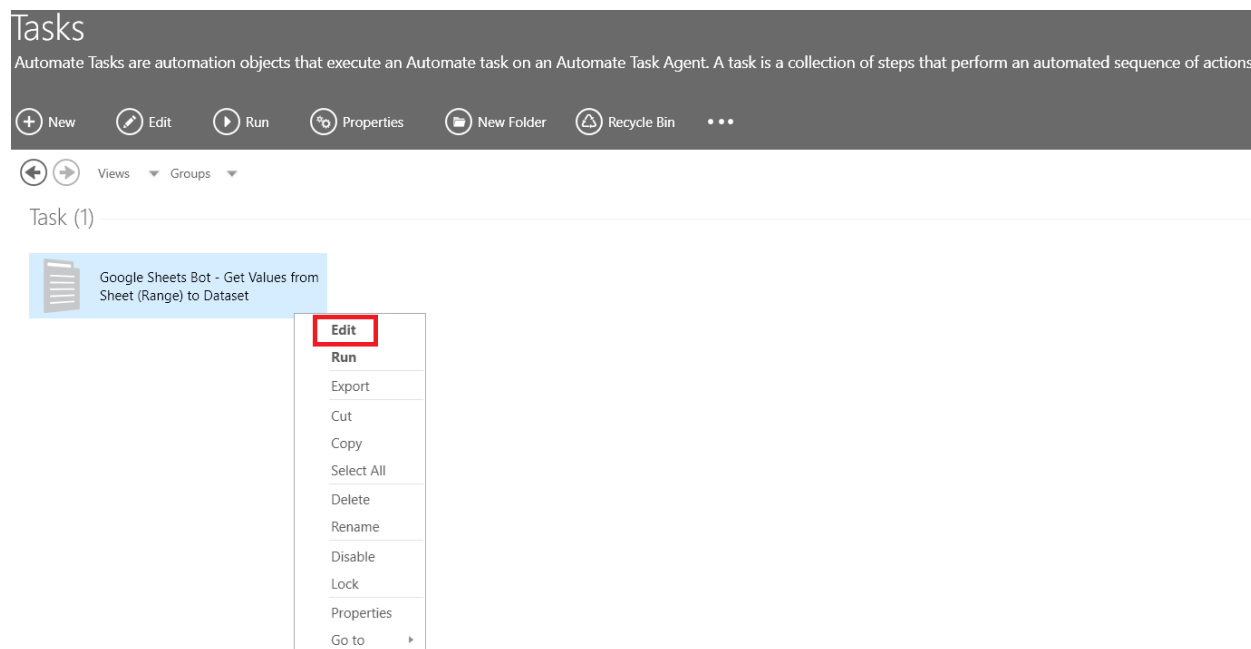


## HOW TO EXECUTE THE GOOGLE SHEETS BOT - GET VALUES FROM SHEET (RANGE) TO DATASET

If this is the first time running this task, we will need to set some parameters

Open the [Server Management Console](#) and locate the imported task

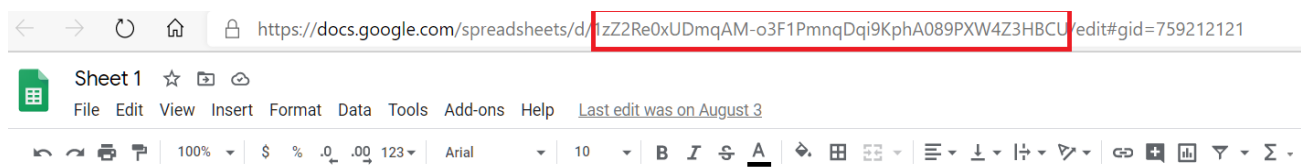
Edit the imported task by right clicking on the task and selecting [edit](#)



From [step 2](#) edit the next variables according to you desired output:

**IMPORTANT:** A Dataset with Name "*ds\_SheetValues*" will be created. Every row will contain the values separated in columns. If the selected range has headers column, they will be stored in the first row of the dataset.

- [var\\_GoogleSpreadsheetID](#): Set the ID of the Google Spreadsheet that you want to retrieve the values from. You can get the ID within the URL that's appear when you open the Google Sheet in your browser. Example: <https://docs.google.com/spreadsheets/d/1zZ2Re0xUDmqAM-o3F1PmnqDqi9KphA089PXW4Z3HBCU/edit#gid=75921212>



- [var\\_GoogleSheetName](#): In case of multiple sheets in a spreadsheet, set the name of the sheet the values will be retrieve from.
- [var\\_Range](#): Select the Range with the values to retrieve. Format example: A1:B10.



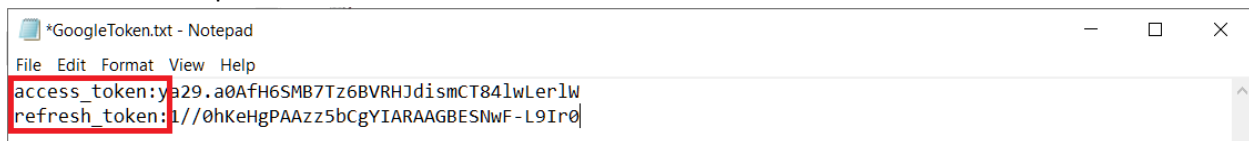
- **var\_Token:** Set your Token for connection (String) or the path of the file with the token information. Remember that the token will be associated to the account we connect/request with. It can be set to the token value itself or the path in which the .txt file with the token inside is located.

Example: **C:\Automate\ GoogleToken.txt**

In the case of using a .txt file, the token must have the next format/structure:

***access\_token:TokenString***

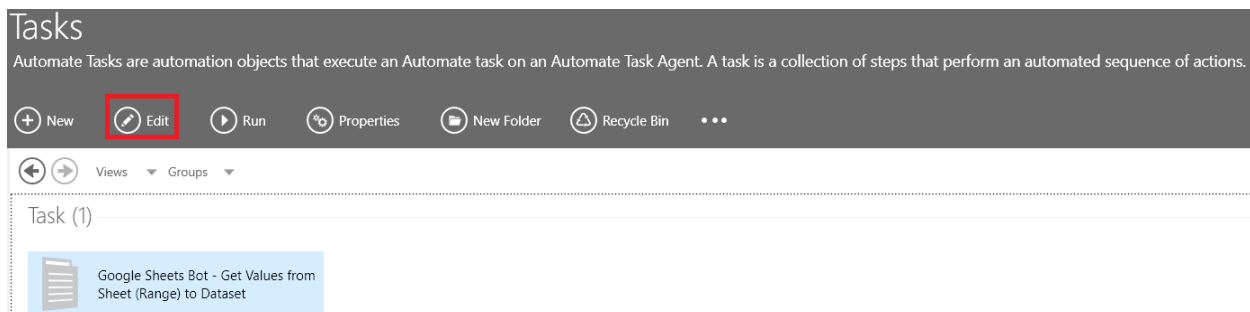
Example of the file:



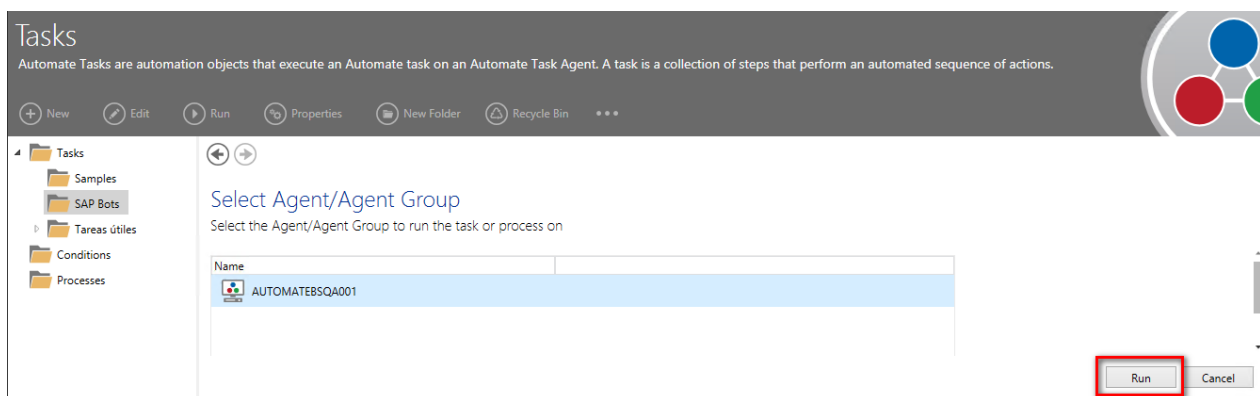
*Note that there is a refresh\_token value, this is managed by Google Bot - Generate and Refresh Token*

Save and close the task

Select the task and click on **Run**



Select your **Agent** and click on **Run** again







## APPENDIX A - TROUBLESHOOTING

- Logs: Each iteration of our bot creates a log file for troubleshooting. You can locate the log file under C:\Automate\Tasks\<<TASK NAME>>. By Default, the task name is *Google Sheets Bot - Get Values from Sheet (Range) to Dataset*.



[www.helpsystems.com](http://www.helpsystems.com)

### About HelpSystems

Organizations around the world rely on HelpSystems to make IT lives easier and keep business running smoothly. Our software and services monitor and AutoMate processes, encrypt and secure data, and provide easy access to the information people