
gsread Documentation

Release 0.2.5

Anton Burnashev

June 18, 2015

1	Main Interface	3
2	Models	5
3	Exceptions	11
4	Internal Modules	13
4.1	gspread.httpsession	13
4.2	gspread.urls	13
5	Indices and tables	15
	Python Module Index	17

gsread is a simple Google Spreadsheets API wrapper.

- *Main Interface*
- *Models*
- *Exceptions*
- *Internal Modules*
 - *gsread.httpsession*
 - *gsread.urls*

Main Interface

`gspread.login(email, password)`

Login to Google API using *email* and *password*.

This is a shortcut function which instantiates *Client* and performs login right away.

Returns *Client* instance.

`gspread.authorize(credentials)`

Login to Google API using OAuth2 credentials.

This is a shortcut function which instantiates *Client* and performs login right away.

Returns *Client* instance.

class `gspread.Client(auth, http_session=None)`

An instance of this class communicates with Google Data API.

Parameters

- **auth** – A tuple containing an *email* and a *password* used for ClientLogin authentication or an OAuth2 credential object. Credential objects are those created by the `oauth2client` library. <https://github.com/google/oauth2client>
- **http_session** – (optional) A session object capable of making HTTP requests while persisting headers. Defaults to *HTTPSession*.

```
>>> c = gspread.Client(auth=('user@example.com', 'qwertypassword'))
```

or

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

login()

Authorize client using ClientLogin protocol.

The credentials provided in *auth* parameter to class' constructor will be used.

This method is using API described at: <http://code.google.com/apis/accounts/docs/AuthForInstalledApps.html>

Raises AuthenticationError if login attempt fails.

open(title)

Opens a spreadsheet, returning a *Spreadsheet* instance.

Parameters title – A title of a spreadsheet.

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 = gsread.Client(auth=('user@example.com', 'qwertypassword'))
>>> c.login()
>>> c.open('My fancy spreadsheet')
```

open_by_key (*key*)

Opens a spreadsheet specified by *key*, returning a *Spreadsheet* instance.

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

Raises *gsread.SpreadsheetNotFound* if no spreadsheet with specified *key* is found.

```
>>> c = gsread.Client(auth=('user@example.com', 'qwertypassword'))
>>> c.login()
>>> c.open_by_key('0BmgG6nO_6dprdS1MN3d3MkdPa142WFRrdnRRUWl1UFE')
```

open_by_url (*url*)

Opens a spreadsheet specified by *url*, returning a *Spreadsheet* instance.

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

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

```
>>> c = gsread.Client(auth=('user@example.com', 'qwertypassword'))
>>> c.login()
>>> c.open_by_url('https://docs.google.com/spreadsheet/cc?key=0Bm...FE&hl')
```

openall (*title=None*)

Opens all available spreadsheets, returning a list of a *Spreadsheet* instances.

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

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 `gsread.Spreadsheet` (*client, feed_entry*)

A class for a spreadsheet object.

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*.

The returning object is an instance of *Worksheet*.

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

Example. To get first worksheet of a spreadsheet:

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

Returns *None* if the worksheet is not found.

sheet1

Shortcut property for getting the first worksheet.

worksheet (*title*)

Returns a worksheet with specified *title*.

The returning object is an instance of *Worksheet*.

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

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 gsread.**Worksheet** (*spreadsheet, element*)

A class for worksheet object.

acell (*label*)

Returns an instance of a *Cell*.

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

Example:

```
>>> wks.acell('A1') # this could be 'a1' as well
<Cell R1C1 "I'm cell A1">
```

add_cols (*cols*)

Adds columns to worksheet.

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

add_rows (*rows*)

Adds rows to worksheet.

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

append_row (*values*)

“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*)

Returns an instance of a *Cell* positioned in *row* and *col* column.

Parameters

- **row** – Integer row number.
- **col** – Integer column number.

Example:

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

col_count

Number of columns

col_values (*col*)

Returns a list of all values in column *col*.

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

export (*format='csv'*)

Export the worksheet in specified format.

Parameters **format** – A format of the output.

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_addr_int(*row, col*)

Translates cell's tuple of integers to a cell label.

The result is a string containing the cell's coordinates in label form.

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.

Example:

```
>>> wks.get_addr_int(1, 1)
A1
```

get_all_records(*empty2zero=False, head=1*)

Returns a list of dictionaries, all of them having:

- the contents of the spreadsheet's 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.

get_all_values()

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

get_int_addr(*label*)

Translates cell's label address to a tuple of integers.

The result is a tuple containing *row* and *column* numbers.

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

Example:

```
>>> wks.get_int_addr('A1')
(1, 1)
```

id

Id of a worksheet.

insert_row(*values, index=1*)

“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.

range (*alphanum*)

Returns a list of *Cell* objects from specified range.

Parameters **alphanum** – A string with range value in common format, e.g. 'A1:A5'.

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*)

Returns a list of all values in a *row*.

Empty cells in this list will be rendered as None.

title

Title of a worksheet.

update_acell (*label, val*)

Sets the new value to a cell.

Parameters

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

Example:

```
>>> wks.update_acell('A1', '42') # this could be 'a1' as well
<Cell R1C1 "I'm cell A1">
```

update_cell (*row, col, val*)

Sets the new value to a cell.

Parameters

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

update_cells (*cell_list*)

Updates cells in batch.

Parameters **cell_list** – List of a *Cell* objects to update.

updated

Updated time in RFC 3339 format

class `gsread.Cell` (*worksheet, element*)

An instance of this class represents a single cell in a *worksheet*.

col

Column number of the cell.

row

Row number of the cell.

value = None

Value of the cell.

Exceptions

exception `gspread.GSreadException`

A base class for gspread's exceptions.

exception `gsread.AuthenticationError`

An error during authentication process.

exception `gsread.SpreadsheetNotFound`

Trying to open non-existent or inaccessible spreadsheet.

exception `gsread.WorksheetNotFound`

Trying to open non-existent or inaccessible worksheet.

exception `gsread.NoValidUrlKeyFound`

No valid key found in URL.

exception `gsread.UpdateCellError`

Error while setting cell's value.

exception `gsread.RequestError`

Error while sending API request.

Internal Modules

Following modules are for internal use only.

4.1 `gsread.httpsession`

This module contains a class for working with http sessions.

class `gsread.httpsession.HTTPSession` (*headers=None*)
Handles HTTP activity while keeping headers persisting across requests.
Parameters **headers** – A dict with initial headers.

4.2 `gsread.urls`

This module is Google API url patterns storage.

`gsread.urls.construct_url` (*feedtype=None, obj=None, visibility='private', projection='full', spreadsheet_id=None, worksheet_id=None, cell_id=None, worksheet_version=None*)
Constructs URL to be used for API request.

Indices and tables

- `genindex`
- `modindex`
- `search`

g

`gsread`, [1](#)
`gsread.httpsession`, [13](#)
`gsread.urls`, [13](#)

A

`acell()` (gsread.Worksheet method), 6
`add_cols()` (gsread.Worksheet method), 6
`add_rows()` (gsread.Worksheet method), 6
`add_worksheet()` (gsread.Spreadsheet method), 5
`append_row()` (gsread.Worksheet method), 6
`AuthenticationError`, 11
`authorize()` (in module gsread), 3

C

`Cell` (class in gsread), 8
`cell()` (gsread.Worksheet method), 6
`Client` (class in gsread), 3
`col` (gsread.Cell attribute), 8
`col_count` (gsread.Worksheet attribute), 6
`col_values()` (gsread.Worksheet method), 6
`construct_url()` (in module gsread.urls), 13

D

`del_worksheet()` (gsread.Spreadsheet method), 5

E

`export()` (gsread.Worksheet method), 6

F

`find()` (gsread.Worksheet method), 7
`findall()` (gsread.Worksheet method), 7

G

`get_addr_int()` (gsread.Worksheet method), 7
`get_all_records()` (gsread.Worksheet method), 7
`get_all_values()` (gsread.Worksheet method), 7
`get_int_addr()` (gsread.Worksheet method), 7
`get_worksheet()` (gsread.Spreadsheet method), 5
`gsread` (module), 1
`gsread.httpsession` (module), 13
`gsread.urls` (module), 13
`GSpreadException`, 11

H

`HTTPSession` (class in gsread.httpsession), 13

I

`id` (gsread.Worksheet attribute), 7
`insert_row()` (gsread.Worksheet method), 7

L

`login()` (gsread.Client method), 3
`login()` (in module gsread), 3

N

`NoValidUrlKeyFound`, 11

O

`open()` (gsread.Client method), 3
`open_by_key()` (gsread.Client method), 4
`open_by_url()` (gsread.Client method), 4
`openall()` (gsread.Client method), 4

R

`range()` (gsread.Worksheet method), 8
`RequestError`, 11
`resize()` (gsread.Worksheet method), 8
`row` (gsread.Cell attribute), 8
`row_count` (gsread.Worksheet attribute), 8
`row_values()` (gsread.Worksheet method), 8

S

`sheet1` (gsread.Spreadsheet attribute), 5
`Spreadsheet` (class in gsread), 5
`SpreadsheetNotFound`, 11

T

`title` (gsread.Worksheet attribute), 8

U

`update_acell()` (gsread.Worksheet method), 8
`update_cell()` (gsread.Worksheet method), 8

`update_cells()` (`gsread.Worksheet` method), 8
`UpdateCellError`, 11
`updated` (`gsread.Worksheet` attribute), 8

V

`value` (`gsread.Cell` attribute), 9

W

`Worksheet` (class in `gsread`), 6
`worksheet()` (`gsread.Spreadsheet` method), 5
`WorksheetNotFound`, 11
`worksheets()` (`gsread.Spreadsheet` method), 6