
python-twitch-client Documentation

Release 0.1

Tomaz Sifrer

May 19, 2018

Contents

1	Installation	3
2	Authentication	5
2.1	Contents:	5

An easy to use Python library for accessing the Twitch API

Warning: This documentation is a work in progress

Note: `python-twitch-client` currently supports Twitch API v5.

If you find a missing endpoint or a bug please raise an [issue](#) or contribute and open a [pull request](#).

CHAPTER 1

Installation

You can install `python-twitch-client` with `pip`:

```
$ pip install python-twitch-client
```


CHAPTER 2

Authentication

Before you can use Twitch API you need to get the client ID. To get one, you should follow the steps on [Twitch Authentication page](#).

Some of the endpoints also require OAuth token. To get one for testing purposes, you can use the free [tokengen tool](#).

There are two ways to pass credentials into the TwitchClient. The first and easiest way is to just pass the credentials as an argument:

```
client = TwitchClient(client_id='<my client id>', oauth_token='<my oauth token>')
```

Other option is to create a config file `~/.twitch.cfg` which is a text file formatted as .ini configuration file.

An example of the config file might look like:

```
[Credentials]
client_id = <my client id>
oauth_token = <my oauth token>
```

Note: You only need to provide `oauth_token` if you're calling endpoints that need it.

If you call functions that require `oauth_token` and you did not provide it, functions will raise `TwitchAuthException` exception.

2.1 Contents:

2.1.1 Basic Usage

The *python-twitch-client* allows you to easily access to Twitch API endpoints

This package is a modular wrapper designed to make Twitch API calls simpler and easier for you to use. Provided below are examples of how to interact with commonly used API endpoints, but this is by no means a complete list.

Getting a channel by ID

```
from twitch import TwitchClient

client = TwitchClient(client_id='<my client id>')
channel = client.channels.get_by_id(44322889)

print(channel.id)
print(channel.name)
print(channel.display_name)
```

2.1.2 Clips

class twitch.api.clips.Clips

This class provides methods for easy access to [Twitch Clips API](#).

classmethod get_by_slug(slug)

Gets a clip object based on the slug provided

Parameters slug(*string*) – Twitch Slug.

classmethod get_top(channel, cursor, game, language, limit, period, trending)

Gets all clips emoticons in one or more specified sets.

Parameters

- **channel** (*string*) – Channel name. If this is specified, top clips for only this channel are returned; otherwise, top clips for all channels are returned. If both channel and game are specified, game is ignored.
- **cursor** (*string*) – Tells the server where to start fetching the next set of results, in a multi-page response.
- **game** (*string*) – Game name. (Game names can be retrieved with the Search Games endpoint.) If this is specified, top clips for only this game are returned; otherwise, top clips for all games are returned. If both channel and game are specified, game is ignored.
- **language** (*string*) – Comma-separated list of languages, which constrains the languages of videos returned. Examples: es, en,es,th. If no language is specified, all languages are returned.
- **limit** (*int*) – Maximum number of most-recent objects to return. Default: 10. Maximum: 100.
- **period** (*string*) – The window of time to search for clips. Valid values: day, week, month, all. Default: week.
- **trending** (*boolean*) – If True, the clips returned are ordered by popularity; otherwise, by viewcount. Default: False.

classmethod get_followed()

Gets top clips for games followed by the user identified by OAuth token. Results are ordered by popularity.

2.1.3 Chat

class twitch.api.chat.Chat

This class provides methods for easy access to [Twitch Chat API](#).

classmethod get_badges_by_channel(channel_id)

Gets a list of badges that can be used in chat for a specified channel.

Parameters channel_id(string) – Channel ID.

classmethod get_emoticons_by_set(emotesets)

Gets all chat emoticons in one or more specified sets.

Parameters emotesets(list) – List of emoticon sets to be returned.

classmethod get_all_emoticons()

Gets all chat emoticons.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> emotes = client.chat.get_all_emoticons()
```

2.1.4 Channel feed

class twitch.api.channel_feed.ChannelFeed

This class provides methods for easy access to [Twitch Channel Feed API](#).

classmethod get_posts(channel_id, limit, cursor, comments)

Gets posts from a specified channel feed.

Parameters

- **channel_id**(string) – Channel ID
- **limit**(int) – Maximum number of objects to return. Default 10. Maximum 100.
- **cursor**(string) – Cursor of the next page
- **comments**(int) – Number of comments to return. Default 5. Maximum 5.

classmethod get_post(channel_id, post_id, comments)

Gets a specified post from a specified channel feed.

Parameters

- **channel_id**(string) – Channel ID
- **post_id**(string) – Post ID
- **comments**(int) – Number of comments to return. Default 5. Maximum 5.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> post = client.channel_feed.get_post('12345', '12345', comments=0)
```

classmethod create_post(channel_id, content, share)

Creates a post in a specified channel feed.

Parameters

- **channel_id**(string) – Channel ID
- **content**(string) – Content of the post

- **share** (*boolean*) – When set to true, the post is shared on the channel’s Twitter feed.

classmethod delete_post (*channel_id, post_id*)

Deletes a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID

classmethod create_reaction_to_post (*channel_id, post_id, emote_id*)

Creates a reaction to a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID
- **emote_id** (*string*) – Emote ID

classmethod delete_reaction_to_post (*channel_id, post_id, emote_id*)

Deletes a specified reaction to a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID
- **emote_id** (*string*) – Emote ID

classmethod get_post_comments (*channel_id, post_id, limit, cursor*)

Gets all comments on a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID
- **limit** (*int*) – Maximum number of objects to return. Default 10. Maximum 100.
- **cursor** (*string*) – Cursor of the next page

classmethod create_post_comment (*channel_id, post_id, content*)

Creates a comment to a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID
- **content** (*string*) – Content of the comment

classmethod delete_post_comment (*channel_id, post_id, comment_id*)

Deletes a specified comment on a specified post in a specified channel feed.

Parameters

- **channel_id** (*string*) – Channel ID
- **post_id** (*string*) – Post ID
- **comment_id** (*string*) – Comment ID

classmethod `create_reaction_to_comment(channel_id, post_id, comment_id, emote_id)`

Creates a reaction to a specified comment on a specified post in a specified channel feed.

Parameters

- `channel_id(string)` – Channel ID
- `post_id(string)` – Post ID
- `comment_id(string)` – Comment ID
- `emote_id(string)` – Emote ID

classmethod `delete_reaction_to_comment(channel_id, post_id, comment_id, emote_id)`

Deletes a reaction to a specified comment on a specified post in a specified channel feed.

Parameters

- `channel_id(string)` – Channel ID
- `post_id(string)` – Post ID
- `comment_id(string)` – Comment ID
- `emote_id(string)` – Emote ID

2.1.5 Channels

class `twitch.api.channels.Channels`

This class provides methods for easy access to [Twitch Channels API](#).

classmethod `get()`

Gets a channel object based on the OAuth token.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>', '<my oauth token>')
>>> channel = client.channels.get()
```

classmethod `get_by_id(channel_id)`

Gets a specified channel object.

Parameters `channel_id(string)` – Channel ID

classmethod `update(channel_id, status, game, delay, channel_feed_enabled)`

Updates specified properties of a specified channel.

Parameters

- `channel_id(string)` – Channel ID
- `status(string)` – Description of the broadcaster's status.
- `game(string)` – Name of game.
- `delay(int)` – Channel delay, in seconds.
- `channel_feed_enabled(boolean)` – If true, the channel's feed is turned on.

classmethod `get_editors(channel_id)`

Gets a list of users who are editors for a specified channel.

Parameters `channel_id(string)` – Channel ID

classmethod `get_followers(channel_id, limit, offset, cursor, direction)`

Gets a list of users who follow a specified channel.

Parameters

- **channel_id** (*string*) – Channel ID
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **cursor** (*string*) – Cursor of the next page.
- **direction** (*string*) – Direction of sorting.

classmethod **get_teams** (*channel_id*)

Gets a list of teams to which a specified channel belongs.

Parameters **channel_id** (*string*) – Channel ID

classmethod **get_subscribers** (*channel_id, limit, offset, direction*)

Gets a list of users subscribed to a specified channel.

Parameters

- **channel_id** (*string*) – Channel ID
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **direction** (*string*) – Direction of sorting.

classmethod **check_subscription_by_user** (*channel_id, user_id*)

Checks if a specified channel has a specified user subscribed to it.

Parameters

- **channel_id** (*string*) – Channel ID
- **user_id** (*string*) – User ID

classmethod **get_videos** (*channel_id, limit, offset, broadcast_type, language, sort*)

Gets a list of videos from a specified channel.

Parameters

- **channel_id** (*string*) – Channel ID
- **limit** (*int*) – Maximum number of objects to return. Default 10. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **broadcast_type** (*string*) – Constrains the type of videos returned.
- **language** (*string*) – Constrains the language of the videos that are returned.
- **sort** (*string*) – Sorting order of the returned objects.

classmethod **start_commercial** (*channel_id, duration*)

Starts a commercial (advertisement) on a specified channel.

Parameters

- **channel_id** (*string*) – Channel ID
- **duration** (*string*) – Duration of the commercial in seconds. Default 30.

classmethod **reset_stream_key** (*channel_id*)

Deletes the stream key for a specified channel. Stream key is automatically reset.

Parameters **channel_id** (*string*) – Channel ID

classmethod `get_community(channel_id)`

Gets the community for a specified channel.

Parameters `channel_id(string)` – Channel ID

classmethod `set_community(channel_id, community_id)`

Sets a specified channel to be in a specified community.

Parameters

- `channel_id(string)` – Channel ID
- `community_id(string)` – Community ID

classmethod `delete_from_community(channel_id)`

Deletes a specified channel from its community.

Parameters `channel_id(string)` – Channel ID

2.1.6 Collections

class `twitch.api.collections.Collections`

This class provides methods for easy access to [Twitch Collections API](#).

classmethod `get_metadata(collection_id)`

Gets summary information about a specified collection.

Parameters `collection_id(string)` – Collection ID

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> collection = client.collections.get_metadata('12345')
```

classmethod `get(collection_id, include_all_items)`

Gets all items in a specified collection.

Parameters

- `collection_id(string)` – Collection ID
- `include_all_items(boolean)` – If True, unwatchable VODs are included in the response. Default: false.

classmethod `get_by_channel(channel_id, limit, cursor, containig_item)`

Gets all collections owned by a specified channel.

Parameters

- `channel_id(string)` – Channel ID
- `limit(int)` – Maximum number of objects to return. Default 10. Maximum 100.
- `cursor(string)` – Cursor of the next page
- `containig_item(string)` – Returns only collections containing the specified video.
Example: video:89917098.

classmethod `create(channel_id, title)`

Creates a new collection owned by a specified channel.

Parameters

- `channel_id(string)` – Channel ID

- **title** (*string*) – Collection title

classmethod update (*collection_id*, *title*)

Updates the title of a specified collection.

Parameters

- **collection_id** (*string*) – Collection ID
- **title** (*string*) – Collection title

classmethod create_thumbnail (*collection_id*, *item_id*)

Adds the thumbnail of a specified collection item as the thumbnail for the specified collection.

Parameters

- **collection_id** (*string*) – Collection ID
- **item_id** (*string*) – Item ID

classmethod delete (*collection_id*)

Deletes a specified collection.

Parameters **collection_id** (*string*) – Collection ID

classmethod add_item (*collection_id*, *item_id*, *item_type*)

Adds a specified item to a specified collection.

Parameters

- **collection_id** (*string*) – Collection ID
- **item_id** (*string*) – Item ID
- **item_type** (*string*) – Type of the item. Example: *video*.

classmethod delete_item (*collection_id*, *collection_item_id*)

Deletes a specified collection item from a specified collection.

Parameters

- **collection_id** (*string*) – Collection ID
- **collection_item_id** (*string*) – Collection Item ID

classmethod move_item (*collection_id*, *collection_item_id*, *position*)

Deletes a specified collection item from a specified collection.

Parameters

- **collection_id** (*string*) – Collection ID
- **collection_item_id** (*string*) – Collection Item ID
- **position** (*int*) – New item position

2.1.7 Communities

class twitch.api.communities.**Communities**

This class provides methods for easy access to [Twitch Communities API](#).

classmethod get_by_name (*community_name*)

Gets a specified community.

Parameters **community_name** (*string*) – Name of the community

classmethod `get_by_id (community_id)`

Gets a Community object based on specified user id.

Parameters `community_id (string)` – Community ID

classmethod `create (name, summary, description, rules)`

Creates a community.

Parameters

- **name** (*string*) – Community name.
- **summary** (*string*) – Short description of the community.
- **description** (*string*) – Long description of the community.
- **rules** (*string*) – Rules displayed when viewing a community page.

classmethod `update (community_id, summary, description, rules, email)`

Updates a community.

Parameters

- **community_id** (*string*) – Community ID
- **summary** (*string*) – Short description of the community.
- **description** (*string*) – Long description of the community.
- **rules** (*string*) – Rules displayed when viewing a community page.
- **email** (*string*) – Email address of the community owner.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>', '<my oauth token>')
>>> community = client.communities.update(12345, 'foo', 'bar')
```

2.1.8 Games

class `twitch.api.games.Games`

This class provides methods for easy access to [Twitch Games API](#).

classmethod `get_top (limit, offset)`

Gets a list of games sorted by number of current viewers.

Parameters

- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> games = client.games.get_top()
```

2.1.9 Ingests

class `twitch.api.ingests.Ingests`

This class provides methods for easy access to [Twitch Ingests API](#).

classmethod `get_server_list ()`

Gets a list of ingest servers.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> ingests = client.ingests.get_server_list()
```

2.1.10 Search

class twitch.api.search.**Search**

This class provides methods for easy access to [Twitch Search API](#).

classmethod **channels** (*query*, *limit*, *offset*)

Searches for channels based on a specified query parameter.

Parameters

- **query** (*string*) – Search query
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> channels = client.search.channels('lirik', limit=69, offset=420)
```

classmethod **games** (*query*, *live*)

Searches for games based on a specified query parameter.

Parameters

- **query** (*string*) – Search query
- **live** (*boolean*) – If *True*, returns only games that are live on at least one channel. Default: *False*.

classmethod **streams** (*query*, *limit*, *offset*, *hls*)

Searches for streams based on a specified query parameter.

Parameters

- **query** (*string*) – Search query
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **hls** (*boolean*) – If *True*, returns only HLS streams; if *False*, only RTMP streams; if *None*, both HLS and RTMP streams.

2.1.11 Streams

class twitch.api.streams.**Streams**

This class provides methods for easy access to [Twitch Streams API](#).

classmethod **get_stream_by_user** (*channel_id*, *stream_type*)

Gets stream information for a specified user.

Parameters

- **channel_id** (*string*) – ID of the channel you want to get information of

- **stream_type** (*string*) – Constrains the type of streams returned. Default `STREAM_TYPE_LIVE`.

classmethod `get_live_streams` (*channel, game, language, stream_type, limit, offset*)

Gets a list of live streams.

Parameters

- **channel** (*string*) – Comma-separated list of channel IDs you want to get
- **game** (*string*) – Game of the streams returned
- **language** (*string*) – Constrains the language of the streams returned
- **stream_type** (*string*) – Constrains the type of streams returned. Default `STREAM_TYPE_LIVE`.
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

classmethod `get_summary` (*game*)

Gets a list of summaries of live streams.

Parameters **game** (*string*) – Game of the streams returned

classmethod `get_featured` (*limit, offset*)

Gets a list of all featured live streams.

Parameters

- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

classmethod `get_followed` (*stream_type, limit, offset*)

Gets a list of online streams a user is following, based on a specified OAuth token.

Parameters

- **stream_type** (*string*) – Constrains the type of streams returned. Default `STREAM_TYPE_LIVE`.
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>', '<my oauth token>')
>>> followed = client.streams.get_followed()
```

classmethod `get_streams_in_community` (*community_id*)

Gets a list of streams in a community. (From Twitch forum [Communities API Release](#))

Parameters **community_id** (*string*) – Community ID

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> streams = client.streams.get_streams_in_community('5181e78f-2280-42a6-
↳873d-758e25a7c313')
```

2.1.12 Teams

class twitch.api.teams.Teams

This class provides methods for easy access to [Twitch Teams API](#).

classmethod get(*team_name*)

Gets a specified team object.

Parameters *team_name* (*string*) – Name of the team you want to get information of

classmethod get_all(*limit*, *offset*)

Gets all active teams.

Parameters

- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> teams = client.teams.get_all()
```

2.1.13 Users

class twitch.api.users.Users

This class provides methods for easy access to [Twitch Users API](#).

classmethod get()

Gets a user object based on the OAuth token provided.

classmethod get_by_id(*user_id*)

Gets a user object based on specified user id.

Parameters *user_id* (*string*) – User ID

classmethod get_emotes(*user_id*)

Gets a list of the emojis and emoticons that the specified user can use in chat

Parameters *user_id* (*string*) – User ID

classmethod check_subscribed_to_channel(*user_id*, *channel_id*)

Checks if a specified user is subscribed to a specified channel.

Parameters

- **user_id** (*string*) – User ID
- **channel_id** (*string*) – ID of the channel you want to check if user is subscribed to

classmethod get_follows(*user_id*, *limit*, *offset*, *direction*, *sort_by*)

Gets a list of all channels followed by a specified user.

Parameters

- **user_id** (*string*) – User ID
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **direction** (*string*) – Sorting direction. Default DIRECTION_DESC.
- **sort_by** (*string*) – Sorting key. Default USERS_SORT_BY_CREATED_AT.

classmethod check_follows_channel (*user_id*, *channel_id*)

Checks if a specified user follows a specified channel.

Parameters

- **user_id** (*string*) – User ID
- **channel_id** (*string*) – ID of the channel you want to check if user is following

classmethod follow_channel (*user_id*, *channel_id*, *notifications*)

Adds a specified user to the followers of a specified channel.

Parameters

- **user_id** (*string*) – User ID
- **channel_id** (*string*) – ID of the channel you want user to follow
- **notifications** (*boolean*) – If true, the user gets email or push notifications when the channel goes live. Default False.

classmethod unfollow_channel (*user_id*, *channel_id*)

Deletes a specified user from the followers of a specified channel.

Parameters

- **user_id** (*string*) – User ID
- **channel_id** (*string*) – ID of the channel you want user to unfollow

classmethod get_user_block_list (*user_id*, *limit*, *offset*)

Gets a user's block list.

Parameters

- **user_id** (*string*) – User ID
- **limit** (*int*) – Maximum number of objects to return. Default 25. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.

classmethod block_user (*user_id*, *blocked_user_id*)

Blocks a user.

Parameters

- **user_id** (*string*) – User ID
- **blocked_user_id** (*string*) – ID of the user you wish to block

classmethod unblock_user (*user_id*, *blocked_user_id*)

Unblocks a user.

Parameters

- **user_id** (*string*) – User ID
- **blocked_user_id** (*string*) – ID of the user you wish to unblock

classmethod translate_usernames_to_ids (*usernames*)

Unblocks a user.

Parameters usernames (*list[string]*) – List of usernames you wish to get ID's of

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>')
>>> users = client.users.translate_usernames_to_ids(['lirik', 'giantwaffle'])
>>>
>>> for user in users:
>>>     print('%s: %s' % (user.name, user.id))
'lirik: 23161357'
'giantwaffle: 22552479'
```

2.1.14 Videos

class twitch.api.videos.Videos

This class provides methods for easy access to [Twitch Videos API](#).

classmethod get_by_id(*video_id*)

Gets a Video object based on specified video ID.

Parameters *video_id* (*string*) – Video ID

classmethod get_top(*limit, offset, game, period, broadcast_type*)

Gets a list of top videos.

Parameters

- **limit** (*int*) – Maximum number of objects to return. Default 10. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **game** (*string*) – Name of the game.
- **period** (*string*) – Window of time to search. Default PERIOD_WEEK.
- **broadcast_type** (*string*) – Type of broadcast returned. Default BROADCAST_TYPE_HIGHLIGHT.

classmethod get_followed_videos(*limit, offset, broadcast_type*)

Gets a list of followed videos based on a specified OAuth token.

Parameters

- **limit** (*int*) – Maximum number of objects to return. Default 10. Maximum 100.
- **offset** (*int*) – Object offset for pagination of result. Default 0.
- **broadcast_type** (*string*) – Type of broadcast returned. Default BROADCAST_TYPE_HIGHLIGHT.

```
>>> from twitch import TwitchClient
>>> client = TwitchClient('<my client id>', '<my oauth token>')
>>> videos = client.videos.get_followed_videos()
```

A

`add_item()` (twitch.api.collections.Collections class method), 12

B

`block_user()` (twitch.api.users.Users class method), 17

C

`ChannelFeed` (class in twitch.api.channel_feed), 7

`Channels` (class in twitch.api.channels), 9

`channels()` (twitch.api.search.Search class method), 14

`Chat` (class in twitch.api.chat), 7

`check_follows_channel()` (twitch.api.users.Users class method), 16

`check_subscribed_to_channel()` (twitch.api.users.Users class method), 16

`check_subscription_by_user()`
(twitch.api.channels.Channels class method), 10

`Clips` (class in twitch.api.clips), 6

`Collections` (class in twitch.api.collections), 11

`Communities` (class in twitch.api.communities), 12

`create()` (twitch.api.collections.Collections class method), 11

`create()` (twitch.api.communities.Communities class method), 13

`create_post()` (twitch.api.channel_feed.ChannelFeed class method), 7

`create_post_comment()` (twitch.api.channel_feed.ChannelFeed class method), 8

`create_reaction_to_comment()`
(twitch.api.channel_feed.ChannelFeed class method), 8

`create_reaction_to_post()`
(twitch.api.channel_feed.ChannelFeed class method), 8

`create_thumbnail()` (twitch.api.collections.Collections class method), 12

D

`delete()` (twitch.api.collections.Collections class method), 12

`delete_from_community()` (twitch.api.channels.Channels class method), 11

`delete_item()` (twitch.api.collections.Collections class method), 12

`delete_post()` (twitch.api.channel_feed.ChannelFeed class method), 8

`delete_post_comment()` (twitch.api.channel_feed.ChannelFeed class method), 8

`delete_reaction_to_comment()`
(twitch.api.channel_feed.ChannelFeed class method), 9

`delete_reaction_to_post()`
(twitch.api.channel_feed.ChannelFeed class method), 8

F

`follow_channel()` (twitch.api.users.Users class method), 17

G

`Games` (class in twitch.api.games), 13

`games()` (twitch.api.search.Search class method), 14

`get()` (twitch.api.channels.Channels class method), 9

`get()` (twitch.api.collections.Collections class method), 11

`get()` (twitch.api.teams.Teams class method), 16

`get()` (twitch.api.users.Users class method), 16

`get_all()` (twitch.api.teams.Teams class method), 16

`get_all_emoticons()` (twitch.api.chat.Chat class method), 7

`get_badges_by_channel()` (twitch.api.chat.Chat class method), 7

`get_by_channel()` (twitch.api.collections.Collections class method), 11

`get_by_id()` (twitch.api.channels.Channels class method), 9

`get_by_id()` (twitch.api.communities.Communities class method), 12

`get_by_id()` (twitch.api.users.Users class method), 16
`get_by_id()` (twitch.api.videos.Videos class method), 18
`get_by_name()` (twitch.api.communities.Communities class method), 12
`get_by_slug()` (twitch.api.clips.Clips class method), 6
`get_community()` (twitch.api.channels.Channels class method), 10
`get_editors()` (twitch.api.channels.Channels class method), 9
`get_emotes()` (twitch.api.users.Users class method), 16
`get_emoticons_by_set()` (twitch.api.chat.Chat class method), 7
`get_featured()` (twitch.api.streams.Streams class method), 15
`get_followed()` (twitch.api.clips.Clips class method), 6
`get_followed()` (twitch.api.streams.Streams class method), 15
`get_followed_videos()` (twitch.api.videos.Videos class method), 18
`get_followers()` (twitch.api.channels.Channels class method), 9
`get_follows()` (twitch.api.users.Users class method), 16
`get_live_streams()` (twitch.api.streams.Streams class method), 15
`get_metadata()` (twitch.api.collections.Collections class method), 11
`get_post()` (twitch.api.channel_feed.ChannelFeed class method), 7
`get_post_comments()` (twitch.api.channel_feed.ChannelFeed class method), 8
`get_posts()` (twitch.api.channel_feed.ChannelFeed class method), 7
`get_server_list()` (twitch.api.ingests.Ingests class method), 13
`get_stream_by_user()` (twitch.api.streams.Streams class method), 14
`get_streams_in_community()` (twitch.api.streams.Streams class method), 15
`get_subscribers()` (twitch.api.channels.Channels class method), 10
`get_summary()` (twitch.api.streams.Streams class method), 15
`get_teams()` (twitch.api.channels.Channels class method), 10
`get_top()` (twitch.api.clips.Clips class method), 6
`get_top()` (twitch.api.games.Games class method), 13
`get_top()` (twitch.api.videos.Videos class method), 18
`get_user_block_list()` (twitch.api.users.Users class method), 17
`get_videos()` (twitch.api.channels.Channels class method), 10

I
Ingests (class in twitch.api.ingests), 13

M
`move_item()` (twitch.api.collections.Collections class method), 12

R
`reset_stream_key()` (twitch.api.channels.Channels class method), 10

S
Search (class in twitch.api.search), 14
`set_community()` (twitch.api.channels.Channels class method), 11
`start_commercial()` (twitch.api.channels.Channels class method), 10
Streams (class in twitch.api.streams), 14
`streams()` (twitch.api.search.Search class method), 14

T
Teams (class in twitch.api.teams), 16
`translate_usernames_to_ids()` (twitch.api.users.Users class method), 17

U
`unblock_user()` (twitch.api.users.Users class method), 17
`unfollow_channel()` (twitch.api.users.Users class method), 17
`update()` (twitch.api.channels.Channels class method), 9
`update()` (twitch.api.collections.Collections class method), 12
`update()` (twitch.api.communities.Communities class method), 13
Users (class in twitch.api.users), 16

V
Videos (class in twitch.api.videos), 18