Twitter Bot Utils Documentation

Release 0.14.0

Neil Freeman

Apr 15, 2023

Contents

The goal of Twitter Bot Utils is to ease the creation of creating Twitter bots by abstracting away the repetition involved in dealing with keys.

Twitter bot utils is somewhat opinionated: it assumes that you want to create command line tools, and that you can store your keys in a basic text configuration files. See *Hello World* for a basic run-through.

Some additional documentation is in the readme.

This package is intended to assist with the creation of bots for artistic or personal projects. Don't use it to spam or harrass people.

CHAPTER 1

Hello World

The first step of any bot is to set up an app and a new account. Those steps are an exercise for the reader.

Twitter Bot Utils is opinionated about one thing: it wants you to store authentication keys in a file called `~/bots.yaml` or `~/bots.json`. (It's actually not that opinionated about where the file goes, read on.)

If you're using a YAML file, it should look like this:

```
apps:
    my_app_name:
        consumer_key: LONGSTRINGOFLETTERS-ANDNUMBERS
        consumer_secret: LETTERSANDNUMBERS
users:
    # twitter screen_name
    MyBotName:
        key: LONGSTRINGOFLETTERS-ANDNUMBERS
        secret: LETTERSANDNUMBERS
        # The app key should match a key in apps below
        app: my_app_name
```

Wait, we haven't authenticated the account with the app. Let's do that quickly with the Twitter Bot Utils `twitter-auth` command:

```
$ twitter-auth --app my_app_name
https://api.twitter.com/oauth/authorize?oauth_token=dWXqSAAAAAAALgurAAABUuIOe0c
Please visit this url, click "Authorize app" and enter in the PIN:
>
```

Now visit the URL in your favorite browser, authorize the app, and you'll be rewarded with key and secret, which you can place in `bots.yaml`.

Next, create a python file called `my_twitter_bot.py` that looks like this:

```
import argparse
import twitter_bot_utils as tbu

```

(continues on next page)

(continued from previous page)

```
def main():
    parser = argparse.ArgumentParser(description='my twitter bot')
    tbu.args.add_default_args(parser, version='1.0')

    args = parser.parse_args()
    api = tbu.api.API(args.user)

    if not args.dry_run:
        api.update_status('Hello World!')
        api.logger.info('I just tweeted!')

if __name__ == '__main__':
    main()
```

On the command line, this will create a full-fledged app that will have lots of tricks:

```
$ python my_twitter_bot.py --help
usage: my_twitter_bot.py [-h] [-c PATH] [-n] [-v] [-q] [-V] [-u screen_name]

my twitter bot

optional arguments:
  -h, --help            show this help message and exit
  -c PATH, --config PATH
                        bots config file (json or yaml)
  -n, --dry-run         Don't actually do anything
  -v, --verbose         Run talkatively
  -q, --quiet           Run quietly
  -V, --version         show program's version number and exit
```

To tweet, run this:

```
$ python my_twitter_bot.py -u MyBotName
I just tweeted!
```

Now you can go ahead and add this command to `cron`, and you're good to go!

1.1 Another approach

Create the `bots.yaml` file as above, but when creating your bot, just set it to print a tweet:

```
def main():
    print('This is a tweet!')

 if __name__ == '__main__':
     main()
```

Now, pipe your scripts output to the `tbu post` command:

```
$ python3 my_twitter_bot.py | tbu post MyBotName
```

CHAPTER 2

API

2.1 api

The `api.API` object is a wrapper around `tweepy.API` with some additional methods for tracking keys.

2.2 args

The `args` modules provides short-cuts for creating command-line tools for Twitter bots.

See the *Hello World* for a fleshed-out example.

2.3 helpers

These helpers are useful for interacting with the metadata in a Tweepy tweet object

CHAPTER 3

Command line tools

Twitter Bot Utils comes with the `tbu` command line tool, which has several subcommands:

- tbu auth
- tbu follow
- tbu like
- tbu post

3.1 tbu auth

```
usage: tbu auth [-h] [-c file] [--app app] [-s] [--consumer-key key]
                [--consumer-secret secret] [-V]

Authorize an account with a twitter application.

optional arguments:
  -h, --help            show this help message and exit
  -c file               config file
  --app app             app name in config file
  -s, --save            Save details to config file
  --consumer-key key    consumer key (aka consumer token)
  --consumer-secret secret
                        consumer secret
  -V, --version         show program's version number and exit
```

3.2 tbu follow

```
usage: tbu follow [options] screen_name

automatic following and unfollowing

positional arguments:
  screen_name

optional arguments:
  -h, --help            show this help message and exit
  -U, --unfollow        Unfollow those who don't follow you
  -c PATH, --config PATH
                        bots config file (json or yaml)
  -n, --dry-run         Don't actually do anything
  -v, --verbose         Run talkatively
  -q, --quiet           Run quietly
  -V, --version         show program's version number and exit
```

3.3 tbu like

```
usage: tbu like [options] screen_name

fave/like mentions

positional arguments:
  screen_name

optional arguments:
  -h, --help            show this help message and exit
  -c PATH, --config PATH
                        bots config file (json or yaml)
  -n, --dry-run         Don't actually do anything
  -v, --verbose         Run talkatively
  -q, --quiet           Run quietly
  -V, --version         show program's version number and exit
```

3.4 tbu post

```
usage: tbu post screen_name "update" [options]

Post text to a given twitter account

positional arguments:
  screen_name
  update

optional arguments:
  -h, --help            show this help message and exit
  -m MEDIA_FILE, --media-file MEDIA_FILE
  -c PATH, --config PATH
                        bots config file (json or yaml)
  -n, --dry-run         Don't actually do anything
```

(continues on next page)

(continued from previous page)

```
  -v, --verbose         Run talkatively
  -q, --quiet           Run quietly
```

CHAPTER 4

Index

- genindex
- modindex