
Twython Documentation

Release 3.7.0

Ryan McGrath

Apr 03, 2020

Contents

1	Features	3
2	Usage	5
2.1	Installation	5
2.1.1	Pip or Easy Install	5
2.1.2	Source Code	5
2.2	Starting Out	6
2.2.1	Beginning	6
2.2.2	Authentication	6
2.2.2.1	OAuth 1 (User Authentication)	6
2.2.2.2	OAuth 2 (Application Authentication)	8
2.2.3	The Twython API Table	8
2.2.4	Dynamic Function Arguments	8
2.2.5	What Twython Returns	8
2.3	Basic Usage	9
2.3.1	Authenticated Calls	9
2.3.1.1	OAuth 1	9
2.3.1.2	OAuth 2	9
2.3.2	Searching	10
2.4	Advanced Usage	10
2.4.1	Updating Status with Image	10
2.4.2	Updating Status with Video	11
2.4.3	Posting a Status with an Editing Image	11
2.4.4	Search Generator	11
2.4.5	Manipulate the Request (headers, proxies, etc.)	12
2.4.6	Access Headers of Previous Call	13
2.5	Streaming API	13
2.5.1	Setting Up Your Streamer	13
2.5.2	Filtering Public Statuses	14
2.6	Special Functions	14
2.6.1	Cursor	14
2.6.1.1	The Old Way	14
2.6.1.2	The New Way	14
2.6.1.3	Items vs Pages	15
2.6.2	HTML for Tweet	15
3	Twython API Documentation	17

3.1	Developer Interface	17
3.1.1	Core Interface	17
3.1.2	Streaming Interface	29
3.1.2.1	Streaming Types	31
3.1.3	Exceptions	31
Python Module Index		33
Index		35

Actively maintained, pure Python wrapper for the Twitter API. Supports both normal and streaming Twitter APIs

CHAPTER 1

Features

- **Query data for:**
 - User information
 - Twitter lists
 - Timelines
 - Direct Messages
 - and anything found in [the Twitter API docs](#).
- **Image Uploading:**
 - Update user status with an image
 - Change user avatar
 - Change user background image
 - Change user banner image
- OAuth 2 Application Only (read-only) Support
- Support for Twitter's Streaming API
- Seamless Python 3 support!

2.1 Installation

Information on how to properly install Twython

2.1.1 Pip or Easy Install

Install Twython via `pip`

```
$ pip install twython
```

or, with `easy_install`

```
$ easy_install twython
```

But, hey... *that's up to you.*

2.1.2 Source Code

Twython is actively maintained on GitHub

Feel free to clone the repository

```
git clone git://github.com/ryanmcgrath/twython.git
```

`tarball`

```
$ curl -OL https://github.com/ryanmcgrath/twython/tarball/master
```

`zipball`

```
$ curl -OL https://github.com/ryanmcgrath/twython/zipball/master
```

Now that you have the source code, install it into your site-packages directory

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

So Twython is installed! Now, head over to the [starting out](#) section.

2.2 Starting Out

This section is going to help you understand creating a Twitter Application, authenticating a user, and making basic API calls

2.2.1 Beginning

First, you'll want to head over to <https://apps.twitter.com/> and register an application!

After you register, grab your applications `Consumer Key` and `Consumer Secret` from the application details tab.

Now you're ready to start authentication!

2.2.2 Authentication

Twython offers support for both OAuth 1 and OAuth 2 authentication.

The difference:

- [OAuth 1](#) is for user authenticated calls (tweeting, following people, sending DMs, etc.)
- [OAuth 2](#) is for application authenticated calls (when you don't want to authenticate a user and make read-only calls to Twitter, i.e. searching, reading a public users timeline)

2.2.2.1 OAuth 1 (User Authentication)

Important: Again, if your web app is planning on using interacting with users, this **IS** the authentication type for you. If you're not interested in authenticating a user and plan on making read-only calls, check out the [OAuth 2](#) section.

First, you'll want to import Twython

```
from twython import Twython
```

Now, you'll want to create a Twython instance with your `Consumer Key` and `Consumer Secret`

Obtain Authorization URL

Note: Only pass *callback_url* to *get_authentication_tokens* if your application is a Web Application

Desktop and Mobile Applications **do not** require a *callback_url*

```
APP_KEY = 'YOUR_APP_KEY'
APP_SECRET = 'YOUR_APP_SECRET'

twitter = Twython(APP_KEY, APP_SECRET)
auth = twitter.get_authentication_tokens(callback_url='http://mysite.com/callback')
```

From the *auth* variable, save the *oauth_token_secret* for later use (these are not the final auth tokens). In Django or other web frameworks, you might want to store it to a session variable

```
OAUTH_TOKEN = auth['oauth_token']
OAUTH_TOKEN_SECRET = auth['oauth_token_secret']
```

Send the user to the authentication url, you can obtain it by accessing

```
auth['auth_url']
```

Handling the Callback

Note: If your application is a Desktop or Mobile Application *oauth_verifier* will be the PIN code

After they authorize your application to access some of their account details, they'll be redirected to the callback url you specified in *get_authentication_tokens*

You'll want to extract the *oauth_verifier* from the url.

Django example:

```
oauth_verifier = request.GET['oauth_verifier']
```

Now that you have the *oauth_verifier* stored to a variable, you'll want to create a new instance of Twython and grab the final user tokens

```
twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

final_step = twitter.get_authorized_tokens(oauth_verifier)
```

Once you have the final user tokens, store them in a database for later use!

```
OAUTH_TOKEN = final_step['oauth_token']
OAUTH_TOKEN_SECRET = final_step['oauth_token_secret']
```

2.2.2.2 OAuth 2 (Application Authentication)

Attention: Just a reminder, this authentication type is for when you don't want to authenticate and interact with users and make read-only calls to Twitter

OAuth 2 authentication is 100x easier than OAuth 1. Let's say you *just* made your application and have your Consumer Key and Consumer Secret

First, you'll want to import Twython

```
from twython import Twython
```

Obtain an OAuth 2 Access Token

```
APP_KEY = 'YOUR_APP_KEY'
APP_SECRET = 'YOUR_APP_SECRET'

twitter = Twython(APP_KEY, APP_SECRET, oauth_version=2)
ACCESS_TOKEN = twitter.obtain_access_token()
```

Save ACCESS_TOKEN in a database or something for later use!

Use the Access Token

```
APP_KEY = 'YOUR_APP_KEY'
ACCESS_TOKEN = 'YOUR_ACCESS_TOKEN'

twitter = Twython(APP_KEY, access_token=ACCESS_TOKEN)
```

Now that you have your OAuth 2 access_token, maybe you'll want to perform a *search* or something

2.2.3 The Twython API Table

The Twython package contains a file `endpoints.py` which holds a Mixin of all Twitter API endpoints. This is so Twython's core `api.py` isn't cluttered with 50+ methods.

2.2.4 Dynamic Function Arguments

Keyword arguments to functions are mapped to the functions available for each endpoint in the Twitter API docs. Doing this allows us to be incredibly flexible in querying the Twitter API, so changes to the API aren't held up from you using them by this library.

2.2.5 What Twython Returns

Twython returns native Python objects. We convert the JSON sent to us from Twitter to an object so you don't have to.

Now that you have a little idea of the type of data you'll be receiving, briefed on how arguments are handled, and your application tokens and user oauth tokens (or access token if you're using OAuth 2), check out the *basic usage* section.

2.3 Basic Usage

This section will cover how to use Twython and interact with some basic Twitter API calls

Before you make any API calls, make sure you *authenticated the user* (or *app*)!

Note: All sections on this page will assume you're using a Twython instance

2.3.1 Authenticated Calls

2.3.1.1 OAuth 1

Create a Twython instance with your application keys and the users OAuth tokens

```
from twython import Twython
twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
```

User Information

Documentation: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/get-account-verify_credentials

```
twitter.verify_credentials()
```

Authenticated Users Home Timeline

Documentation: https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-home_timeline

```
twitter.get_home_timeline()
```

Updating Status

This method makes use of dynamic arguments, *read more about them*

Documentation: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-update>

```
twitter.update_status(status='See how easy using Twython is!')
```

2.3.1.2 OAuth 2

Create a Twython instance with your application key and access token

```
from twython import Twython
twitter = Twython(APP_KEY, access_token=ACCESS_TOKEN)
```

2.3.2 Searching

Note: Searching can be done whether you’re authenticated via OAuth 1 or OAuth 2

Documentation: <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets>

```
twitter.search(q='python')
```

Important: To help explain *dynamic function arguments* a little more, you can see that the previous call used the keyword argument `q`, that is because Twitter specifies in their [search documentation](#) that the search call accepts the parameter “`q`”. You can pass multiple keyword arguments. The search documentation also specifies that the call accepts the parameter “`result_type`”

```
twitter.search(q='python', result_type='popular')
```

So, now, you’re pretty well versed on making authenticated calls to Twitter using Twython. Check out the [advanced usage](#) section, for some functions that may be a little more complicated.

2.4 Advanced Usage

This section will cover how to use Twython and interact with some more advanced API calls

Before you make any API calls, make sure you *authenticated the user* (or *app*)!

Note: All sections on this page will assume you’re using a Twython instance

Create a Twython instance with your application keys and the users OAuth tokens

```
from twython import Twython
twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
```

2.4.1 Updating Status with Image

This uploads an image as a media object and associates it with a status update.

```
photo = open('/path/to/file/image.jpg', 'rb')
response = twitter.upload_media(media=photo)
twitter.update_status(status='Checkout this cool image!', media_ids=[response['media_
↪id']])
```

Documentation:

- <https://developer.twitter.com/en/docs/api-reference-index>
- <https://developer.twitter.com/en/docs/media/upload-media/overview>

2.4.2 Updating Status with Video

This uploads a video as a media object and associates it with a status update.

```
video = open('/path/to/file/video.mp4', 'rb')
response = twitter.upload_video(media=video, media_type='video/mp4')
twitter.update_status(status='Checkout this cool video!', media_ids=[response['media_
↪id']])
```

Documentation:

- <https://developer.twitter.com/en/docs/api-reference-index>
- <https://developer.twitter.com/en/docs/media/upload-media/overview>

2.4.3 Posting a Status with an Editing Image

This example resizes an image, then uploads it as a media object and associates it with a status update.

```
# Assume you are working with a JPEG

from PIL import Image
try:
    # Python 3
    from io import StringIO
except ImportError:
    # Python 2
    from StringIO import StringIO

photo = Image.open('/path/to/file/image.jpg')

basewidth = 320
wpercent = (basewidth / float(photo.size[0]))
height = int((float(photo.size[1]) * float(wpercent)))
photo = photo.resize((basewidth, height), Image.ANTIALIAS)

image_io = StringIO.StringIO()
photo.save(image_io, format='JPEG')

# If you do not seek(0), the image will be at the end of the file and
# unable to be read
image_io.seek(0)

response = twitter.upload_media(media=image_io)
twitter.update_status(status='Checkout this cool image!', media_ids=[response['media_
↪id']])
```

2.4.4 Search Generator

So, if you're pretty into Python, you probably know about [generators](#)

That being said, Twython offers a generator for search results and can be accessed by using the following code:

```
from twython import Twython
twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN,
                  OAUTH_TOKEN_SECRET)

results = twitter.cursor(twitter.search, q='python')
for result in results:
    print(result)
```

2.4.5 Manipulate the Request (headers, proxies, etc.)

There are times when you may want to turn SSL verification off, send custom headers, or add proxies for the request to go through.

Twython uses the `requests` library to make API calls to Twitter. `requests` accepts a few parameters to allow developers to manipulate the actual HTTP request.

Here is an example of sending custom headers to a Twitter API request:

```
from twython import Twython

client_args = {
    'headers': {
        'User-Agent': 'My App Name'
    }
}

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                  client_args=client_args)
```

Here is an example of sending the request through proxies:

```
from twython import Twython

client_args = {
    'proxies': {
        'http': 'http://10.0.10.1:8000',
        'https': 'https://10.0.10.1:8001',
    }
}

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                  client_args=client_args)
```

or both (and set a timeout variable):

```
from twython import Twython

client_args = {
    'headers': {
        'User-Agent': 'My App Name'
    },
    'proxies': {
        'http': 'http://10.0.10.1:8000',
        'https': 'https://10.0.10.1:8001',
    }
}
```

(continues on next page)

(continued from previous page)

```

    }
    'timeout': 300,
}

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                  client_args=client_args)

```

2.4.6 Access Headers of Previous Call

There are times when you may want to check headers from the previous call. If you wish to access headers (ex. x-rate-limit-remaining, x-rate-limit-reset, content-type), you'll use the `get_lastfunction_header` method.

```

from twython import Twython

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

twitter.get_home_timeline()
twitter.get_lastfunction_header('x-rate-limit-remaining')

```

So now you can authenticate, update your status (with or without an image), search Twitter, and a few other things! Good luck!

2.5 Streaming API

This section will cover how to use Twython and interact with the Twitter Streaming API.

Streaming Documentation: <https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/streaming-message-types>

Important: The Streaming API requires that you have OAuth 1 authentication credentials. If you don't have credentials, head over to the [authentication section](#) and find out how!

2.5.1 Setting Up Your Streamer

Note: When stream data is sent back to Twython, we send the data through signals (i.e. `on_success`, `on_error`, etc.)

Make sure you import `TwythonStreamer`

```

from twython import TwythonStreamer

```

Now set up how you want to handle the signals.

```

class MyStreamer(TwythonStreamer):
    def on_success(self, data):
        if 'text' in data:

```

(continues on next page)

(continued from previous page)

```

        print(data['text'])

    def on_error(self, status_code, data):
        print(status_code)

        # Want to stop trying to get data because of the error?
        # Uncomment the next line!
        # self.disconnect()

```

More signals that you can extend on can be found in the Developer Interface section under *Streaming Interface*

2.5.2 Filtering Public Statuses

```

stream = MyStreamer(APP_KEY, APP_SECRET,
                    OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
stream.statuses.filter(track='twitter')

```

With the code above, data should be flowing in.

2.6 Special Functions

This section covers methods to are part of Twython but not necessarily connected to the Twitter API.

2.6.1 Cursor

This function returns a generator for Twitter API endpoints that are able to be paginated in some way (either by cursor or since_id parameter)

2.6.1.1 The Old Way

```

from twython import Twython

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

results = twitter.search(q='twitter')
if results.get('statuses'):
    for result in results['statuses']:
        print(result['id_str'])

```

2.6.1.2 The New Way

```

from twython import Twython

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

```

(continues on next page)

(continued from previous page)

```
results = twitter.cursor(twitter.search, q='twitter')
for result in results:
    print(result['id_str'])
```

Another example:

```
results = twitter.cursor(twitter.get_mentions_timeline)
for result in results:
    print(result['id_str'])
```

2.6.1.3 Items vs Pages

By default, the cursor yields one item at a time. If instead you prefer to work with entire pages of results, specify `return_pages=True` as a keyword argument.

```
results = twitter.cursor(twitter.get_mentions_timeline, return_pages=True)
# page is a list
for page in results:
    for result in page:
        print(result['id_str'])
```

2.6.2 HTML for Tweet

This function takes a tweet object received from the Twitter API and returns a string formatted in HTML with the links, user mentions and hashtags replaced.

```
from twython import Twython

twitter = Twython(APP_KEY, APP_SECRET,
                  OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

user_tweets = twitter.get_user_timeline(screen_name='mikehelmick',
                                       include_rts=True)

for tweet in user_tweets:
    tweet['text'] = Twython.html_for_tweet(tweet)
    print(tweet['text'])
```

The above code takes all the tweets from a specific users timeline, loops over them and replaces the value of `tweet['text']` with HTML.

So:

`http://t.co/FCmXyI6VHd` is #cool, lol! @mikehelmick shd #checkitout. Love, @__twython__ \$IBM
`https://t.co/67pwRvY6z9` `http://t.co/N6InAO4B71`

will be replaced with:

```
<a href="http://t.co/FCmXyI6VHd" class="twython-url">google.com</a> is <a href="https://twitter.com/search?q=%23cool" class="twython-hashtag">#cool</a>, lol! <a href="https://twitter.com/mikehelmick" class="twython-mention">@mikehelmick</a> shd <a href="https://twitter.com/search?q=%23checkitout" class="twython-hashtag">#checkitout</a>. Love, <a href="https://twitter.com/__twython__" class="twython-mention">@__twython__</a> <a href="https://twitter.com/?q=%24IBM" class="twython-symbol">$IBM</a> <a href="https://t.co/67pwRvY6z9"
```

```
class="twython-url">github.com</a>    <a    href="http://t.co/N6InAO4B71"    class="twython-  
media">pic.twitter.com/N6InAO4B71</a>
```

Note: When converting the string to HTML we add a class to each HTML tag so that you can manipulate the DOM later on.

- For urls that are replaced we add `class="twython-url "` to the anchor tag
- For media urls that are replaced we add `class="twython-media"` to the anchor tag
- For user mentions that are replaced we add `class="twython-mention"` to the anchor tag
- For hashtags that are replaced we add `class="twython-hashtag"` to the anchor tag
- For symbols that are replaced we add `class="twython-symbol "` to the anchor tag

This function accepts two parameters: `use_display_url` and `use_expanded_url`. By default, `use_display_url` is `True`. Meaning the link displayed in the tweet text will appear as (ex. `google.com`, `github.com`) If `use_expanded_url` is `True`, it overrides `use_display_url`. The urls will then be displayed as (ex. `http://google.com`, `https://github.com`) If `use_display_url` and `use_expanded_url` are `False`, short url will be used (`t.co/xxxxx`)

3.1 Developer Interface

This page of the documentation will cover all methods and classes available to the developer.

Twython, currently, has two main interfaces:

- Twitter's Core API (updating statuses, getting timelines, direct messaging, etc)
- Twitter's Streaming API

3.1.1 Core Interface

```
class twython.Twython (app_key=None, app_secret=None, oauth_token=None,
                       oauth_token_secret=None, access_token=None, token_type='bearer',
                       oauth_version=1, api_version='1.1', client_args=None,
                       auth_endpoint='authenticate')
```

```
__init__ (app_key=None, app_secret=None, oauth_token=None, oauth_token_secret=None,
          access_token=None, token_type='bearer', oauth_version=1, api_version='1.1',
          client_args=None, auth_endpoint='authenticate')
```

Instantiates an instance of Twython. Takes optional parameters for authentication and such (see below).

Parameters

- **app_key** – (optional) Your applications key
- **app_secret** – (optional) Your applications secret key
- **oauth_token** – (optional) When using **OAuth 1**, combined with

oauth_token_secret to make authenticated calls :param **oauth_token_secret**: (optional) When using **OAuth 1** combined with **oauth_token** to make authenticated calls :param **access_token**: (optional) When using **OAuth 2**, provide a valid access token if you have one :param **token_type**: (optional) When using **OAuth 2**, provide your token type. Default: bearer :param **oauth_version**: (optional) Choose which OAuth version to use. Default: 1 :param **api_version**: (optional) Choose which Twitter API version to use. Default: 1.1

Parameters `client_args` – (optional) Accepts some requests Session parameters

and some requests Request parameters. See <http://docs.python-requests.org/en/latest/api/#session-api> and requests section below it for details. [ex. headers, proxies, verify(SSL verification)]

Parameters `auth_endpoint` – (optional) Lets you select which authentication

endpoint will use your application. This will allow the application to have DM access if the endpoint is 'authorize'.

Default: authenticate.

add_list_member (***params*)

Add a member to a list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-members-create>

static construct_api_url (*api_url*, ***params*)

Construct a Twitter API url, encoded, with parameters

Parameters `api_url` – URL of the Twitter API endpoint you are attempting

to construct :param ***params*: Parameters that are accepted by Twitter for the endpoint you're requesting
:rtype: string

Usage:

```
>>> from twython import Twython
>>> twitter = Twython()

>>> api_url = 'https://api.twitter.com/1.1/search/tweets.json'
>>> constructed_url = twitter.construct_api_url(api_url, q='python',
result_type='popular')
>>> print constructed_url
https://api.twitter.com/1.1/search/tweets.json?q=python&result_type=popular
```

create_block (***params*)

Blocks the specified user from following the authenticating user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-blocks-create>

create_favorite (***params*)

Favorites the status specified in the `id` parameter as the authenticating user.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-favorites-create>

create_friendship (***params*)

Allows the authenticating users to follow the user specified in the `id` parameter.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/post-friendships-create>

create_list (***params*)

Creates a new list for the authenticated user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-create>

create_list_members (**params)

Adds multiple members to a list, by specifying a comma-separated list of member ids or screen names.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-members-create_all

create_metadata (**params)

Adds metadata to a media element, such as image descriptions for visually impaired.

Docs: <https://developer.twitter.com/en/docs/media/upload-media/api-reference/post-media-metadata-create>

create_mute (**params)

Mutes the specified user, preventing their tweets appearing in the authenticating user's timeline.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-mutes-users-create>

create_place (**params)

Creates a new place object at the given latitude and longitude.

Docs: <https://dev.twitter.com/docs/api/1.1/post/geo/place>

create_saved_search (**params)

Create a new saved search for the authenticated user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-mutes-users-create>

cursor (function, return_pages=False, **params)

Returns a generator for results that match a specified query.

Parameters **function** – Instance of a Twython function

(Twython.get_home_timeline, Twython.search) :param **params: Extra parameters to send with your request (usually parameters accepted by the Twitter API endpoint) :rtype: generator

Usage:

```
>>> from twython import Twython
>>> twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN,
OAUTH_TOKEN_SECRET)

>>> results = twitter.cursor(twitter.search, q='python')
>>> for result in results:
>>>     print result
```

delete_list (**params)

Deletes the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-destroy>

delete_list_member (**params)

Removes the specified member from the list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-members-destroy>

delete_list_members (**params)

Removes multiple members from a list, by specifying a comma-separated list of member ids or screen names.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-members-destroy_all

destroy_block (**params)

Un-blocks the user specified in the `id` parameter for the authenticating user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-blocks-destroy>

destroy_direct_message (**params)

Destroys the direct message specified in the required `id` parameter

Docs: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/api-reference/delete-message>

destroy_favorite (**params)

Un-favorites the status specified in the `id` parameter as the authenticating user.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-favorites-destroy>

destroy_friendship (**params)

Allows the authenticating user to unfollow the user specified in the `id` parameter.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/post-friendships-destroy>

destroy_mute (**params)

Un-mutes the user specified in the `user` or `id` parameter for the authenticating user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-mutes-users-destroy>

destroy_saved_search (**params)

Destroys a saved search for the authenticating user.

Docs: https://developer.twitter.com/en/docs/tweets/search/api-reference/post-saved_searches-destroy-id

destroy_status (**params)

Destroys the status specified by the required `id` parameter

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-destroy-id>

get (endpoint, params=None, version='1.1')

Shortcut for GET requests via *request*

get_account_settings (**params)

Returns settings (including current trend, geo and sleep time information) for the authenticating user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/get-account-settings>

get_application_rate_limit_status (**params)

Returns the current rate limits for methods belonging to the specified resource families.

Docs: https://developer.twitter.com/en/docs/developer-utilities/rate-limit-status/api-reference/get-application-rate_limit_status

get_authentication_tokens (callback_url=None, force_login=False, screen_name="")

Returns a dict including an authorization URL, `auth_url`, to direct a user to

Parameters

- **callback_url** – (optional) Url the user is returned to after they authorize your app (web clients only)
- **force_login** – (optional) Forces the user to enter their credentials to ensure the correct users account is authorized.
- **screen_name** – (optional) If forced_login is set OR user is not currently logged in, Prefills the username input box of the OAuth login screen with the given value

Return type dict

get_authorized_tokens (*oauth_verifier*)

Returns a dict of authorized tokens after they go through the *get_authentication_tokens* phase.

Parameters **oauth_verifier** – (required) The oauth_verifier (or a.k.a PIN

for non web apps) retrieved from the callback url querystring :rtype: dict

get_available_trends (***params*)

Returns the locations that Twitter has trending topic information for.

Docs: <https://developer.twitter.com/en/docs/trends/locations-with-trending-topics/api-reference/get-trends-available>

get_closest_trends (***params*)

Returns the locations that Twitter has trending topic information for, closest to a specified location.

Docs: <https://developer.twitter.com/en/docs/trends/locations-with-trending-topics/api-reference/get-trends-closest>

get_contributees (***params*)

Returns a collection of users that the specified user can “contribute” to.

Docs: <https://dev.twitter.com/docs/api/1.1/get/users/contributees>

get_contributors (***params*)

Returns a collection of users who can contribute to the specified account.

Docs: <https://dev.twitter.com/docs/api/1.1/get/users/contributors>

get_direct_message (***params*)

Returns a single direct message, specified by an *id* parameter.

Docs: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/api-reference/get-message>

get_direct_messages (***params*)

Returns the 20 most recent direct messages sent to the authenticating user.

Docs: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/api-reference/get-messages>

get_favorites (***params*)

Returns the 20 most recent Tweets favorited by the authenticating or specified user.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-favorites-list>

get_followers_ids (***params*)

Returns a cursored collection of user IDs for every user following the specified user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids>

get_followers_list (**params)

Returns a cursored collection of user objects for users following the specified user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-list>

get_friends_ids (**params)

Returns a cursored collection of user IDs for every user the specified user is following (otherwise known as their “friends”).

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids>

get_friends_list (**params)

Returns a cursored collection of user objects for every user the specified user is following (otherwise known as their “friends”).

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-list>

get_geo_info (**params)

Returns all the information about a known place.

Docs: https://developer.twitter.com/en/docs/geo/place-information/api-reference/get-geo-id-place_id

get_home_timeline (**params)

Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow.

Docs: https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-home_timeline

get_incoming_friendship_ids (**params)

Returns a collection of numeric IDs for every user who has a pending request to follow the authenticating user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friendships-incoming>

get_lastfunction_header (header, default_return_value=None)

Returns a specific header from the last API call This will return None if the header is not present

Parameters **header** – (required) The name of the header you want to get the value of

Most useful for the following header information: x-rate-limit-limit, x-rate-limit-remaining, x-rate-limit-class, x-rate-limit-reset

get_list_members (**params)

Returns the members of the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-members>

get_list_memberships (**params)

Returns the lists the specified user has been added to.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-memberships>

get_list_statuses (**params)

Returns a timeline of tweets authored by members of the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-statuses>

get_list_subscribers (**params)

Returns the subscribers of the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-subscribers>

get_list_subscriptions (**params)

Obtain a collection of the lists the specified user is subscribed to.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-subscriptions>

get_mentions_timeline (**params)

Returns the 20 most recent mentions (tweets containing a users's @screen_name) for the authenticating user.

Docs: <https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-mentions-timeline>

get_oembed_tweet (**params)

Returns information allowing the creation of an embedded representation of a Tweet on third party sites.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-oembed>

get_outgoing_friendship_ids (**params)

Returns a collection of numeric IDs for every protected user for whom the authenticating user has a pending follow request.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friendships-outgoing>

get_place_trends (**params)

Returns the top 10 trending topics for a specific WOEID, if trending information is available for it.

Docs: <https://developer.twitter.com/en/docs/trends/trends-for-location/api-reference/get-trends-place>

get_privacy_policy (**params)

Returns Twitter's Privacy Policy

Docs: <https://developer.twitter.com/en/docs/developer-utilities/privacy-policy/api-reference/get-help-privacy>

get_profile_banner_sizes (**params)

Returns a map of the available size variations of the specified user's profile banner.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/get-users-profile-banner>

get_retweeters_ids (**params)

Returns a collection of up to 100 user IDs belonging to users who have retweeted the tweet specified by the id parameter.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-retweeters-ids>

get_retweets (**params)

Returns up to 100 of the first retweets of a given tweet.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-retweet-id>

get_saved_searches (**params)

Returns the authenticated user's saved search queries.

Docs: https://developer.twitter.com/en/docs/tweets/search/api-reference/get-saved_searches-list

get_sent_messages (***params*)

Returns the 20 most recent direct messages sent by the authenticating user.

Docs: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/api-reference/get-sent-message>

get_similar_places (***params*)

Locates places near the given coordinates which are similar in name.

Docs: https://dev.twitter.com/docs/api/1.1/get/geo/similar_places

get_specific_list (***params*)

Returns the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-show>

get_supported_languages (***params*)

Returns the list of languages supported by Twitter along with their ISO 639-1 code.

Docs: <https://developer.twitter.com/en/docs/developer-utilities/supported-languages/api-reference/get-help-languages>

get_tos (***params*)

Return the Twitter Terms of Service

Docs: <https://developer.twitter.com/en/docs/developer-utilities/terms-of-service/api-reference/get-help-tos>

get_twitter_configuration (***params*)

Returns the current configuration used by Twitter

Docs: <https://developer.twitter.com/en/docs/developer-utilities/configuration/api-reference/get-help-configuration>

get_user_ids_of_blocked_retweets (***params*)

Returns a collection of user_ids that the currently authenticated user does not want to receive retweets from.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friendships-no_retweets-ids

get_user_suggestions (***params*)

Access to Twitter's suggested user list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-suggestions>

get_user_suggestions_by_slug (***params*)

Access the users in a given category of the Twitter suggested user list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-suggestions-slug>

get_user_suggestions_statuses_by_slug (***params*)

Access the users in a given category of the Twitter suggested user list and return their most recent status if they are not a protected user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-suggestions-slug-members>

get_user_timeline (**params)

Returns a collection of the most recent Tweets posted by the user indicated by the `screen_name` or `user_id` parameters.

Docs: https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline

static_html_for_tweet (tweet, use_display_url=True, use_expanded_url=False, expand_quoted_status=False)

Return HTML for a tweet (urls, mentions, hashtags, symbols replaced with links)

Parameters

- **tweet** – Tweet object from received from Twitter API
- **use_display_url** – Use display URL to represent link

(ex. google.com, github.com). Default: True ;param `use_expanded_url`: Use expanded URL to represent link (e.g. <http://google.com>). Default False

If `use_expanded_url` is True, it overrides `use_display_url`. If `use_display_url` and `use_expanded_url` is False, short url will be used (t.co/xxxxx)

invalidate_token (**params)

Allows a registered application to revoke an issued OAuth 2 Bearer Token by presenting its client credentials.

Docs: https://developer.twitter.com/en/docs/basics/authentication/api-reference/invalidate_token

is_list_member (**params)

Check if the specified user is a member of the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-members-show>

is_list_subscriber (**params)

Check if the specified user is a subscriber of the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-subscribers-show>

list_block_ids (**params)

Returns an array of numeric user ids the authenticating user is blocking.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/get-blocks-ids>

list_blocks (**params)

Returns a collection of user objects that the authenticating user is blocking.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/get-blocks-list>

list_mute_ids (**params)

Returns an array of numeric user ids the authenticating user is muting.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/get-mutes-users-ids>

list_mutes (**params)

Returns a collection of user objects that the authenticating user is muting.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/get-mutes-users-list>

lookup_friendships (**params)

Returns the relationships of the authenticating user to the comma-separated list of up to 100 screen_names or user_ids provided.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friendships-lookup>

lookup_status (**params)

Returns fully-hydrated tweet objects for up to 100 tweets per request, as specified by comma-separated values passed to the id parameter.

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup>

lookup_user (**params)

Returns fully-hydrated user objects for up to 100 users per request, as specified by comma-separated values passed to the user_id and/or screen_name parameters.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-lookup>

obtain_access_token ()

Returns an OAuth 2 access token to make OAuth 2 authenticated read-only calls.

Return type string

post (endpoint, params=None, version='1.1')

Shortcut for POST requests via [request](#)

remove_profile_banner (**params)

Removes the uploaded profile banner for the authenticating user. Returns HTTP 200 upon success.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-remove_profile_banner

report_spam (**params)

Report the specified user as a spam account to Twitter.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/mute-block-report-users/api-reference/post-users-report_spam

request (endpoint, method='GET', params=None, version='1.1')

Return dict of response received from Twitter's API

Parameters

- **endpoint** (string) – (required) Full url or Twitter API endpoint (e.g. search/tweets)
- **method** (string) – (optional) Method of accessing data, either GET or POST. (default GET)
- **params** (dict or None) – (optional) Dict of parameters (if any) accepted the by Twitter API endpoint you are trying to access (default None)
- **version** (string) – (optional) Twitter API version to access (default 1.1)

Return type dict

retweet (**params)

Retweets a tweet specified by the id parameter

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-retweet-id>

retweeted_of_me (***params*)

Returns the most recent tweets authored by the authenticating user that have been retweeted by others.

Docs: https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-retweets_of_me

reverse_geocode (***params*)

Given a latitude and a longitude, searches for up to 20 places that can be used as a place_id when updating a status.

Docs: https://developer.twitter.com/en/docs/geo/places-near-location/api-reference/get-geo-reverse_geocode

search (***params*)

Returns a collection of relevant Tweets matching a specified query.

Docs: <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets>

search_geo (***params*)

Search for places that can be attached to a statuses/update.

Docs: <https://developer.twitter.com/en/docs/geo/places-near-location/api-reference/get-geo-search>

search_users (***params*)

Provides a simple, relevance-based search interface to public user accounts on Twitter.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-search>

send_direct_message (***params*)

Sends a new direct message to the specified user from the authenticating user.

Docs: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/api-reference/new-message>

show_friendship (***params*)

Returns detailed information about the relationship between two arbitrary users.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friendships-show>

show_lists (***params*)

Returns all lists the authenticating or specified user subscribes to, including their own.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-list>

show_owned_lists (***params*)

Returns the lists owned by the specified Twitter user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/get-lists-ownerships>

show_saved_search (***params*)

Retrieve the information for the saved search represented by the given id.

Docs: https://developer.twitter.com/en/docs/tweets/search/api-reference/get-saved_searches-show-id

show_status (***params*)

Returns a single Tweet, specified by the id parameter

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-show-id>

show_user (**params)

Returns a variety of information about the user specified by the required `user_id` or `screen_name` parameter.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-show>

subscribe_to_list (**params)

Subscribes the authenticated user to the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-subscribers-create>

unsubscribe_from_list (**params)

Unsubscribes the authenticated user from the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-subscribers-destroy>

update_account_settings (**params)

Updates the authenticating user's settings.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-settings>

update_delivery_service (**params)

Sets which device Twitter delivers updates to for the authenticating user.

Docs: https://dev.twitter.com/docs/api/1.1/post/account/update_delivery_device

update_friendship (**params)

Allows one to enable or disable retweets and device notifications from the specified user.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/post-friendships-update>

update_list (**params)

Updates the specified list.

Docs: <https://developer.twitter.com/en/docs/accounts-and-users/create-manage-lists/api-reference/post-lists-update>

update_profile (**params)

Sets values that users are able to set under the “Account” tab of their settings page.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile

update_profile_background_image (**params)

Uploads a profile banner on behalf of the authenticating user.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile_banner

update_profile_banner_image (**params)

Updates the authenticating user's profile background image.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile_background_image

update_profile_colors (**params)

Sets one or more hex values that control the color scheme of the authenticating user's profile page on twitter.com.

This method is deprecated, replaced by the `profile_link_color` parameter to `update_profile()`.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile

update_profile_image (***params*)

Updates the authenticating user's profile image.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile_image

update_status (***params*)

Updates the authenticating user's current status, also known as tweeting

Docs: <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-update>

update_status_with_media (***params*)

Updates the authenticating user's current status and attaches media for upload. In other words, it creates a Tweet with a picture attached.

Docs: https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-update_with_media

upload_media (***params*)

Uploads media file to Twitter servers. The file will be available to be attached to a status for 60 minutes. To attach to a update, pass a list of returned media ids to the `update_status()` method using the `media_ids` param.

Docs: <https://developer.twitter.com/en/docs/media/upload-media/api-reference/post-media-upload>

upload_video (*media, media_type, media_category=None, size=None, check_progress=False*)

Uploads video file to Twitter servers in chunks. The file will be available to be attached to a status for 60 minutes. To attach to a update, pass a list of returned media ids to the `update_status()` method using the `media_ids` param.

Upload happens in 3 stages: - INIT call with size of media to be uploaded(in bytes). If this is more than 15mb, twitter will return error. - APPEND calls each with media chunk. This returns a 204(No Content) if chunk is received. - FINALIZE call to complete media upload. This returns `media_id` to be used with status update.

Twitter media upload api expects each chunk to be not more than 5mb. We are sending chunk of 1mb each.

Docs: <https://developer.twitter.com/en/docs/media/upload-media/uploading-media/chunked-media-upload>

verify_credentials (***params*)

Returns an HTTP 200 OK response code and a representation of the requesting user if authentication was successful; returns a 401 status code and an error message if not.

Docs: https://developer.twitter.com/en/docs/accounts-and-users/manage-account-settings/api-reference/get-account-verify_credentials

3.1.2 Streaming Interface

```
class twython.TwythonStreamer(app_key, app_secret, oauth_token, oauth_token_secret, time-
out=300, retry_count=None, retry_in=10, client_args=None,
handlers=None, chunk_size=1)
```

__init__ (*app_key, app_secret, oauth_token, oauth_token_secret, timeout=300, retry_count=None, retry_in=10, client_args=None, handlers=None, chunk_size=1*)

Streaming class for a friendly streaming user experience Authentication IS required to use the Twitter Streaming API

Parameters

- **app_key** – (required) Your applications key
- **app_secret** – (required) Your applications secret key
- **oauth_token** – (required) Used with `oauth_token_secret` to make authenticated calls
- **oauth_token_secret** – (required) Used with `oauth_token` to make authenticated calls
- **timeout** – (optional) How long (in secs) the streamer should wait for a response from Twitter Streaming API
- **retry_count** – (optional) Number of times the API call should be retired
- **retry_in** – (optional) Amount of time (in secs) the previous API call should be tried again
- **client_args** – (optional) Accepts some requests Session parameters and some requests Request parameters. See <http://docs.python-requests.org/en/latest/api/#sessionapi> and requests section below it for details. [ex. headers, proxies, verify(SSL verification)]
- **handlers** – (optional) Array of message types for which corresponding handlers will be called
- **chunk_size** – (optional) Define the buffer size before data is actually returned from the Streaming API. Default: 1

disconnect ()

Used to disconnect the streaming client manually

on_error (*status_code, data*)

Called when stream returns non-200 status code

Feel free to override this to handle your streaming data how you want it handled.

Parameters

- **status_code** (*int*) – Non-200 status code sent from stream
- **data** (*dict*) – Error message sent from stream

on_success (*data*)

Called when data has been successfully received from the stream. Returns True if other handlers for this message should be invoked.

Feel free to override this to handle your streaming data how you want it handled. See <https://developer.twitter.com/en/docs/tweets/filter-realtime/guides/streaming-message-types> for messages sent along in stream responses.

Parameters **data** (*dict*) – data recieved from the stream

on_timeout ()

Called when the request has timed out

3.1.2.1 Streaming Types

class twython.streaming.types.**TwythonStreamerTypes** (*streamer*)

Class for different stream endpoints

Not all streaming endpoints have nested endpoints. User Streams and Site Streams are single streams with no nested endpoints Status Streams include filter, sample and firehose endpoints

site (***params*)

Stream site

Accepted params found at: <https://dev.twitter.com/docs/api/1.1/get/site>

user (***params*)

Stream user

Accepted params found at: <https://dev.twitter.com/docs/api/1.1/get/user>

class twython.streaming.types.**TwythonStreamerTypesStatuses** (*streamer*)

Class for different statuses endpoints

Available so `TwythonStreamer.statuses.filter()` is available. Just a bit cleaner than `TwythonStreamer.statuses_filter()`, `statuses_sample()`, etc. all being single methods in `TwythonStreamer`.

dynamic_filter ()

Stream statuses/filter with dynamic parameters

filter (***params*)

Stream statuses/filter

Parameters ****params** – Parameters to send with your stream request

Accepted params found at: <https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter>

firehose (***params*)

Stream statuses/firehose

Parameters ****params** – Parameters to send with your stream request

Accepted params found at: <https://dev.twitter.com/docs/api/1.1/get/statuses/firehose>

sample (***params*)

Stream statuses/sample

Parameters ****params** – Parameters to send with your stream request

Accepted params found at: <https://developer.twitter.com/en/docs/tweets/sample-realtime/api-reference/get-statuses-sample>

set_dynamic_filter (***params*)

Set/update statuses/filter

Parameters ****params** – Parameters to send with your stream request

Accepted params found at: <https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter>

3.1.3 Exceptions

exception twython.**TwythonError** (*msg, error_code=None, retry_after=None*)

Generic error class, catch-all for most Twython issues. Special cases are handled by `TwythonAuthError` &

TwythonRateLimitError.

```
from twython import TwythonError, TwythonRateLimitError, TwythonAuthError
```

exception `twython.TwythonAuthError(msg, error_code=None, retry_after=None)`

Raised when you try to access a protected resource and it fails due to some issue with your authentication.

exception `twython.TwythonRateLimitError(msg, error_code, retry_after=None)`

Raised when you've hit a rate limit.

The amount of seconds to retry your request in will be appended to the message.

t

twython, [17](#)

Symbols

`__init__()` (*twython.Twython method*), 17
`__init__()` (*twython.TwythonStreamer method*), 29

A

`add_list_member()` (*twython.Twython method*), 18

C

`construct_api_url()` (*twython.Twython static method*), 18
`create_block()` (*twython.Twython method*), 18
`create_favorite()` (*twython.Twython method*), 18
`create_friendship()` (*twython.Twython method*), 18
`create_list()` (*twython.Twython method*), 18
`create_list_members()` (*twython.Twython method*), 18
`create_metadata()` (*twython.Twython method*), 19
`create_mute()` (*twython.Twython method*), 19
`create_place()` (*twython.Twython method*), 19
`create_saved_search()` (*twython.Twython method*), 19
`cursor()` (*twython.Twython method*), 19

D

`delete_list()` (*twython.Twython method*), 19
`delete_list_member()` (*twython.Twython method*), 19
`delete_list_members()` (*twython.Twython method*), 19
`destroy_block()` (*twython.Twython method*), 20
`destroy_direct_message()` (*twython.Twython method*), 20
`destroy_favorite()` (*twython.Twython method*), 20
`destroy_friendship()` (*twython.Twython method*), 20
`destroy_mute()` (*twython.Twython method*), 20

`destroy_saved_search()` (*twython.Twython method*), 20

`destroy_status()` (*twython.Twython method*), 20

`disconnect()` (*twython.TwythonStreamer method*), 30

`dynamic_filter()` (*twython.streaming.types.TwythonStreamerTypesStatuses method*), 31

F

`filter()` (*twython.streaming.types.TwythonStreamerTypesStatuses method*), 31

`firehose()` (*twython.streaming.types.TwythonStreamerTypesStatuses method*), 31

G

`get()` (*twython.Twython method*), 20

`get_account_settings()` (*twython.Twython method*), 20

`get_application_rate_limit_status()` (*twython.Twython method*), 20

`get_authentication_tokens()` (*twython.Twython method*), 20

`get_authorized_tokens()` (*twython.Twython method*), 21

`get_available_trends()` (*twython.Twython method*), 21

`get_closest_trends()` (*twython.Twython method*), 21

`get_contributees()` (*twython.Twython method*), 21

`get_contributors()` (*twython.Twython method*), 21

`get_direct_message()` (*twython.Twython method*), 21

`get_direct_messages()` (*twython.Twython method*), 21

`get_favorites()` (*twython.Twython method*), 21

`get_followers_ids()` (*twython.Twython method*), 21

[get_followers_list\(\)](#) (*twython.Twython method*), 21
[get_friends_ids\(\)](#) (*twython.Twython method*), 22
[get_friends_list\(\)](#) (*twython.Twython method*), 22
[get_geo_info\(\)](#) (*twython.Twython method*), 22
[get_home_timeline\(\)](#) (*twython.Twython method*), 22
[get_incoming_friendship_ids\(\)](#) (*twython.Twython method*), 22
[get_lastfunction_header\(\)](#) (*twython.Twython method*), 22
[get_list_members\(\)](#) (*twython.Twython method*), 22
[get_list_memberships\(\)](#) (*twython.Twython method*), 22
[get_list_statuses\(\)](#) (*twython.Twython method*), 22
[get_list_subscribers\(\)](#) (*twython.Twython method*), 22
[get_list_subscriptions\(\)](#) (*twython.Twython method*), 23
[get_mentions_timeline\(\)](#) (*twython.Twython method*), 23
[get_oembed_tweet\(\)](#) (*twython.Twython method*), 23
[get_outgoing_friendship_ids\(\)](#) (*twython.Twython method*), 23
[get_place_trends\(\)](#) (*twython.Twython method*), 23
[get_privacy_policy\(\)](#) (*twython.Twython method*), 23
[get_profile_banner_sizes\(\)](#) (*twython.Twython method*), 23
[get_retweeters_ids\(\)](#) (*twython.Twython method*), 23
[get_retweets\(\)](#) (*twython.Twython method*), 23
[get_saved_searches\(\)](#) (*twython.Twython method*), 23
[get_sent_messages\(\)](#) (*twython.Twython method*), 24
[get_similar_places\(\)](#) (*twython.Twython method*), 24
[get_specific_list\(\)](#) (*twython.Twython method*), 24
[get_supported_languages\(\)](#) (*twython.Twython method*), 24
[get_tos\(\)](#) (*twython.Twython method*), 24
[get_twitter_configuration\(\)](#) (*twython.Twython method*), 24
[get_user_ids_of_blocked_retweets\(\)](#) (*twython.Twython method*), 24
[get_user_suggestions\(\)](#) (*twython.Twython method*), 24

[get_user_suggestions_by_slug\(\)](#) (*twython.Twython method*), 24
[get_user_suggestions_statuses_by_slug\(\)](#) (*twython.Twython method*), 24
[get_user_timeline\(\)](#) (*twython.Twython method*), 24

H

[html_for_tweet\(\)](#) (*twython.Twython static method*), 25

I

[invalidate_token\(\)](#) (*twython.Twython method*), 25
[is_list_member\(\)](#) (*twython.Twython method*), 25
[is_list_subscriber\(\)](#) (*twython.Twython method*), 25

L

[list_block_ids\(\)](#) (*twython.Twython method*), 25
[list_blocks\(\)](#) (*twython.Twython method*), 25
[list_mute_ids\(\)](#) (*twython.Twython method*), 25
[list_mutes\(\)](#) (*twython.Twython method*), 25
[lookup_friendships\(\)](#) (*twython.Twython method*), 25
[lookup_status\(\)](#) (*twython.Twython method*), 26
[lookup_user\(\)](#) (*twython.Twython method*), 26

O

[obtain_access_token\(\)](#) (*twython.Twython method*), 26
[on_error\(\)](#) (*twython.TwythonStreamer method*), 30
[on_success\(\)](#) (*twython.TwythonStreamer method*), 30
[on_timeout\(\)](#) (*twython.TwythonStreamer method*), 30

P

[post\(\)](#) (*twython.Twython method*), 26

R

[remove_profile_banner\(\)](#) (*twython.Twython method*), 26
[report_spam\(\)](#) (*twython.Twython method*), 26
[request\(\)](#) (*twython.Twython method*), 26
[retweet\(\)](#) (*twython.Twython method*), 26
[retweeted_of_me\(\)](#) (*twython.Twython method*), 26
[reverse_geocode\(\)](#) (*twython.Twython method*), 27

S

[sample\(\)](#) (*twython.streaming.types.TwythonStreamerTypesStatuses method*), 31
[search\(\)](#) (*twython.Twython method*), 27

[search_geo\(\) \(twython.Twython method\), 27](#)
[search_users\(\) \(twython.Twython method\), 27](#)
[send_direct_message\(\) \(twython.Twython method\), 27](#)
[set_dynamic_filter\(\) \(twython.streaming.types.TwythonStreamerTypesStatuses method\), 31](#)
[show_friendship\(\) \(twython.Twython method\), 27](#)
[show_lists\(\) \(twython.Twython method\), 27](#)
[show_owned_lists\(\) \(twython.Twython method\), 27](#)
[show_saved_search\(\) \(twython.Twython method\), 27](#)
[show_status\(\) \(twython.Twython method\), 27](#)
[show_user\(\) \(twython.Twython method\), 27](#)
[site\(\) \(twython.streaming.types.TwythonStreamerTypes method\), 31](#)
[subscribe_to_list\(\) \(twython.Twython method\), 28](#)

V

[verify_credentials\(\) \(twython.Twython method\), 29](#)

T

[Twython \(class in twython\), 17](#)
[twython \(module\), 17](#)
[TwythonAuthError, 32](#)
[TwythonError, 31](#)
[TwythonRateLimitError, 32](#)
[TwythonStreamer \(class in twython\), 29](#)
[TwythonStreamerTypes \(class in twython.streaming.types\), 31](#)
[TwythonStreamerTypesStatuses \(class in twython.streaming.types\), 31](#)

U

[unsubscribe_from_list\(\) \(twython.Twython method\), 28](#)
[update_account_settings\(\) \(twython.Twython method\), 28](#)
[update_delivery_service\(\) \(twython.Twython method\), 28](#)
[update_friendship\(\) \(twython.Twython method\), 28](#)
[update_list\(\) \(twython.Twython method\), 28](#)
[update_profile\(\) \(twython.Twython method\), 28](#)
[update_profile_background_image\(\) \(twython.Twython method\), 28](#)
[update_profile_banner_image\(\) \(twython.Twython method\), 28](#)
[update_profile_colors\(\) \(twython.Twython method\), 28](#)
[update_profile_image\(\) \(twython.Twython method\), 29](#)
[update_status\(\) \(twython.Twython method\), 29](#)
[update_status_with_media\(\) \(twython.Twython method\), 29](#)