txTwitter Documentation

Release 0.1.6a0

Jeremy Thurgood and Justin van der Merwe

May 09, 2015

Contents

1 First steps **3**
1.1 Authorization 3
1.2 Making a request 3

2 Testing code that uses txTwitter **5**

3 API reference **7**
3.1 `txtwitter.twitter` 7

4 Indices and tables **15**

Python Module Index **17**

Contents:

CHAPTER 1

First steps

Todo

Finish writing this.

1.1 Authorization

Todo

Do we support application-only usage?

In order to sign API requests, txTwitter needs the following credentials:

- The `consumer_key` and `consumer_secret` identify your application and can be found in the Application Management page for your application at https://apps.twitter.com
- The `token_key` and `token_secret` identify a Twitter account and can be obtained via OAuth or generated in the Application Management interface mentioned above for that account that owns the application.

The full application-user OAuth flow involves user interaction with a web browser, so txTwitter leaves all of that up to the application. [1]

1.2 Making a request

Todo

Finish writing this.

To make a request, you need an instance of *`TwitterClient`* with suitable credentials: [2]

```
from txtwitter.twitter import TwitterClient
from twisted.internet.task import react

# Create an authenticated twitter client instance.
```

[1] While txTwitter could probably implement parts of the flow that don't involve the browser, we decided the extra complexity wasn't worth the minimal value it may provide.

[2] These credentials are from Twitter's auth documentation and were valid at the time of writing.

```
twitter = TwitterClient(
    consumer_key='cChZNFj6T5R0TigYB9yd1w',
    consumer_secret='L8qq9PZyRg6ieKGEKhZolGC0vJWLw8iEJ88DRdyOg',
    token_key='7588892-kagSNqWge8gB1WwE3plnFsJHAZVfxWD7Vb57p0b4',
    token_secret='PbKfYqSryyeKDWz4ebtY3o5ogNLG11WJuZBc9fQrQo')


# We need a callback function to do something with the tweets we get.
# This one just prints them.
def print_tweets(tweets):
    for tweet in tweets:
        print tweet['text']

def main(reactor):
    # We'll be getting the most recent tweets from @twistedmatrix.
    tweets_d = twitter.statuses_user_timeline(screen_name="twistedmatrix")
    # Add our callback to the Deferred result so the tweets will be printed.
    tweets_d.addCallback(print_tweets)
    # Return the Deferred so react() waits for it to finish.
    return tweets

# We're ready, run the Twisted reactor!
react(main)
```

CHAPTER 2

Testing code that uses txTwitter

Todo

Write this.

CHAPTER 3

API reference

Todo

Finish this.

3.1 txtwitter.twitter

Module containing the client library implementation.

class txtwitter.twitter.**TwitterClient**(*token_key*, *token_secret*, *consumer_key*, *consumer_secret*, *api_url='https://api.twitter.com/1.1/'*, *stream_url='https://stream.twitter.com/1.1/'*, *user-stream_url='https://userstream.twitter.com/1.1/'*, *agent=None*)

TODO: Document this.

direct_messages(*since_id=None*, *max_id=None*, *count=None*, *include_entities=None*, *skip_status=None*)

Gets the 20 most recent direct messages received by the authenticating user.

https://dev.twitter.com/docs/api/1.1/get/direct_messages

Parameters

- **since_id** (*str*) – Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occured since the since_id, the since_id will be forced to the oldest ID available.
- **max_id** (*str*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **count** (*int*) – Specifies the number of direct messages to try and retrieve, up to a maximum of `200`. The value of count is best thought of as a limit to the number of Tweets to return because suspended or deleted content is removed after the count has been applied.
- **include_entities** (*bool*) – The entities node will not be included when set to `False`.
- **skip_status** (*bool*) – When set to `True`, statuses will not be included in the returned user objects.

Returns A list of direct message dicts.

direct_messages_destroy(*id*, *include_entities=None*)

Destroys the direct message with the given id.

https://dev.twitter.com/docs/api/1.1/post/direct_messages/destroy

Parameters

- **id** (*str*) – (*required*) The ID of the direct message.
- **include_entities** (*bool*) – The entities node will not be included when set to `False`.

Returns A direct message dict containing the destroyed direct message.

direct_messages_new(*text*, *user_id=None*, *screen_name=None*)

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

https://dev.twitter.com/docs/api/1.1/post/direct_messages/new

Parameters

- **text** (*str*) – (*required*) The text of your direct message.
- **user_id** (*str*) – The ID of the user who should receive the direct message. Required if `screen_name` isn't given.
- **screen_name** (*str*) – The screen name of the user who should receive the direct message. Required if `user_id` isn't given.

Returns A direct message dict containing the sent direct message.

direct_messages_sent(*since_id=None*, *max_id=None*, *count=None*, *include_entities=None*, *page=None*)

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

https://dev.twitter.com/docs/api/1.1/get/direct_messages/sent

Parameters

- **since_id** (*str*) – Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occured since the since_id, the since_id will be forced to the oldest ID available.
- **max_id** (*str*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **count** (*int*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **page** (*int*) – Specifies the page of results to retrieve.
- **include_entities** (*bool*) – The entities node will not be included when set to `False`.

Returns A list of direct message dicts.

direct_messages_show(*id*)

Gets the direct message with the given id.

https://dev.twitter.com/docs/api/1.1/get/direct_messages/show

Parameters **id** (*str*) – (*required*) The ID of the direct message.

Returns A direct message dict.

friendships_create(*user_id=None*, *screen_name=None*, *follow=None*)

Allows the authenticating users to follow the specified user.

https://dev.twitter.com/docs/api/1.1/post/friendships/create

Parameters

- **user_id** (*str*) – The screen name of the user for whom to befriend. Required if `screen_name` isn't given.
- **screen_name** (*str*) – The ID of the user for whom to befriend. Required if `user_id` isn't given.
- **follow** (*bool*) – Enable notifications for the target user.

Returns A dict containing the newly followed user.

friendships_destroy(*user_id=None*, *screen_name=None*)

Allows the authenticating user to unfollow the specified user.

https://dev.twitter.com/docs/api/1.1/post/friendships/destroy

Parameters

- **user_id** (*str*) – The screen name of the user for whom to unfollow. Required if `screen_name` isn't given.
- **screen_name** (*str*) – The ID of the user for whom to unfollow. Required if `user_id` isn't given.

Returns A dict containing the newly unfollowed user.

statuses_destroy(*id*, *trim_user=None*)

Destroys the status specified by the ID parameter.

https://dev.twitter.com/docs/api/1.1/post/statuses/destroy/%3Aid

Parameters

- **id** (*str*) – (*required*) The numerical ID of the desired tweet.
- **trim_user** (*bool*) – When set to `True`, the return value's user object includes only the status author's numerical ID.

Returns A tweet dict containing the destroyed tweet.

statuses_home_timeline(*count=None*, *since_id=None*, *max_id=None*, *trim_user=None*, *exclude_replies=None*, *contributor_details=None*, *include_entities=None*)

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

https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline

Parameters

- **count** (*int*) – Specifies the number of tweets to try and retrieve, up to a maximum of 200.
- **since_id** (*str*) – Returns results with an ID greater than (that is, more recent than) the specified ID. Tweets newer than this may not be returned due to certain API limits.
- **max_id** (*str*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **trim_user** (*bool*) – When set to `True`, the tweet's user object includes only the status author's numerical ID.

- **exclude_replies** (*bool*) – When set to `True`, replies will not appear in the timeline.
- **contributor_details** (*bool*) – This parameter enhances the contributors element of the status response to include the screen_name of the contributor. By default only the user_id of the contributor is included.
- **include_entities** (*bool*) – When set to `False`, the `entities` node will not be included.

Returns A list of tweet dicts.

statuses_mentions_timeline(*count=None*, *since_id=None*, *max_id=None*, *trim_user=None*, *contributor_details=None*, *include_entities=None*)

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

https://dev.twitter.com/docs/api/1.1/get/statuses/mentions_timeline

Parameters

- **count** (*int*) – Specifies the number of tweets to try and retrieve, up to a maximum of 200.
- **since_id** (*str*) – Returns results with an ID greater than (that is, more recent than) the specified ID. Tweets newer than this may not be returned due to certain API limits.
- **max_id** (*str*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **trim_user** (*bool*) – When set to `True`, the tweet's user object includes only the status author's numerical ID.
- **contributor_details** (*bool*) – This parameter enhances the contributors element of the status response to include the screen_name of the contributor. By default only the user_id of the contributor is included.
- **include_entities** (*bool*) – When set to `False`, the `entities` node will not be included.

Returns A list of tweet dicts.

statuses_retweet(*id*, *trim_user=None*)

Retweets the status specified by the ID parameter.

https://dev.twitter.com/docs/api/1.1/post/statuses/retweet/%3Aid

Parameters

- **id** (*str*) – (*required*) The numerical ID of the desired tweet.
- **trim_user** (*bool*) – When set to `True`, the return value's user object includes only the status author's numerical ID.

Returns A tweet dict containing the retweet. (Contains the retweeted tweet in the `retweeted_status` field.)

statuses_retweets(*id*, *count=None*, *trim_user=None*)

Returns a list of the most recent retweets of the Tweet specified by the id parameter.

https://dev.twitter.com/docs/api/1.1/get/statuses/retweets/%3Aid

Parameters

- **id** (*str*) – (*required*) The numerical ID of the desired tweet.
- **count** (*int*) – The maximum number of retweets to return. (Max 100)

- **trim_user** (*bool*) – When set to `True`, the tweet's user object includes only the status author's numerical ID.

Returns A tweet dict.

statuses_show(*id*, *trim_user=None*, *include_my_retweet=None*, *include_entities=None*)

Returns a single Tweet, specified by the id parameter.

https://dev.twitter.com/docs/api/1.1/get/statuses/show/%3Aid

Parameters

- **id** (*str*) – (*required*) The numerical ID of the desired tweet.
- **trim_user** (*bool*) – When set to `True`, the tweet's user object includes only the status author's numerical ID.
- **include_my_retweet** (*bool*) – When set to `True`, any Tweet returned that has been retweeted by the authenticating user will include an additional `current_user_retweet` node, containing the ID of the source status for the retweet.
- **include_entities** (*bool*) – When set to `False`, the `entities` node will not be included.

Returns A tweet dict.

statuses_update(*status*, *in_reply_to_status_id=None*, *lat=None*, *long=None*, *place_id=None*, *display_coordinates=None*, *trim_user=None*)

Posts a tweet.

https://dev.twitter.com/docs/api/1.1/post/statuses/update

Parameters

- **status** (*str*) – (*required*) The text of your tweet, typically up to 140 characters. URL encode as necessary. t.co link wrapping may affect character counts.

  There are some special commands in this field to be aware of. For instance, preceding a message with "D " or "M " and following it with a screen name can create a direct message to that user if the relationship allows for it.

- **in_reply_to_status_id** (*str*) – The ID of an existing status that the update is in reply to.

  Note: This parameter will be ignored unless the author of the tweet this parameter references is mentioned within the status text. Therefore, you must include @username, where username is the author of the referenced tweet, within `status`.

- **lat** (*float*) – The latitude of the location this tweet refers to. This parameter will be ignored unless it is inside the range -90.0 to +90.0 (North is positive) inclusive. It will also be ignored if there isn't a corresponding long parameter.
- **long** (*float*) – The longitude of the location this tweet refers to. The valid ranges for longitude is -180.0 to +180.0 (East is positive) inclusive. This parameter will be ignored if outside that range, if it is not a number, if geo_enabled is disabled, or if there not a corresponding lat parameter.
- **place_id** (*str*) – A place in the world. These IDs can be retrieved from GET geo/reverse_geocode. (TODO: Reference method when it exists.)
- **display_coordinates** (*bool*) – Whether or not to put a pin on the exact coordinates a tweet has been sent from.
- **trim_user** (*bool*) – When set to `True`, the return value's user object includes only the status author's numerical ID.

Returns A tweet dict containing the posted tweet.

statuses_user_timeline(*user_id=None*, *screen_name=None*, *since_id=None*, *count=None*, *max_id=None*, *trim_user=None*, *exclude_replies=None*, *contributor_details=None*, *include_rts=None*)

Returns a list of the most recent tweets posted by the specified user.

https://dev.twitter.com/docs/api/1.1/get/statuses/user_timeline

Either `user_id` or `screen_name` must be provided.

Parameters

- **user_id** (*str*) – The ID of the user to return tweets for.
- **screen_name** (*str*) – The screen name of the user to return tweets for.
- **since_id** (*str*) – Returns results with an ID greater than (that is, more recent than) the specified ID. Tweets newer than this may not be returned due to certain API limits.
- **count** (*int*) – Specifies the number of tweets to try and retrieve, up to a maximum of 200.
- **max_id** (*str*) – Returns results with an ID less than (that is, older than) or equal to the specified ID.
- **trim_user** (*bool*) – When set to `True`, the tweet's user object includes only the status author's numerical ID.
- **exclude_replies** (*bool*) – When set to `True`, replies will not appear in the timeline.
- **contributor_details** (*bool*) – This parameter enhances the contributors element of the status response to include the screen_name of the contributor. By default only the user_id of the contributor is included.
- **include_rts** (*bool*) – When set to `False`, retweets will not appear in the timeline.

Returns A list of tweet dicts.

stream_filter(*delegate*, *follow=None*, *track=None*, *locations=None*, *stall_warnings=None*)

Streams public messages filtered by various parameters.

https://dev.twitter.com/docs/api/1.1/post/statuses/filter

At least one of `follow`, `track`, or `locations` must be provided. See the API documentation linked above for details on these parameters and the various limits on this API.

Parameters

- **delegate** – A delegate function that will be called for each message in the stream and will be passed the message dict as the only parameter. The message dicts passed to this function may represent any message type and the delegate is responsible for any dispatch that may be required. (`txtwitter.messagetools` may be helpful here.)
- **follow** (*list*) – A list of user IDs, indicating the users to return statuses for in the stream.
- **track** (*list*) – List of keywords to track.
- **locations** (*list*) – List of location bounding boxes to track. XXX: Currently unsupported.
- **stall_warnings** (*bool*) – Specifies whether stall warnings should be delivered.

Returns An unstarted `TwitterStreamService`.

userstream_user(*delegate*, *stall_warnings=None*, *with_='followings'*, *replies=None*)

Streams messages for a single user.

https://dev.twitter.com/docs/api/1.1/get/user

The `stringify_friend_ids` parameter is always set to `'true'` for consistency with the use of string identifiers elsewhere.

Parameters

- **`delegate`** – A delegate function that will be called for each message in the stream and will be passed the message dict as the only parameter. The message dicts passed to this function may represent any message type and the delegate is responsible for any dispatch that may be required. (`txtwitter.messagetools` may be helpful here.)
- **`stall_warnings`** (*bool*) – Specifies whether stall warnings should be delivered.
- **`with`** (*str*) – If `'followings'` (the default), the stream will include messages from both the authenticated user and the authenticated user's followers. If `'user'`, the stream will only include messages from (or mentioning) the autheticated user. All other values are invalid. (The underscore appended to the parameter name is to avoid conflicting with Python's `with` keyword.)
- **`replies`** (*str*) – If set to `'all'`, replies to tweets will be included even if the authenticated user does not follow both parties.

Returns An unstarted `TwitterStreamService`.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

t

txtwitter.twitter, 7

D

direct_messages() (txtwitter.twitter.TwitterClient method), 7
direct_messages_destroy() (txtwitter.twitter.TwitterClient method), 7
direct_messages_new() (txtwitter.twitter.TwitterClient method), 8
direct_messages_sent() (txtwitter.twitter.TwitterClient method), 8
direct_messages_show() (txtwitter.twitter.TwitterClient method), 8

F

friendships_create() (txtwitter.twitter.TwitterClient method), 8
friendships_destroy() (txtwitter.twitter.TwitterClient method), 9

S

statuses_destroy() (txtwitter.twitter.TwitterClient method), 9
statuses_home_timeline() (txtwitter.twitter.TwitterClient method), 9
statuses_mentions_timeline() (txtwitter.twitter.TwitterClient method), 10
statuses_retweet() (txtwitter.twitter.TwitterClient method), 10
statuses_retweets() (txtwitter.twitter.TwitterClient method), 10
statuses_show() (txtwitter.twitter.TwitterClient method), 11
statuses_update() (txtwitter.twitter.TwitterClient method), 11
statuses_user_timeline() (txtwitter.twitter.TwitterClient method), 12
stream_filter() (txtwitter.twitter.TwitterClient method), 12

T

TwitterClient (class in txtwitter.twitter), 7
txtwitter.twitter (module), 7

U

userstream_user() (txtwitter.twitter.TwitterClient method), 12