
EZSheets Documentation

Al Sweigart

Jul 11, 2023

Contents

1	Installation	3
2	Quickstart	5
3	Unit Tests	7
4	API	9
	Python Module Index	11
	Index	13

A Pythonic interface to the Google Sheets API that actually works as of November 2019.

EZSheets can be installed from PyPI using *pip*:

```
pip install ezsheets
```

On macOS and Linux, installing EZSheets for Python 3 is done with *pip3*:

```
pip3 install ezsheets
```

If you run into permissions errors, try installing with the *--user* option:

```
pip install --user ezsheets
```

Before you can use EZSheets, you need to enable the Google Sheets and Google Drive APIs for your Google account. Visit the following web pages and click the Enable API buttons at the top of each:

- <https://console.developers.google.com/apis/library/sheets.googleapis.com/>
- <https://console.developers.google.com/apis/library/drive.googleapis.com/>

You'll also need to obtain three files, which you should save in the same folder as your .py Python script that uses EZSheets:

- A credentials file named `credentials-sheets.json`
- A token for Google Sheets named `token-sheets.pickle`
- A token for Google Drive named `token-drive.pickle`

The credentials file will generate the token files. The easiest way to obtain a credentials file is to go to the Google Sheets Python Quickstart page at <https://developers.google.com/sheets/api/quickstart/python/> and click the blue **Enable the Google Sheets API** button. You'll need to log in to your Google account to view this page.

Clicking this button will bring up a window with a **Download Client Configuration** link that lets you download a `credentials.json` file. Rename this file to `credentials-sheets.json` and place it in the same folder as your Python scripts.

Once you have a `credentials-sheets.json` file, run the import `ezsheets` module. The first time you import the EZSheets module, it will open a new browser window for you to log in to your Google account. Click **Allow**.

The message about Quickstart comes from the fact that you downloaded the credentials file from the Google Sheets Python Quickstart page. Note that this window will open twice: first for Google Sheets access and second for Google Drive access. EZSheets uses Google Drive access to upload, download, and delete spreadsheets.

After you log in, the browser window will prompt you to close it, and the *token-sheets.pickle* and *token-drive.pickle* files will appear in the same folder as *credentials-sheets.json*. You only need to go through this process the first time you run `import ezsheets`.

If you encounter an error after clicking **Allow** and the page seems to hang, make sure you have first enabled the Google Sheets and Drive APIs from the links at the start of this section. It may take a few minutes for Google's servers to register this change, so you may have to wait before you can use EZSheets.

Don't share the credential or token files with anyone: treat them like passwords.

CHAPTER 2

Quickstart

```
>>> import ezsheets

>>> ss = ezsheets.createSpreadsheet(title='My New Spreadsheet')

>>> ss.sheetTitles
('Sheet1',)

>>> ss.sheets
(Sheet(sheetId=0, title='Sheet1', rowCount=1000, columnCount=26),)

>>> sh = ss.sheets[0]
>>> sh.title
'Sheet1'

>>> sh.updateRow(1, ['Name', 'Species', 'Color', 'Weight'])
>>> sh.updateRow(2, ('Zophie', 'Cat', 'Gray', 11))
>>> sh[1, 1]
'Name'
>>> sh[1, 2]
'Zophie'
>>> sh['A2']
'Zophie'
>>> sh['A2'] = 'Pooka'
>>> sh['A2']
'Pooka'
```

The API section contains complete documentation.

CHAPTER 3

Unit Tests

The unit test suite takes approximately 7 minutes to run, due to the throttling.

CHAPTER 4

API

exception `ezsheets.EZSheetsException`

The base class for all EZSheets-specific problems. If the `ezsheets` module raises something that isn't this or a subclass of this exception, you can assume it is caused by a bug in EZSheets.

`ezsheets.IGNORE_QUOTA = False`

TODO - create a context manager to wrap calls, so that we can do both preventative throttling and automated retries if it somehow raises an exception. Also, use a sqlite database so that multiple scripts use the same queue.

Features to add: - delete spreadsheets - download as csv/excel/whatever

class `ezsheets.Sheet` (*spreadsheet, sheetId*)

This class represents an individual worksheet inside a spreadsheet. Sheets are composed of columns and rows of cells, which contain a single string value.

columnCount

The number of columns in this Sheet object.

columnGroupControlAfter

TODO

frozenColumnCount

The integer number of frozen columns in this Sheet object. Frozen columns remain visible in the browser even as the user scrolls the Sheet. There must be at least one non-frozen column in the Sheet.

frozenRowCount

The integer number of frozen rows in this Sheet object. Frozen rows remain visible in the browser even as the user scrolls down the Sheet. There must be at least one non-frozen row in the Sheet.

get (**args*)

Retrieve the value in a cell. The arguments to *get()* can either be two integers (column and row) or a single string such as 'A1'.

hideGridlines

The Boolean setting of whether the gridlines in the Sheet are visible or not.

id

The unique, read-only ID string of this Sheet object in its Spreadsheet. (This is the old, deprecated name. Use id instead.)

index

The integer index of this Sheet in its Spreadsheet's tuple of Sheet objects.

rowCount

The number of rows in this Sheet object.

rowGroupControlAfter

TODO

sheetId

The unique, read-only ID string of this Sheet object in its Spreadsheet. (This is the old, deprecated name. Use id instead.)

spreadsheet

The Spreadsheet object that contains this Sheet object.

tabColor

The color of the Sheet's tab as displayed in the browser.

title

The title of this Sheet on Google Sheets. Both Spreadsheets and Sheets have titles.

class `ezsheets.Spreadsheet` (*spreadsheetId*)

This class represents a Spreadsheet on Google Sheets. Spreadsheets can contain one or more sheets, also called worksheets.

createSheet (*title=""*, *index=None*, *columnCount=26*, *rowCount=1000*)

Create a new Sheet object in this Spreadsheet.

id

The unique, read-only id for this Spreadsheet on Google Sheets.

refresh ()

Updates this Spreadsheet object's Sheet objects with the current data of the spreadsheet and sheets on Google sheets.

sheetTitles

A tuple of the Sheet objects' titles (as strings) in this Spreadsheet object.

spreadsheetId

The unique, read-only id for this Spreadsheet on Google Sheets. (This is the old, deprecated name. Use id instead.)

title

The string title for this Spreadsheet on Google Sheets. Both Spreadsheets and Sheets have titles.

url

The URL for this Spreadsheet on Google Sheets.

`ezsheets.getColumnLetterOf` (*columnNumber*)

`getColumnLetterOf(1) => 'A', getColumnLetterOf(27) => 'AA'`

`ezsheets.getColumnNumberOf` (*columnLetter*)

`getColumnNumberOf('A') => 1, getColumnNumberOf('AA') => 27`

e

ezsheets, 9

C

columnCount (*ezsheets.Sheet* attribute), 9
columnGroupControlAfter (*ezsheets.Sheet*
attribute), 9
createSheet () (*ezsheets.Spreadsheet* method), 10

E

ezsheets (*module*), 9
EZSheetsException, 9

F

frozenColumnCount (*ezsheets.Sheet* attribute), 9
frozenRowCount (*ezsheets.Sheet* attribute), 9

G

get () (*ezsheets.Sheet* method), 9
getColumnLetterOf () (*in module ezsheets*), 10
getColumnNumberOf () (*in module ezsheets*), 10

H

hideGridlines (*ezsheets.Sheet* attribute), 9

I

id (*ezsheets.Sheet* attribute), 9
id (*ezsheets.Spreadsheet* attribute), 10
IGNORE_QUOTA (*in module ezsheets*), 9
index (*ezsheets.Sheet* attribute), 10

R

refresh () (*ezsheets.Spreadsheet* method), 10
rowCount (*ezsheets.Sheet* attribute), 10
rowGroupControlAfter (*ezsheets.Sheet* attribute),
10

S

Sheet (*class in ezsheets*), 9
sheetId (*ezsheets.Sheet* attribute), 10
sheetTitles (*ezsheets.Spreadsheet* attribute), 10
Spreadsheet (*class in ezsheets*), 10

spreadsheet (*ezsheets.Sheet* attribute), 10
spreadsheetId (*ezsheets.Spreadsheet* attribute), 10

T

tabColor (*ezsheets.Sheet* attribute), 10
title (*ezsheets.Sheet* attribute), 10
title (*ezsheets.Spreadsheet* attribute), 10

U

url (*ezsheets.Spreadsheet* attribute), 10