EZGmail Documentation

Al Sweigart

Oct 10, 2022

Contents

1 Installation 3

2 Quickstart 5

3 API 9

Python Module Index 13

Index 15

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

CHAPTER 1

Installation

EZGmail can be installed from PyPI using *pip*:

```
pip install ezgmail
```

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

```
pip3 install ezgmail
```

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

```
pip install --user ezgmail
```

You must also sign up for a Gmail account at https://gmail.com/. Then, go to https://developers.google.com/gmail/api/quickstart/python/, click the **Enable the Gmail API** button on that page, and fill out the form that appears.

After you've filled out the form, the page will present a link to the *credentials.json file*, which you'll need to download and place in the same folder as your *.py* file. The *credentials.json* file contains the Client ID and Client Secret information, which you should treat the same as your Gmail password and not share with anyone else.

Make sure you set your current working directory to the same folder that *credentials.json* is in and that you're connected to the internet. Running **import ezgmail** (which calls the **ezgmail.init()** function) will open your browser to a Google sign-in page.

Enter your Gmail address and password. The page may warn you "This app isn't verified," but this is fine; click **Advanced** and then **Go to Quickstart (unsafe)**. (If you write Python scripts for others and don't want this warning appearing for them, you'll need to learn about Google's app verification process, which is beyond the scope of this documentaiton.) When the next page prompts you with "Quickstart wants to access your Google Account," click **Allow** and then close the browser.

A *token.json* file will be generated to give your Python scripts access to the Gmail account you entered. The browser will only open to the login page if it can't find an existing token.json file. With *credentials.json* and *token.json*, your Python scripts can send and read emails from your Gmail account without requiring you to include your Gmail password in your source code.

CHAPTER 2

Quickstart

To log in to your Gmail account, simple import EZGmail after setting up the *credentials.json* and *token.json* files:

```
>>> import ezgmail
```

If you'd like to log in as a different user (based on different *credentials.json* and *token.json* files), call `ezgmail.init()` and pass the filenames for the credential and token files you generated.

```
>>> ezgmail.init(tokenFile='token.json', credentialsFile='credentials.json')
```

At this point, the `EMAIL_ADDRESS` global variable will contain a string of the logged in user, and `LOGGED_IN` is set to `True`. (Otherwise, they are set to `None` and `False`, respectively.)

```
>>> ezgmail.EMAIL_ADDRESS
'test.sweigart@gmail.com'
>>> ezgmail.LOGGED_IN
True
```

To send an email from the logged in account, call `ezgmail.send()`:

```
>>> ezgmail.send('recipient@example.com', 'Subject Line', 'Hello, this is the
body of the message.')
```

To get a list of all the email threads of unread emails, call `ezgmail.unread()`. This returns a list of `GmailThread` objects. Each `GmailThread` contains a `messages` attribute which is a list of `GmailMessage` objects. Each `GmailMessage` object has attributes `sender`, `timestamp`, `subject`, `body`:

```
>>> ezgmail.unread()
[<GmailThread len=4 snippet='Quickstart was granted access to your Google Account␣
↪test.sweigart@gmail.com If you did not grant access, you should check this activity␣
↪and secure your account. Check activity You received this email'>]

>>> unreadThreads = ezgmail.unread()

>>> unreadThreads
[<GmailThread len=4 snippet='Quickstart was granted access to your Google Account␣
↪test.sweigart@gmail.com If you did not grant access, you should check this activity␣
↪and secure your account. Check activity You received this email'>]
```

(continues on next page)

(continued from previous page)

```
>>> thread = unreadThreads[0]

>>> thread.messages
[<ezgmail.GmailMessage object at 0x000002B7F1FD8A90>, <ezgmail.GmailMessage object at␣
↪0x000002B7F1FD80B8>, <ezgmail.GmailMessage object at 0x000002B7F1FD8A58>, <ezgmail.
↪GmailMessage object at 0x000002B7F1FD8160>]

>>> thread.messages[0].sender
'Alice Smith <alice@example.com>'

>>> thread.messages[0].timestamp
datetime.datetime(2019, 11, 11, 10, 7, 55)

>>> thread.messages[0].subject
'Good to meet you!'

>>> thread.messages[0].body
"It was good to meetup!\r\n\r\n\r\nWe should hang out again soon.\r\n"
```

You can call `ezgmail.recent()` to get the recent email threads as a list of `GmailThread` objects.

You can search for messages by calling `ezgmail.search('query string')`, which also returns a list of `GmailThread` objects. The query string is exactly the same as you would type in the Gmail search box, and you can use the search operatives for it too:

- `ezgmail.search('label:UNREAD')`
- `ezgmail.search('from:al@inventwithpython.com')`
- `ezgmail.search('subject:hello')`
- `ezgmail.search('has:attachment')`

More search operatives are described at https://support.google.com/mail/answer/7190?hl=en

By default, EZGmail sends messages as plaintext. You can send HTML emails by passing `'html'` for the `mimeSubtype` parameter in `send()`. (By default, this parameter is set to `'plain'`.) This email has "Hello" appear in bold and "body" appear italicized:

```
>>> ezgmail.send('recipient@example.com', 'Subject Line', '<strong>Hello</strong>,␣
↪this is the <em>body</em> of the message.', mimeSubtype='html')
```

Accessing an email or thread doesn't mark it as unread automatically. You must do that yourself by calling the `markAsRead()` method of the `GmailThread` or `GmailMessage` object. (There is also a corresponding `markAsUnread()` function.) You can also call `ezgmail.markAsRead()` and pass it a list of `GmailThread` or `GmailMessage` objects.

```
>>> import ezgmail
>>> unreadThreads = ezgmail.unread()
>>> ezgmail.markAsRead(unreadThreads) # Marks all the GmailThread objects in the␣
↪unreadThreads list as read.
>>> # Or you can do:
>>> for unreadThread in unreadThreads:
...     unreadThread.markAsRead() # Mark the individual GmailThread objects as read.
```

These two functions make add/remove the `'UNREAD'` label using EZGmail's `addLabel()` and `removeLabel()` functions:

```
>>> import ezgmail
>>> unreadThreads = ezgmail.unread()
>>> ezgmail.removeLabel(unreadThreads, 'UNREAD') # Also marks threads as read.
>>> ezgmail.addLabel(unreadThreads, 'UNREAD') # Marks them as unread again.
>>> # Or you can do:
>>> for unreadThread in unreadThreads:
...     unreadThread.removeLabel(unreadThreads, 'UNREAD') # Mark the individual␣
↪GmailThread objects as read.
```

(Currently EZGmail doesn't have functions for adding/deleting/managing custom labels.)

The `trash()` method deletes the message or messages in a `GmailMessage` or `GmailThread` object:

```
>>> import ezgmail
>>> threads = ezgmail.search('mancala')
>>> threads[0].trash()  # Move the entire first thread to the Trash folder.
```

To view the attachments of an email, look at the `GmailMessage` object's `attachments` dictionary. The keys are the filenames of the attachments. You can either call the `downloadAttachment()` or `downloadAllAttachments()` methods:

```
>>> import ezgmail
>>> threads = ezgmail.search('See the attached files')
>>> threads[0].messages[0].attachments
>>> import pprint
>>> pprint.pprint(threads[0].messages[0].attachments)
{'a.png': {'id': 'ANGjdJ8eLDbjBpFTfvpuQ2HfR_iwp59XLUIl-
↪IHW8eJcexMsxBYoPCZAXcX16rnqcbJZTknF5r3GmnM1W9n4vAE1oiVgUa4S4zBmNs7rd5PzFwLjO2vU3hp3_
↪9SEZv-KBqVxi9nuNjarxhFqp3mxw6E5mqEYmFOYtT7Gx6CZbLaJuUox9GaWu-W9B4-
↪XPDjwKkEfCdJ21FlOl-CsC6isZgD2Vh-ghh1haZN_2sifccznLv61ZW_
↪KmqPKFcV1j7cXMQVqWU7bkgdH8do4Msc3QsG2ly_PNRid4-7gihsXaLI1ko_
↪j3LSvsoLHFP3edhxh6YKQ2OdMhyZh5lqjmfT1TXgSo7hY16P_ScDO5MnWvmKscf_Hm5y5D4DHfwOq4--
↪Otivoq2WVkVucVUJBkAoB',
           'size': 833609},
 'b.png': {'id': 'ANGjdJ_
↪WYMmPmy2Dd2VBgvVoLAd1p3ARxGXKIzVfKqAiLhvKSBmEowYqFCdHbMJYlDZy4IWBGLg0eQCllMI0icqamM7vfMxBW2irJVogL
↪LHchowO56vTdKLRRsaJ2gfW0esz3cDFZzvthdR4wyBKEIeCJv7OJmFiaJIRf9f1KmFfKPLo9GZSyD2RMXdd6Qa2M3uN9pgT6sZ
↪OQx3e6aNDAKWh5GCeSiuIt_Z7GsDCdzVJjakMJx5FRFhp5zIck0p04AHnYhKfy1BipWmf7G-
↪DAKzgJHAhFimBVUIBeFsHrqEGxDlevD7lK4ZBeb8cluSmYyEsRkSPSMYMlp-
↪x1GVw25gqMnMVkGMKPfwj38iB',
           'size': 335911}}
>>> threads[0].messages[0].downloadAttachment('a.png') # Download to current working␣
↪directory.
>>> threads[0].messages[0].downloadAttachment('b.png', '/path/to/save/in')
>>> threads[0].messages[0].downloadAllAttachments() # Easier way to save all␣
↪attachments.
```

The API section contains complete documentation.

CHAPTER 3

API

EZGmail - A Pythonic interface to the Gmail API that actually works as of October 2022.

exception ezgmail.**EZGmailException**

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

exception ezgmail.**EZGmailTypeError**

The EZGmail module equivalent of ValueError. This exception is raised when a parameter of an incorrect type is passed to an EZGmail function.

exception ezgmail.**EZGmailValueError**

The EZGmail module equivalent of ValueError. This exception is raised when a parameter of an incorrect value (but not necessarily an incorrect type) is passed to an EZGmail function.

class ezgmail.**GmailMessage**(*messageObj*)

Represents a Gmail messages. These objects are returned by the users.messages.get() API call. They contain all the header/subject/body information of a single email.

The `sender` attribute has a string like `'Google <no-reply@accounts.google.com>'`.

The `recipient` attribute has a string like `al@inventwithpython.com'`.

The `subject` attribute contains a string of the subject line.

The `body` attribute contains text up to the quoted "reply" text that begins with "On Sun, Jan 1, 2018 at 12:00 PM al@inventwithpython.com wrote:" part.

The `originalBody` attribute contains the full message body.

The `timestamp` attribute contains a `datetime.datetime` object of the internal message creation timestamp (epoch ms), which determines ordering in the inbox.

The `snippet` attribute contains a string of up to the first 200 characters of the body.

These attributes are based on the Gmail API: https://developers.google.com/gmail/api/v1/reference/users/messages

addLabel(*label*)

Add the label `label` to every message in this thread.

downloadAllAttachments(*downloadFolder='.'*, *overwrite=True*)
Download all of the attachments in this message to the local folder `downloadFolder`. If `overwrite` is `True`, existing local files will be overwritten by attachments with the same filename.

downloadAttachment(*filename*, *downloadFolder='.'*, *duplicateIndex=0*)
Download the file attachment in this message with the name `filename` to the local folder `downloadFolder`. If there are multiple attachments with the same name, `duplicateIndex` needs to be passed to specify which attachment to download.

markAsRead()
Mark this message as read. (This does the same thing as removing the UNREAD label from the message.)

markAsUnread()
Mark this message as unread. (This does the same thing as adding the UNREAD label to the message.)

removeLabel(*label*)
Remove the label `label` from every message in this thread, if it's there.

reply(*body*, *attachments=None*, *cc=None*, *bcc=None*, *mimeSubtype='plain'*)
Like the send() function, but replies to the last message in this thread.

replyAll(*body*, *attachments=None*, *cc=None*, *bcc=None*, *mimeSubtype='plain'*)
Like the send() function, but replies to the last message in this thread.

trash()
Move this message to the Trash folder. It will be automatically removed in 30 days.

class ezgmail.**GmailThread**(*threadObj*)
Represents a thread of Gmail messages. These objects are returned by the users.threads.get() API call. They contain references to a list of GmailMessage objects.

addLabel(*label*)
Add the label `label` to every message in this thread.

latestTimestamp()
The

markAsRead()
Mark every message in this thread as read. (This does the same thing as removing the UNREAD label from the messages.)

markAsUnread()
Mark every message in this thread as unread. (This does the same thing as adding the UNREAD label to the messages.)

messages
The GmailMessage objects of the emails in this thread, starting from the oldest at index 0 to the most recent.

removeLabel(*label*)
Remove the label `label` from every message in this thread, if it's there.

senders()
Returns a list of strings of the senders in this thread, from the oldest at index 0 to the most recent.

text
A list of strings, where each string is the message of a single email in this thread of emails, starting from the oldest at index 0 to the most recent.

trash()
Move every message in this thread to the Trash folder. It will be automatically removed in 30 days.

`ezgmail.`**`init`**`(`*userId='me'*, *tokenFile='token.json'*, *credentialsFile='credentials.json'*, *_raiseException=True*`)`

This function must be called before any other function in EZGmail (and is automatically called by them anyway, so you don't have to explicitly call this yourself).

This function populates the `SERVICE_GMAIL` global variable used in all Gmail API calls. It also populates `EMAIL_ADDRESS` with a string of the Gmail account's email address (and sets the global `LOGGED_IN` to `True`). This account is determined by the *credentials.json* file, downloaded from Google, and *token.json*. If the `tokenFile` file hasn't been generated yet, this function will open the browser to a page to let the user log in to the Gmail account that this module will use.

If you want to switch to a different Gmail account, call this function again with a different `tokenFile` and `credentialsFile` arguments.

`ezgmail.`**`recent`**`(`*maxResults=25*, *userId='me'*`)`

Return a list of `GmailThread` objects for the most recent emails. Essentially a wrapper for `search()`.

First index is the most recent.

`ezgmail.`**`removeQuotedParts`**`(`*emailText*`)`

Returns the text in `emailText` up to the quoted "reply" text that begins with "On Sun, Jan 1, 2018 at 12:00 PM al@inventwithpython.com wrote:" part.

`ezgmail.`**`search`**`(`*query*, *maxResults=25*, *userId='me'*`)`

Returns a list of GmailThread objects that match the search query.

The `query` string is exactly the same as you would type in the Gmail search box, and you can use the search operatives for it too:

- label:UNREAD
- from:al@inventwithpython.com
- subject:hello
- has:attachment

More are described at https://support.google.com/mail/answer/7190?hl=en

`ezgmail.`**`send`**`(`*recipient*, *subject*, *body*, *attachments=None*, *sender=None*, *cc=None*, *bcc=None*, *mimeSubtype='plain'*, *_threadId=None*`)`

Sends an email from the configured Gmail account.

Note that the `sender` argument seems to be ignored by Gmail, which uses the account's actual email address.

TODO - Add additional details to this docstring.

`ezgmail.`**`summary`**`(`*gmailObjects*, *printInfo=True*`)`

Prints out a summary of the `GmailThread` or `GmailMessage` in the `gmailObjects` list.

`ezgmail.`**`unread`**`(`*maxResults=25*, *userId='me'*`)`

Return a list of `GmailThread` objects for unread emails. Essentially a wrapper for `search()`.

Python Module Index

e

ezgmail, 9

Index

A

addLabel() (*ezgmail.GmailMessage method*), 9
addLabel() (*ezgmail.GmailThread method*), 10

D

downloadAllAttachments() (*ezgmail.GmailMessage method*), 10
downloadAttachment() (*ezgmail.GmailMessage method*), 10

E

ezgmail (*module*), 9
EZGmailException, 9
EZGmailTypeError, 9
EZGmailValueError, 9

G

GmailMessage (*class in ezgmail*), 9
GmailThread (*class in ezgmail*), 10

I

init() (*in module ezgmail*), 10

L

latestTimestamp() (*ezgmail.GmailThread method*), 10

M

markAsRead() (*ezgmail.GmailMessage method*), 10
markAsRead() (*ezgmail.GmailThread method*), 10
markAsUnread() (*ezgmail.GmailMessage method*), 10
markAsUnread() (*ezgmail.GmailThread method*), 10
messages (*ezgmail.GmailThread attribute*), 10

R

recent() (*in module ezgmail*), 11
removeLabel() (*ezgmail.GmailMessage method*), 10
removeLabel() (*ezgmail.GmailThread method*), 10
removeQuotedParts() (*in module ezgmail*), 11
reply() (*ezgmail.GmailMessage method*), 10
replyAll() (*ezgmail.GmailMessage method*), 10

S

search() (*in module ezgmail*), 11
send() (*in module ezgmail*), 11
senders() (*ezgmail.GmailThread method*), 10
summary() (*in module ezgmail*), 11

T

text (*ezgmail.GmailThread attribute*), 10
trash() (*ezgmail.GmailMessage method*), 10
trash() (*ezgmail.GmailThread method*), 10

U

unread() (*in module ezgmail*), 11