
stream-python Documentation

Release 0.1.0

Thierry Schellenbach

November 28, 2014

1	stream package	3
1.1	Submodules	3
1.2	stream.client module	3
1.3	stream.exceptions module	4
1.4	stream.feed module	5
1.5	stream.signing module	6
1.6	stream.tests module	6
1.7	stream.utils module	6
1.8	Module contents	6
2	Indices and tables	7
	Python Module Index	9

Contents:

stream package

1.1 Submodules

1.2 stream.client module

class `stream.client.StreamClient` (*api_key, api_secret, app_id, version='v1.0', timeout=3.0, base_url=None*)

Bases: `object`

base_url = `'https://getstream.io/api/'`

delete (**args, **kwargs*)
Shortcut for make request

feed (*feed_slug, user_id*)
Returns a Feed object

Parameters

- **feed_slug** – the slug of the feed
- **user_id** – the user id

get (**args, **kwargs*)
Shortcut for make request

get_default_params ()
Returns the params with the API key present

get_full_url (*relative_url*)

get_user_agent ()

post (**args, **kwargs*)
Shortcut for make request

raise_exception (*result, status_code*)
Map the exception code to an exception class and raise it If `result.exception` and `result.detail` are available use that Otherwise just raise a generic error

1.3 stream.exceptions module

exception `stream.exceptions.ApiKeyException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when there is an issue with your Access Key

code = 2

status_code = 401

exception `stream.exceptions.CustomFieldException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when there are missing or misconfigured custom fields

code = 5

status_code = 400

exception `stream.exceptions.FeedConfigException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when there are missing or misconfigured custom fields

code = 6

status_code = 400

exception `stream.exceptions.InputException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when you send the wrong data to the API

code = 4

status_code = 400

exception `stream.exceptions.SignatureException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when there is an issue with the signature you provided

code = 3

status_code = 401

exception `stream.exceptions.SiteSuspendedException` (*error_message, status_code=None*)

Bases: `stream.exceptions.StreamApiException`

Raised when the site requesting the data is suspended

code = 7

status_code = 401

exception `stream.exceptions.StreamApiException` (*error_message, status_code=None*)

Bases: `exceptions.Exception`

code = 1

`stream.exceptions.get_exception_dict()`

`stream.exceptions.get_exceptions()`

1.4 stream.feed module

class `stream.feed.Feed`(*client, feed_slug, user_id, token*)

Bases: `object`

add_activities(*activity_list*)

Adds a list of activities to the feed

Parameters `activity_list` – a list with the activity data dicts

Example:

```
activity_data = [
    {'actor': 1, 'verb': 'tweet', 'object': 1},
    {'actor': 2, 'verb': 'watch', 'object': 2},
]
result = feed.add_activities(activity_data)
```

add_activity(*activity_data*)

Adds an activity to the feed, this will also trigger an update to all the feeds which follow this feed

Parameters `activity_data` – a dict with the activity data

Example:

```
activity_data = {'actor': 1, 'verb': 'tweet', 'object': 1}
activity_id = feed.add_activity(activity_data)
```

add_to_signature(*recipients*)

Takes a list of recipients such as ['user:1', 'user:2'] and turns it into a list with the tokens included ['user:1 token', 'user:2 token']

follow(*target_feed_slug, target_user_id*)

Follows the given feed

Parameters

- **target_feed_slug** – the slug of the target feed
- **target_user_id** – the user id

followers(*offset=0, limit=25, feeds=None*)

Lists the followers for the given feed

following(*offset=0, limit=25, feeds=None*)

List the feeds which this feed is following

get(***params*)

Get the activities in this feed

Example:

```
# fast pagination using id filtering
feed.get(limit=10, id_lte=100292310)

# slow pagination using offset
feed.get(limit=10, offset=10)
```

remove_activity(*activity_id=None, foreign_id=None*)

Removes an activity from the feed

Parameters `activity_id` – the activity id to remove from this feed

(note this will also remove the activity from feeds which follow this feed) :param foreign_id: the foreign id you provided when adding the activity

unfollow (*target_feed_slug*, *target_user_id*)
Unfollow the given feed

1.5 stream.signing module

`stream.signing.b64_encode` (*s*)

`stream.signing.sign` (*api_secret*, *feed_id*)
Base64 encoded sha1 signature

Parameters

- **api_secret** – the api secret
- **feed_id** – the feed_id to sign (commonly a feed id such as user1)

Example:: signature = sign('secret', 'user1')

1.6 stream.tests module

1.7 stream.utils module

`stream.utils.validate_feed_id` (*feed_id*)
Validates the input is in the format of user:1

Parameters **feed_id** – a feed such as user:1

Raises ValueError if the format doesnt match

`stream.utils.validate_feed_slug` (*feed_slug*)
Validates the feed slug falls into w

`stream.utils.validate_user_id` (*user_id*)
Validates the user id falls into w

1.8 Module contents

`stream.connect` (*api_key=None*, *api_secret=None*, *app_id=None*, *version='v1.0'*, *timeout=3.0*)
Returns a Client object

Parameters

- **api_key** – your api key or heroku url
- **api_secret** – the api secret
- **app_id** – the app id (used for listening to feed changes)

Indices and tables

- *genindex*
- *modindex*
- *search*

S

stream, 6
stream.client, 3
stream.exceptions, 4
stream.feed, 5
stream.signing, 6
stream.utils, 6

A

add_activities() (stream.feed.Feed method), 5
 add_activity() (stream.feed.Feed method), 5
 add_to_signature() (stream.feed.Feed method), 5
 ApiKeyException, 4

B

b64_encode() (in module stream.signing), 6
 base_url (stream.client.StreamClient attribute), 3

C

code (stream.exceptions.ApiKeyException attribute), 4
 code (stream.exceptions.CustomFieldException attribute), 4
 code (stream.exceptions.FeedConfigException attribute), 4
 code (stream.exceptions.InputException attribute), 4
 code (stream.exceptions.SignatureException attribute), 4
 code (stream.exceptions.SiteSuspendedException attribute), 4
 code (stream.exceptions.StreamApiException attribute), 4
 connect() (in module stream), 6
 CustomFieldException, 4

D

delete() (stream.client.StreamClient method), 3

F

Feed (class in stream.feed), 5
 feed() (stream.client.StreamClient method), 3
 FeedConfigException, 4
 follow() (stream.feed.Feed method), 5
 followers() (stream.feed.Feed method), 5
 following() (stream.feed.Feed method), 5

G

get() (stream.client.StreamClient method), 3
 get() (stream.feed.Feed method), 5

get_default_params() (stream.client.StreamClient method), 3
 get_exception_dict() (in module stream.exceptions), 4
 get_exceptions() (in module stream.exceptions), 4
 get_full_url() (stream.client.StreamClient method), 3
 get_user_agent() (stream.client.StreamClient method), 3

I

InputException, 4

P

post() (stream.client.StreamClient method), 3

R

raise_exception() (stream.client.StreamClient method), 3
 remove_activity() (stream.feed.Feed method), 5

S

sign() (in module stream.signing), 6
 SignatureException, 4
 SiteSuspendedException, 4
 status_code (stream.exceptions.ApiKeyException attribute), 4
 status_code (stream.exceptions.CustomFieldException attribute), 4
 status_code (stream.exceptions.FeedConfigException attribute), 4
 status_code (stream.exceptions.InputException attribute), 4
 status_code (stream.exceptions.SignatureException attribute), 4
 status_code (stream.exceptions.SiteSuspendedException attribute), 4
 stream (module), 6
 stream.client (module), 3
 stream.exceptions (module), 4
 stream.feed (module), 5
 stream.signing (module), 6
 stream.utils (module), 6
 StreamApiException, 4

StreamClient (class in stream.client), 3

U

unfollow() (stream.feed.Feed method), 6

V

validate_feed_id() (in module stream.utils), 6

validate_feed_slug() (in module stream.utils), 6

validate_user_id() (in module stream.utils), 6