Gmail Client Documentation

Release 0.0.4

Wilberto Morales

February 13, 2016

Contents

A Pythonic interface to Google's GMail, with all the tools you'll need. Search, read and send multipart emails, archive, mark as read/unread, delete emails, and manage labels.

CHAPTER 1

Installation

Install using pip

```
pip install gmail_client
```

CHAPTER 2

Features

1. Search emails
2. Read emails
3. Emails: label, archive, delete, mark as read/unread/spam, star
4. Manage labels

CHAPTER 3

Basic usage

To start, import the *gmail_client* library.

```
import gmail_client
```

3.1 Authenticating sessions

To easily get up and running:

```
import gmail_client

g = gmail_client.login(username, password)
```

Which will automatically log you into a GMail account. This is actually a shortcut for creating a new *Gmail* object:

```
from gmail_client import Gmail

g = Gmail()
g.login(username, password)
# play with your gmail...
g.logout()
```

You can also check if you are logged in at any time:

```
g = gmail_client.login(username, password)
g.logged_in # should be True, AuthenticationError if login fails
```

3.2 OAuth authentication

If you have already received an OAuth2 access token from Google for a given user, you can easily log the user in. (Because OAuth 1.0 usage was deprecated in April 2012, this library does not currently support its usage).

```
gmail = gmail_client.authenticate(username, access_token)
```

3.3 Filtering emails

Get all messages in your inbox:

```
g.inbox().mail()
```

Get messages that fit some criteria:

```
g.inbox().mail(after=datetime.date(2013, 6, 18), before=datetime.date(2013, 8, 3))
g.inbox().mail(on=datetime.date(2009, 1, 1)
g.inbox().mail(sender="myfriend@gmail.com") # "from" is reserved, use "fr" or "sender"
g.inbox().mail(to="directlytome@gmail.com")
```

Combine flags and options:

```
g.inbox().mail(unread=True, sender="myboss@gmail.com")
```

Browsing labeled emails is similar to working with your inbox.

```
g.mailbox('Urgent').mail()
```

Every message in a conversation/thread will come as a separate message.

```
g.inbox().mail(unread=True, before=datetime.date(2013, 8, 3) sender="myboss@gmail.com")
```

3.4 Working with emails

Important: calls to 'mail()' will return a list of empty email messages (with unique IDs). To work with labels, headers, subjects, and bodies, call 'fetch()' on an individual message. You can call mail with 'prefetch=True', which will fetch the bodies automatically.

```
unread = g.inbox().mail(unread=True)
print unread[0].body # None

unread[0].fetch()
print unread[0].body
```

Also calling *Message.fetch* twice is lazy in order to save requests. You can force fetch:

```
unread.forced_fetch()
```

Mark news past a certain date as read and archive it:

```
emails = g.inbox().mail(before=datetime.date(2013, 4, 18), sender="news@nbcnews.com")
for email in emails:
    email.read()
    email.archive()
```

Delete all emails from a certain person:

```
emails = g.inbox().mail(sender="junkmail@gmail.com")
for email in emails:
    email.delete()
```

You can use also *label* method instead of *mailbox*:

```
g.label("Faxes").mail()
```

Add a label to a message:

```
email.add_label("Faxes")
```

Download message attachments:

```
for attachment in email.attachments:
    print 'Saving attachment: ' + attachment.name
    print 'Size: ' + str(attachment.size) + ' KB'
    attachment.save('attachments/' + attachment.name)
```

3.5 Roadmap

- Write tests
- Better label support
- Moving between labels/mailboxes
- Intuitive thread fetching & manipulation
- Sending mail via Google's SMTP servers (for now, check out https://github.com/paulchakravarti/gmail-sender)

CHAPTER 4

Authors

This is a fork of Charlie Guo's gmail library. Heavily inspired by Kriss "nu7hatch" Kowalik's GMail for Ruby library.

1. Wilberto Morales
2. Brian Everett Peterson

class gmail_client.**Gmail**

class gmail_client.**Mailbox**(*gmail*, *name='INBOX'*)
A Mailbox object.

class gmail_client.**Message**(*mailbox*, *uid*)

class gmail_client.**Attachment**(*name*, *content_type*, *content*)
Attachments are files sent in the email.

class gmail_client.**GmailException**
The base exception in the library. Also raised when there was an ambiguous exception that occurred while handling a request.

class gmail_client.**ConnectionError**
A Connection error occurred.

class gmail_client.**AuthenticationError**
Gmail Authentication failed.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

g

A

C

G

M