gspread Documentation

Release 3.0.0

Anton Burnashev

Apr 12, 2018

Contents

gspread is a Python client library for the Google Sheets API.

CHAPTER 1

Main Interface

gspread.**authorize**(*credentials*, *client_class=<class 'gspread.client.Client'>*)

Login to Google API using OAuth2 credentials. This is a shortcut function which instantiates `gspread.client.Client` and performs login right away.

Returns `gspread.client.Client` instance.

class gspread.**Client**(*auth*, *session=None*)

An instance of this class communicates with Google API.

Parameters

- **auth** – An OAuth2 credential object. Credential objects are those created by the oauth2client library. https://github.com/google/oauth2client
- **session** – (optional) A session object capable of making HTTP requests while persisting some parameters across requests. Defaults to requests.Session.

```
>>> c = gspread.Client(auth=OAuthCredentialObject)
```

create(*title*)

Creates a new spreadsheet.

Parameters **title** – A title of a new spreadsheet.

Returns a *Spreadsheet* instance.

Note: In order to use this method, you need to add `https://www.googleapis.com/auth/drive` to your oAuth scope.

Example:

```
scope = [
    'https://spreadsheets.google.com/feeds',
    'https://www.googleapis.com/auth/drive'
]
```

Otherwise you will get an `Insufficient Permission` error when you try to create a new spreadsheet.

---

del_spreadsheet(*file_id*)
Deletes a spreadsheet.

Parameters **file_id** – a spreadsheet ID (aka file ID.)

import_csv(*file_id*, *data*)
Imports data into the first page of the spreadsheet.

Parameters **data** – A CSV string of data.

insert_permission(*file_id*, *value*, *perm_type*, *role*, *notify=True*, *email_message=None*)
Creates a new permission for a file.

Parameters

- **file_id** – a spreadsheet ID (aka file ID.)
- **value** – user or group e-mail address, domain name or None for 'default' type.
- **perm_type** – the account type. Allowed values are: `user`, `group`, `domain`, `anyone`
- **role** – the primary role for this user. Allowed values are: `owner`, `writer`, `reader`
- **notify** – Whether to send an email to the target user/domain.
- **email_message** – an email message to be sent if notify=True.

Examples:

```
# Give write permissions to otto@example.com

gc.insert_permission(
    '0BmgG6nO_6dprnRRUWl1UFE',
    'otto@example.org',
    perm_type='user',
    role='writer'
)

# Make the spreadsheet publicly readable

gc.insert_permission(
    '0BmgG6nO_6dprnRRUWl1UFE',
    None,
    perm_type='anyone',
    role='reader'
)
```

list_permissions(*file_id*)
Retrieve a list of permissions for a file.

Parameters **file_id** – a spreadsheet ID (aka file ID.)

login()
Authorize client.

open(*title*)
Opens a spreadsheet.

Parameters **title** – A title of a spreadsheet.

Returns a *Spreadsheet* instance.

If there's more than one spreadsheet with same title the first one will be opened.

Raises **gspread.SpreadsheetNotFound** – if no spreadsheet with specified *title* is found.

```
>>> c = gspread.authorize(credentials)
>>> c.open('My fancy spreadsheet')
```

open_by_key(*key*)

Opens a spreadsheet specified by *key*.

Parameters **key** – A key of a spreadsheet as it appears in a URL in a browser.

Returns a *Spreadsheet* instance.

```
>>> c = gspread.authorize(credentials)
>>> c.open_by_key('0BmgG6nO_6dprdS1MN3d3MkdPa142WFRrdnRRUWl1UFE')
```

open_by_url(*url*)

Opens a spreadsheet specified by *url*.

Parameters **url** – URL of a spreadsheet as it appears in a browser.

Returns a Spreadsheet instance.

Raises **gspread.SpreadsheetNotFound** – if no spreadsheet with specified *url* is found.

```
>>> c = gspread.authorize(credentials)
>>> c.open_by_url('https://docs.google.com/spreadsheet/ccc?key=0Bm...FE&hl')
```

openall(*title=None*)

Opens all available spreadsheets.

Parameters **title** – (optional) If specified can be used to filter spreadsheets by title.

Returns a list of *Spreadsheet* instances.

remove_permission(*file_id*, *permission_id*)

Deletes a permission from a file.

Parameters

- **file_id** – a spreadsheet ID (aka file ID.)
- **permission_id** – an ID for the permission.

CHAPTER 2

Models

The models represent common spreadsheet objects: `a spreadsheet`, `a worksheet` and `a cell`.

Note: The classes described below should not be instantiated by end-user. Their instances result from calling other objects' methods.

class gspread.models.**Spreadsheet**(*client*, *properties*)

The class that represents a spreadsheet.

add_worksheet(*title*, *rows*, *cols*)

Adds a new worksheet to a spreadsheet.

Parameters

- **title** – A title of a new worksheet.
- **rows** – Number of rows.
- **cols** – Number of columns.

Returns a newly created `worksheets`.

del_worksheet(*worksheet*)

Deletes a worksheet from a spreadsheet.

Parameters **worksheet** – The worksheet to be deleted.

get_worksheet(*index*)

Returns a worksheet with specified *index*.

Parameters **index** – An index of a worksheet. Indexes start from zero.

Returns an instance of `gsperad.models.Worksheet` or *None* if the worksheet is not found.

Example. To get first worksheet of a spreadsheet:

```
>>> sht = client.open('My fancy spreadsheet')
>>> worksheet = sht.get_worksheet(0)
```

id
Spreadsheet ID.

list_permissions()
Lists the spreadsheet's permissions.

remove_permissions(*value*, *role='any'*)
Example:

```
# Remove Otto's write permission for this spreadsheet
sh.remove_permissions('otto@example.com', role='writer')

# Remove all Otto's permissions for this spreadsheet
sh.remove_permissions('otto@example.com')
```

share(*value*, *perm_type*, *role*, *notify=True*, *email_message=None*)
Share the spreadsheet with other accounts. :param value: user or group e-mail address, domain name

or None for 'default' type.

Parameters

- **perm_type** – the account type. Allowed values are: `user`, `group`, `domain`, `anyone`.
- **role** – the primary role for this user. Allowed values are: `owner`, `writer`, `reader`.
- **notify** – Whether to send an email to the target user/domain.
- **email_message** – The email to be sent if notify=True

Example:

```
# Give Otto a write permission on this spreadsheet
sh.share('otto@example.com', perm_type='user', role='writer')

# Transfer ownership to Otto
sh.share('otto@example.com', perm_type='user', role='owner')
```

sheet1
Shortcut property for getting the first worksheet.

title
Spreadsheet title.

updated
Deprecated since version 2.0.

This feature is not supported in Sheets API v4.

worksheet(*title*)
Returns a worksheet with specified *title*.

Parameters **title** – A title of a worksheet. If there're multiple worksheets with the same title, first one will be returned.

Returns an instance of `gsperad.models.Worksheet`.

Example. Getting worksheet named 'Annual bonuses'

```
>>> sht = client.open('Sample one')
>>> worksheet = sht.worksheet('Annual bonuses')
```

worksheets()
Returns a list of all `worksheets` in a spreadsheet.

class gspread.models.**Worksheet**(*spreadsheet*, *properties*)
The class that represents a single sheet in a spreadsheet (aka "worksheet").

acell(*label*, *value_render_option='FORMATTED_VALUE'*)
Returns an instance of a *gspread.models.Cell*.

Parameters

- **label** – String with cell label in common format, e.g. 'B1'. Letter case is ignored.
- **value_render_option** – Determines how values should be rendered in the the output. See ValueRenderOption in the Sheets API.

Example:

```
>>> worksheet.acell('A1')
<Cell R1C1 "I'm cell A1">
```

add_cols(*cols*)
Adds colums to worksheet.

Parameters **cols** – Columns number to add.

add_rows(*rows*)
Adds rows to worksheet.

Parameters **rows** – Rows number to add.

append_row(*values*, *value_input_option='RAW'*)
Adds a row to the worksheet and populates it with values. Widens the worksheet if there are more values than columns.

Parameters **values** – List of values for the new row.

cell(*row*, *col*, *value_render_option='FORMATTED_VALUE'*)
Returns an instance of a *gspread.models.Cell* positioned in *row* and *col* column.

Parameters

- **row** – Integer row number.
- **col** – Integer column number.
- **value_render_option** – Determines how values should be rendered in the the output. See ValueRenderOption in the Sheets API.

Example:

```
>>> worksheet.cell(1, 1)
<Cell R1C1 "I'm cell A1">
```

clear()
Clears all cells in the worksheet.

col_count
Number of columns.

col_values(*col*, *value_render_option='FORMATTED_VALUE'*)
Returns a list of all values in column *col*.

Empty cells in this list will be rendered as `None`.

Parameters

- **col** – Integer column number.
- **value_render_option** – Determines how values should be rendered in the the output. See ValueRenderOption in the Sheets API.

delete_row(*index*)
"Deletes a row from the worksheet at the specified index.

Parameters **index** – Index of a row for deletion.

export(*format*)
Deprecated since version 2.0.

This feature is not supported in Sheets API v4.

find(*query*)
Finds first cell matching query.

Parameters **query** – A text string or compiled regular expression.

findall(*query*)
Finds all cells matching query.

Parameters **query** – A text string or compiled regular expression.

get_all_records(*empty2zero=False*, *head=1*, *default_blank=''*)
Returns a list of dictionaries, all of them having the contents of the spreadsheet with the head row as keys and each of these dictionaries holding the contents of subsequent rows of cells as values.

Cell values are numericised (strings that can be read as ints or floats are converted).

Parameters

- **empty2zero** – determines whether empty cells are converted to zeros.
- **head** – determines wich row to use as keys, starting from 1 following the numeration of the spreadsheet.
- **default_blank** – determines whether empty cells are converted to something else except empty string or zero.

get_all_values()
Returns a list of lists containing all cells' values as strings.

id
Id of a worksheet.

insert_row(*values*, *index=1*, *value_input_option='RAW'*)
Adds a row to the worksheet at the specified index and populates it with values.

Widens the worksheet if there are more values than columns.

Parameters

- **values** – List of values for the new row.
- **value_input_option** – Determines how input data should be interpreted. See ValueInputOption in the Sheets API.

range(**args*, ***kwargs*)
Returns a list of *Cell* objects from a specified range.

Parameters **name** – A string with range value in A1 notation, e.g. 'A1:A5'.

Alternatively, you may specify numeric boundaries. All values index from 1 (one):

Parameters

- **first_row** – Integer row number
- **first_col** – Integer row number
- **last_row** – Integer row number
- **last_col** – Integer row number

Example:

```
>>> # Using A1 notation
>>> worksheet.range('A1:B7')
[<Cell R1C1 "42">, ...]

>>> # Same with numeric boundaries
>>> worksheet.range(1, 1, 7, 2)
[<Cell R1C1 "42">, ...]
```

resize(*rows=None*, *cols=None*)
Resizes the worksheet.

Parameters

- **rows** – New rows number.
- **cols** – New columns number.

row_count
Number of rows.

row_values(*row*, *value_render_option='FORMATTED_VALUE'*)
Returns a list of all values in a *row*.

Empty cells in this list will be rendered as `None`.

Parameters

- **row** – Integer row number.
- **value_render_option** – Determines how values should be rendered in the the output. See ValueRenderOption in the Sheets API.

title
Title of a worksheet.

update_acell(*label*, *value*)
Sets the new value to a cell.

Parameters

- **label** – String with cell label in common format, e.g. 'B1'. Letter case is ignored.
- **value** – New value.

Example:

```
worksheet.update_acell('A1', '42')
```

update_cell(*row*, *col*, *value*)
Sets the new value to a cell.

Parameters

- **row** – Row number.
- **col** – Column number.
- **value** – New value.

Example:

```
worksheet.update_cell(1, 1, '42')
```

update_cells(*cell_list*, *value_input_option='RAW'*)
Updates cells in batch.

Parameters

- **cell_list** – List of a *Cell* objects to update.
- **value_input_option** – Determines how input data should be interpreted. See ValueInputOption in the Sheets API.

Example:

```
# Select a range
cell_list = worksheet.range('A1:C7')

for cell in cell_list:
    cell.value = 'O_o'

# Update in batch
worksheet.update_cells(cell_list)
```

update_title(*title*)
Renames the worksheet.

Parameters **title** – A new title.

updated
Deprecated since version 2.0.

This feature is not supported in Sheets API v4.

class gspread.models.**Cell**(*row*, *col*, *value=''*)
An instance of this class represents a single cell in a *worksheet*.

col
Column number of the cell.

input_value
Deprecated since version 2.0.

This feature is not supported in Sheets API v4.

row
Row number of the cell.

value = None
Value of the cell.

CHAPTER 3

Utils

3.1 gspread.utils

This module contains utility functions.

gspread.utils.**rowcol_to_a1**(*row*, *col*)

Translates a row and column cell address to A1 notation.

Parameters

- **row** – The row of the cell to be converted. Rows start at index 1.
- **col** – The column of the cell to be converted. Columns start at index 1.

Returns a string containing the cell's coordinates in A1 notation.

Example:

```
>>> rowcol_to_a1(1, 1)
A1
```

gspread.utils.**a1_to_rowcol**(*label*)

Translates a cell's address in A1 notation to a tuple of integers.

Parameters **label** – String with cell label in A1 notation, e.g. 'B1'. Letter case is ignored.

Returns a tuple containing *row* and *column* numbers. Both indexed from 1 (one).

Example:

```
>>> a1_to_rowcol('A1')
(1, 1)
```

CHAPTER 4

Exceptions

exception gspread.exceptions.**GSpreadException**
A base class for gspread's exceptions.

exception gspread.exceptions.**APIError**(*response*)

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

g

Index

T

U

V

W