
df2gsread Documentation

Eduard Trott

May 03, 2018

Contents

1	df2gsread	3
1.1	Description	3
1.2	Status	3
1.3	Install	3
1.4	Access Credentials	4
1.5	Usage	4
1.6	Documentation	5
1.7	Testing	5
1.8	Development	5
2	Install	7
2.1	PIP	7
3	Examples	9
3.1	Google Spreadsheet to Pandas DataFrame	9
3.2	Pandas DataFrame to Google Spreadsheet	10
4	Indices and tables	11

Python library, which allows you easily interact with Google Spreadsheets.

Table of Contents:

CHAPTER 1

df2gsread

Transfer data between Google Spreadsheets and Pandas DataFrame.

1.1 Description

Python library that provides possibility to transport table-data between Google Spreadsheets and Pandas DataFrame for further management or processing. Can be useful in all cases, when you need to handle the data located in Google Drive.

1.2 Status

Latest Release	
Build	
Docs	
License	

1.3 Install

Example install, using VirtualEnv:

```
# install/use python virtual environment
virtualenv ~/virtenv_scratch --no-site-packages

# activate the virtual environment
source ~/virtenv_scratch/bin/activate
```

(continues on next page)

(continued from previous page)

```
# upgrade pip in the new virtenv
pip install -U pip setuptools

# install this package in DEVELOPMENT mode
# python setup.py develop

# simply install
# python setup.py install

# or install via pip
pip install df2gsread
```

1.4 Access Credentials

To allow a script to use Google Drive API we need to authenticate our self towards Google. To do so, we need to create a project, describing the tool and generate credentials. Please use your web browser and go to [Google console](#) and :

- Choose **“Create Project”** in popup menu on the top.
- A dialog box appears, so give your project a name and click on **“Create”** button.
- On the left-side menu click on **“API Manager”**.
- A table of available APIs is shown. Switch **“Drive API”** and click on **“Enable API”** button. Other APIs might be switched off, for our purpose.
- On the left-side menu click on **“Credentials”**.
- In section **“OAuth consent screen”** select your email address and give your product a name. Then click on **“Save”** button.
- In section **“Credentials”** click on **“Add credentials”** and switch **“OAuth 2.0 client ID”**.
- A dialog box **“Create Cliend ID”** appears. Select **“Application type”** item as **“Other”**.
- Click on **“Create”** button.
- Click on **“Download JSON”** icon on the right side of created **“OAuth 2.0 client IDs”** and store the downloaded file on your file system. Please be aware, the file contains your private credentials, so take care of the file in the same way you care of your private SSH key; i.e. move downloaded JSON file to **~/.gdrive_private**.
- Then, the first time you run it your browser window will open a google authorization request page. Approve authorization and then the credentials will work as expected.

1.5 Usage

Run df2gsread like:

```
from df2gsread import df2gsread as d2g
import pandas as pd
d = [pd.Series([1., 2., 3.], index=['a', 'b', 'c']),
     pd.Series([1., 2., 3., 4.], index=['a', 'b', 'c', 'd'])]
df = pd.DataFrame(d)
```

(continues on next page)

(continued from previous page)

```
# use full path to spreadsheet file
spreadsheet = '/some/folder/New Spreadsheet'
# or spreadsheet file id
# spreadsheet = '1cIOgi90...'

wks_name = 'New Sheet'

d2g.upload(df, spreadsheet, wks_name)
# if spreadsheet already exists, all data of provided worksheet(or first as default)
# will be replaced with data of given DataFrame, make sure that this is what you need!
```

Run gsread2df like:

```
from df2gsread import gsread2df as g2d

# use full path to spreadsheet file
spreadsheet = '/some/folder/New Spreadsheet'
# or spreadsheet file id
# spreadsheet = '1cIOgi90...'
wks_name = 'New Sheet'

df = g2d.download(spreadsheet, wks_name, col_names = True, row_names = True)
```

1.6 Documentation

Documentation is available [here](#).

1.7 Testing

Testing is py.test based. Run with:

```
py.test tests/ -v
```

Or with coverage:

```
coverage run --source df2gsread -m py.test
coverage report
```

1.8 Development

Install the supplied githooks; eg:

```
ln -s ~/repos/df2gsread/_githooks/commit-msg ~/repos/df2gsread/.git/hooks/commit-msg
ln -s ~/repos/df2gsread/_githooks/pre-commit ~/repos/df2gsread/.git/hooks/pre-commit
```


2.1 PIP

Basic dependencies for buiding/installing pip packages:

```
sudo yum install gcc krb5-devel  
sudo yum install python-devel python-pip python-virtualenv
```

Upgrade to the latest pip/setup/virtualenv installer code:

```
sudo pip install --upgrade pip setuptools virtualenv
```

Install into a python virtual environment (OPTIONAL):

```
virtualenv ~/df2gspread  
source ~/df2gspread/bin/activate
```

Install df2gspread (sudo required if not in a virtualenv):

```
pip install df2gspread
```

See the [pypi package index](#) for detailed package information.

3.1 Google Spreadsheet to Pandas DataFrame

`df2gspread.gspread2df.download(gfile, wks_name=None, col_names=False, row_names=False, credentials=None, start_cell='A1')`

Download Google Spreadsheet and convert it to Pandas DataFrame

Parameters

- **gfile** (*str*) – path to Google Spreadsheet or gspread ID
- **wks_name** (*str*) – worksheet name
- **col_names** (*bool*) – assing top row to column names for Pandas DataFrame
- **row_names** (*bool*) – assing left column to row names for Pandas DataFrame
- **credentials** (*class 'oauth2client.client.OAuth2Credentials'*) – provide own credentials
- **start_cell** (*str*) – specify where to start capturing of the DataFrame; default is A1

Returns Pandas DataFrame

Return type class 'pandas.core.frame.DataFrame'

Example

```
>>> from df2gspread import gspread2df as g2d
>>> df = g2d.download(gfile="1U-kSDyeD-...", col_names=True, row_
↳ names=True)
>>> df
```

	col1	col2
field1	1	2
field2	3	4

3.2 Pandas DataFrame to Google Spreadsheet

```
df2gsread.df2gsread.upload(df, gfile='/New Spreadsheet', wks_name=None,
                           chunk_size=1000, col_names=True, row_names=True,
                           clean=True, credentials=None, start_cell='A1', df_size=False,
                           new_sheet_dimensions=(1000, 100))
```

Upload given Pandas DataFrame to Google Drive and returns gsread Worksheet object

Parameters

- **df** (*class 'pandas.core.frame.DataFrame'*) – Pandas DataFrame
- **gfile** (*str*) – path to Google Spreadsheet or gsread ID
- **wks_name** (*str*) – worksheet name
- **chunk_size** (*int*) – size of chunk to upload
- **col_names** (*bool*) – passing top row to column names for Pandas DataFrame
- **row_names** (*bool*) – passing left column to row names for Pandas DataFrame
- **clean** (*bool*) – clean all data in worksheet before uploading
- **credentials** (*class 'oauth2client.client.OAuth2Credentials'*) – provide own credentials
- **start_cell** (*str*) – specify where to insert the DataFrame; default is A1
- **df_size** (*bool*) – -If True and worksheet name does NOT exist, will create a new worksheet that is the size of the df; otherwise, by default, creates sheet of 1000x100 cells. -If True and worksheet does exist, will resize larger or smaller to fit new dataframe. -If False and dataframe is larger than existing sheet, will resize the sheet larger. -If False and dataframe is smaller than existing sheet, does not resize.
- **new_sheet_dimensions** (*tuple*) – tuple of (row, cols) for size of a new sheet

Returns gsread Worksheet

Return type class 'gsread.models.Worksheet'

Example

```
>>> from df2gsread import df2gsread as d2g
>>> import pandas as pd
>>> df = pd.DataFrame([1 2 3])
>>> wks = d2g.upload(df, wks_name='Example worksheet')
>>> wks.title
'Example worksheet'
```

CHAPTER 4

Indices and tables

- `genindex`

D

`download()` (in module `df2gsread.gspread2df`), [9](#)

U

`upload()` (in module `df2gsread.df2gsread`), [10](#)