aiofb Documentation

Release 0.1.4

Tehamalab

Jun 09, 2021

Contents:

CHAPTER 1

aiofb

A thin asynchronous Python wrapper for Facebook graph API.

This library requires Python 3.5+

1.1 Installation

Using pip

```
$ pip install aiofb
```

1.2 Basic usage

Example

```
import asyncio
import aiofb

# initialize Graph API
fb = aiofb.GraphAPI(access_token='YOUR_ACCESS_TOKEN')

# Get an event loop
loop = asyncio.get_event_loop()

# Get results
data = loop.run_until_complete(fb.get('/{some-endpoint}'))
```

CHAPTER 2

Installation

2.1 Stable release

To install aiofb, run this command in your terminal:

```
$ pip install aiofb
```

This is the preferred method to install aiofb, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for aiofb can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/tehamalab/aiofb
```

Or download the tarball:

```
$ curl  -OL https://github.com/tehamalab/aiofb/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

To use aiofb in a project:

```
import aiofb
```

CHAPTER 4

aiofb

4.1 aiofb package

4.1.1 Submodules

4.1.2 aiofb.api module

class aiofb.api.**GraphAPI**(*access_token=None*, *session=None*, *timeout=10*)

Bases: object

A Facebook Graph API wrapper.

https://developers.facebook.com/docs/graph-api/

ROOT_URL = 'https://graph.facebook.com'

URL = 'https://graph.facebook.com/v3.0'

VERSION = '3.0'

access_token = None

str: API access token

async_timeout = None

int: timeout for HTTP requests

get(*path*, *session=None*, ***kwargs*)

Make a HTTP GET request to the API.

A wrapper to *request()* for GET requests.

post(*path*, *session=None*, ***kwargs*)

Make a HTTP POST request to the API.

A wrapper to *request()* for POST requests.

request(*method*, *path*, *session=None*, ***kwargs*)

Make an HTTP request to the API

method [str] HTTP method

path [str] API endpoint

session [ClientSession, optional] An aiohttp.ClientSession to be used for making the request.

****kwargs** Keyword arguments to be passed to aiohttp request. For more info check http://aiohttp.readthedocs.io/en/stable/client_reference.html#aiohttp.ClientSession.request

session = None
aiohttp.ClientSession: Aiohttp session.

class aiofb.api.**Messenger**(*access_token=None*, *session=None*, *timeout=10*)
Bases: *aiofb.api.GraphAPI*

Messenger Platform API wrapper.

This class provides some additional for accessing Messenger API a bit more conveniently

DEFAULT_USER_PROFILE_FIELDS = ['name', 'first_name', 'last_name', 'profile_pic']
list of str: Default user profile properties to be requested

get_user_profile(*psid*, *fields=None*, *session=None*)
Retrieve user profile information using PSID.

https://developers.facebook.com/docs/messenger-platform/identity/user-profile

psid [str] User PSID

fields [list] List of field to be retieved. If not provided `Messenger.DEFAULT_USER_PROFILE_FIELDS` will be used.

pass_thread_control(*data*, *session=None*)
Pass thread control from your app to another app.

https://developers.facebook.com/docs/messenger-platform/reference/handover-protocol/pass-thread-control

data [dict] Data for the handover.

send_message(*data*, *session=None*)
Send messages to user

This may include text, attachments, structured message templates, sender actions, and more.

https://developers.facebook.com/docs/messenger-platform/reference/send-api

data [dict] Message data.

take_thread_control(*data*, *session=None*)
Take thread control from another app.

https://developers.facebook.com/docs/messenger-platform/handover-protocol/take-thread-control

data [dict] Data for the handover.

update_profile(*data*, *session=None*)
Update bot's Messenger profile properties.

Sets the values of one or more Messenger Profile properties. Only properties set in the request body will be overwritten.

To set or update Messenger Profile properties you must have the 'Administrator' role for the Page associated with the bot.

https://developers.facebook.com/docs/graph-api/reference/page/messenger_profile#post

data [dict] data for the update

4.1.3 aiofb.exceptions module

exception aiofb.exceptions.**GraphAPIException**(*message=''*, *response=None*)
Bases: Exception

4.1.4 Module contents

Top-level package for aiofb.

CHAPTER 5

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/tehamalab/aiofb/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

5.1.4 Write Documentation

aiofb could always use more documentation, whether as part of the official aiofb docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/tehamalab/aiofb/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *aiofb* for local development.

1. Fork the *aiofb* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/aiofb.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv aiofb
$ cd aiofb/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 aiofb tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/tehamalab/aiofb/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_aiofb
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER 6

Credits

6.1 Development Lead

- Tehamalab <developers@tehamalab.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

7.1 0.1.0 (2018-04-06)

- Packege created.

7.2 0.1.1 (2018-05-17)

- Clean up
- First release on PyPI

7.3 0.1.2 (2018-10-02)

- Return raw python data decoded from json response instead of `aiohttp.ClientResponse` object.

7.4 0.1.3 (2019-01-11)

- Change default Messenger user profile fields to name,first_name,last_name and profile_pic to reflect new Messenger API policy.

7.5 0.1.3 (2019-02-10)

- Add method for taking Messebger thread control (`messenger.take_thread_control(data, session=None)`)

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index