Google Sheets Connect

Description

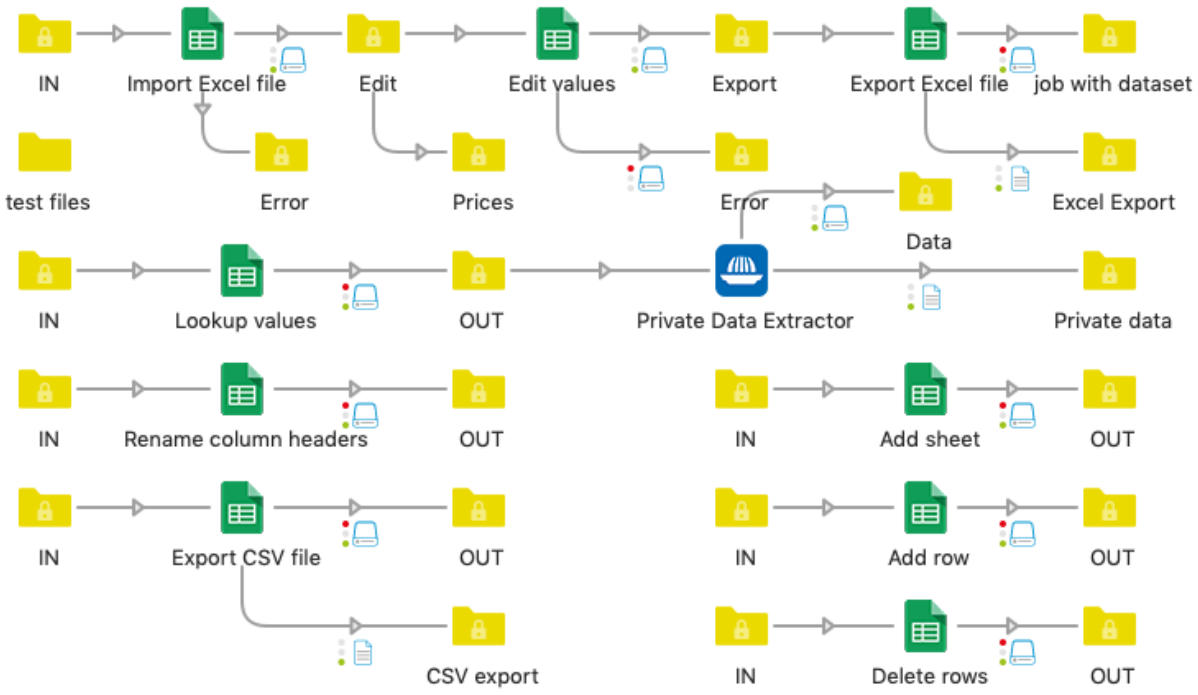
Google Sheets Connect allows you to process all these tasks with your Google Sheets:

- Add new sheet
- Delete sheet
- Rename sheet
- Rename column headers
- Add row
- Delete rows
- Lookup values
- Edit values (including formula's)
- Import CSV file
- Import Excel file
- Export sheet to Excel file
- Export sheet to XML file

For most of the tasks you can filter the lines of your sheets using up to 5 filters.
Lookup value will create private data with each cell of the found row(s).
You can also attach cell values as a dataset by exporting the sheet (or a filtered subset) as an XML dataset.

Moreover, install Google Sheets on your mobile devices, and log your Switch activity on your smartphone or tablet!

Please read the documentation before upgrading from versions 1 or 2, since the app will need a different setup and will not work with previous settings right after updating.



Setup

To allow the Google Sheets Connect app to edit your sheet(s) through the API, you need to:

1. create a Google Sheets Service Account key
2. share your sheet(s) with the email address from step 1
3. get the Spreadsheet ID

1. Create a Google Sheets Service Account key

Simply watch this short video and follow the steps: https://youtu.be/7H8oy0kQBF0
Click on this url to create the key: https://console.cloud.google.com
When you have the possibility to download the JSON key file, do so and keep your file in a safe place. You can download this JSON key file only once.

Open the JSON key file in a text editor, and copy the (very long) "private_key" value, without the quotes, and including the -----BEGIN PRIVATE KEY----- and -----END PRIVATE KEY-----\n
Paste it in the app's **Private Key property**.

Copy the JSON "client_email" value (same as the email address automatically created, as shown in the video) and paste it in the app's **Client email property**.

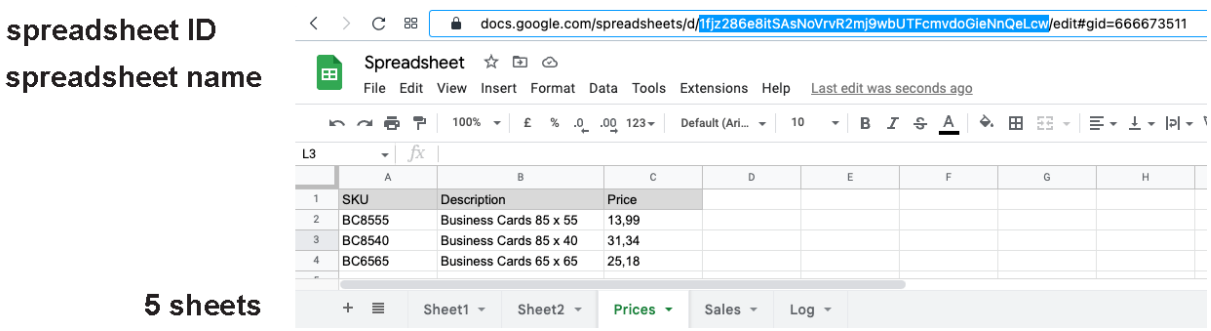
2. Share your sheet(s) with the client email (email address from step 1)

As described in the video, with "Editor" access rights.

3. Get the spreadsheet ID

It is part of the spreadsheet's url, between "https://docs.google.com/spreadsheets/d/" and "/edit". Paste it in the app's **Spreadsheet ID property**.

Note
Step 1 has to be done once per Google account.
Steps 2 and 3 have to be done once per spreadsheet (not sheet).



Flow Element Properties

- Private key, Client email, Spreadsheet ID
  - See setup on previous page
- Sheet name
  - Name of the sheet to use, modify, export, create, delete
- Task
  - Task you want to process
- Sheet header row (when adding new sheet)
  - Headers separated by ,
- Mapping (when renaming column headers)
  - Combination of old and new column headers, separated by =
- Content (when adding row)
  - Line content, one row per column, separated by ::
    Ex: columnHeader::SwitchVariable
- New row number private data tag (when adding row)
  - You can use this row number value further with Edit values or Delete rows
    Default: GSC.newRowNumber
- Content (when editing values)
  - Line content, one row per column, separated by ::
    Specify if you want to insert the new content as a value or a formula
    Example: columnHeader::SwitchVariable::value
    If you need the row number in your formula, use <<rowNumber>>
    Examples:
    Sum::E<<rowNumber>>+F<<rowNumber>>::formula
    Price::VLOOKUP(E<<rowNumber>>; prices!A1:B999; 2; FALSE)::formula
    TotalPrice::[Stats.NumberOfPages]*P<<rowNumber>>::formula
- New name (when renaming sheet)
  - New sheet's name
- Private data key prefix (when Looking up for values)
  - Private data prefix. The name of each column will be added to this prefix to define the complete private data tags. Default: GSC.
- Save the found line(s) as private data (when Looking up values)
  - First line, last line, all with separator
- File path (when importing Excel or CSV file)
  - File path of the file to import
- Destination (when importing Excel or CSV file)
  - New sheet, Existing sheet or Existing sheet, new if sheet doesn't exist
- Add timestamp column (when importing Excel or CSV file or adding row)
  - Yes-No. If Yes, define Timestamp value and format
- Source sheet choice (when importing Excel file)
  - Define source sheet, based on name or based on number

- Selection
  - Define the row(s) you want to lookup, edit, delete, export
    Choices: All rows, Last row, Use row number, Use filter
    When using filter(s), you can define the condition n column name (up to 5), then the operator and the value (and Decimal separator or Date or time format if needed)
- Add missing columns if needed (when importing, adding row or editing values)
  - Create additional columns if they do not already exist in the sheet
- Column width and Row height (when exporting to Excel file)
  - Column width and Row height values
- Send exported file to (when exporting sheet to Excel or XML file)
  - Dataset, Log success connection, Both

Important note concerning API calls

The Google API has a default maximum quota of 60 API calls per minute:
https://developers.google.com/sheets/api/limits

With Google Sheets Connect v3 and v4, you may exceed this limit when processing a large number of jobs at the same time, and some jobs may fail and go into the Problem Jobs.

Using the Hold element before Google Sheets Connect, and setting a "space jobs apart" value (in the connection properties) was a useful workaround. This is no longer necessary.

Google Sheets Connect v5 introduces a new mechanism to avoid exceeding this limit. The use of API calls is monitored, and Google Sheets Connect will wait a few seconds if necessary, to stay under the API call limit.

Tips and tricks, based on support experience

- All column headers have to be unique. If not, the API returns an empty list.
- The A1 cell may not be empty.
- With 'Edit values', if you get a ' sign before a number, use 'formula' instead of 'value'.
- Google Sheet formulas are so powerful! You can place an image in a cell this way:
  =IMAGE("https://avatars.githubusercontent.com/u/84084132", 4, 100,100)
- You can even do translations automatically, get stock prices, etc:
  Google Sheets function list: https://support.google.com/docs/table/25273

Private data

These private data are always added to the job:

GSC.outputLevel	Success or Error
GSC.outputMessage	The same message as in the log (messages), giving more details on the execution. Can be used in a condition, email, status message, etc.
GSC.task	Name of task processed
GSC.foundRows	Number of found rows (only with "Lookup values" task)

Compatibility

Switch 2021 Spring and higher, Windows & Mac

Connections

Google Sheets Connect requires at least one input and one output connection.
The job will be sent to the Data "success" output connection in case of success, or to the Data "error" output connection if the action fails.
The export tasks (export to Excel or CSV file) require a Log "success" output connection if you want to export the file separately (vs attached as a dataset).

What's new

v4

- Export sheet to XML as one file or separate files

v5

- Mechanism to avoid API quota error: app monitors API calls used and waits if needed
- Add extra column(s) during import of XLSX or CSV file
- Edit values now also works based on cell coordinates:
  H12::E<<rowNumber>>*F5::formula
- Bug fixes