
twitchAPI

Release 1.0

Oct 08, 2020

Contents:

1	Installation	3
2	Indices and tables	5
2.1	twitchAPI.twitch	5
2.2	twitchAPI.webhook	16
2.3	twitchAPI.oauth	19
2.4	twitchAPI.types	19
2.5	twitchAPI.helper	22
	Python Module Index	25
	Index	27

This is a full implementation of the Twitch Helix API and its Webhook in python 3.7.

On Github: <https://github.com/Teekeks/pyTwitchAPI>

On PyPi: <https://pypi.org/project/twitchAPI/>

CHAPTER 1

Installation

Install using pip:

```
`pip install twitchAPI`
```


- `genindex`
- `modindex`
- `search`

<code>twitchAPI.twitch</code>	The Twitch API client.
<code>twitchAPI.webhook</code>	
<code>twitchAPI.oauth</code>	User OAuth Authenticator and helper functions
<code>twitchAPI.types</code>	
<code>twitchAPI.helper</code>	

2.1 twitchAPI.twitch

The Twitch API client.

class `twitchAPI.twitch.Twitch` (*app_id: str, app_secret: str*)

Twitch API client

Parameters

- **app_id** (*str*) – Your app id
- **app_secret** (*str*) – Your app secret

authenticate_app (*scope: List[twitchAPI.types.AuthScope]*) → None

Authenticate with a fresh generated app token

Parameters **scope** – List of `twitchAPI.types.AuthScope` to use

Returns None

auto_refresh_auth = True

If set to true, auto refresh the auth token once it expires

check_automod_status (*broadcaster_id: str, msg_id: str, msg_text: str, user_id: str*) → dict

Requires User authentication with scope MODERATION_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#check-automod-status>

Parameters

- **broadcaster_id** – str
- **msg_id** – str
- **msg_text** – str
- **user_id** – str

Return type dict

create_clip (*broadcaster_id: str, has_delay: bool = False*) → dict

Requires User authentication with scope CLIPS_EDIT

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#create-clip>

Parameters

- **broadcaster_id** – str
- **has_delay** – optional bool

Return type dict

create_entitlement_grants_upload_url (*manifest_id: str*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#create-entitlement-grants-upload-url>

Parameters **manifest_id** – str

Return type dict

create_stream_marker (*user_id: str, description: Optional[str] = None*) → dict

Requires User authentication with scope USER_EDIT_BROADCAST

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#create-stream-marker>

Parameters

- **user_id** – str
- **description** – optional str with max length of 140

Return type dict

create_user_follows (*from_id: str, to_id: str, allow_notifications: Optional[bool] = False*) → bool

Requires User authentication with AuthScope.USER_EDIT_FOLLOWS

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#create-user-follows>

Parameters

- **from_id** – str
- **to_id** – str
- **allow_notifications** – optional bool

Return type bool

delete_user_follows (*from_id: str, to_id: str*) → bool

Requires User authentication with AuthScope.USER_EDIT_FOLLOWS

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#delete-user-follows>

Parameters

- **from_id** – str
- **to_id** – str

Return type bool

get_all_stream_tags (*after: Optional[str] = None, first: int = 20, tag_ids: Optional[List[str]] = None*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-all-stream-tags>

Parameters

- **after** – optional str
- **first** – optional int in range 1 to 100
- **tag_ids** – optional list str with maximum 100 entries

Return type dict

get_app_token () → Optional[str]

Returns the app token that the api uses or None when not authenticated.

Returns app token

Return type Union[str, None]

get_banned_events (*broadcaster_id: str, user_id: Optional[str] = None, after: Optional[str] = None, first: int = 20*) → dict

Requires User authentication with scope MODERATION_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-banned-events>

Parameters

- **broadcaster_id** – str
- **user_id** – optional str
- **after** – optional str
- **first** – optional int in range 1 to 100

Return type dict

get_banned_users (*broadcaster_id: str, user_id: Optional[str] = None, after: Optional[str] = None, before: Optional[str] = None*) → dict

Requires User authentication with scope MODERATION_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-banned-users>

Parameters

- **broadcaster_id** – str
- **user_id** – optional str
- **after** – optional str
- **before** – optional str

Return type dict

get_bits_leaderboard (*count*: int = 10, *period*: twitchAPI.types.TimePeriod = <TimePeriod.ALL: 'all'>, *started_at*: Optional[datetime.datetime] = None, *user_id*: Optional[str] = None) → dict

Requires User authentication with scope BITS_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-bits-leaderboard>

Parameters

- **count** – optional int in range 1 to 100
- **period** – optional *TimePeriod*
- **started_at** – optional datetime
- **user_id** – optional str

Return type dict

get_broadcaster_subscriptions (*broadcaster_id*: str, *user_ids*: Optional[List[str]] = None) → dict

Requires User authentication with scope CHANNEL_READ_SUBSCRIPTIONS

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-broadcaster-subscriptions>

Parameters

- **broadcaster_id** – str
- **user_ids** – optional list of str with maximum 100 entries

Return type dict

get_channel_information (*broadcaster_id*: str) → dict

Requires App or user authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-channel-information>

Parameters **broadcaster_id** – str

:rtype dict

get_cheermotes (*broadcaster_id*: str) → dict

Requires App or User authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-cheermotes>

Parameters **broadcaster_id** – str

Return type dict

get_clips (*broadcaster_id*: str, *game_id*: str, *clip_id*: List[str], *after*: Optional[str] = None, *before*: Optional[str] = None, *ended_at*: Optional[datetime.datetime] = None, *started_at*: Optional[datetime.datetime] = None) → dict

Requires no authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-clips>

Parameters

- **broadcaster_id** – str
- **game_id** – str
- **clip_id** – list of str

- **after** – optional str
- **before** – optional str
- **ended_at** – optional datetime
- **started_at** – optional datetime

Return type dict

get_code_status (*code: List[str], user_id: int*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-code-status>

Parameters

- **code** – list of str, maximum of 20 entries
- **user_id** – int

Return type dict

get_extension_analytics (*after: Optional[str] = None, extension_id: Optional[str] = None, first: int = 20, ended_at: Optional[datetime.datetime] = None, started_at: Optional[datetime.datetime] = None, report_type: Optional[twitchAPI.types.AnalyticsReportType] = None*) → dict

Requires User authentication with scope `twitchAPI.types.AuthScope.ANALYTICS_READ_EXTENSION`

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-extension-analytics>

Parameters

- **after** (*str*) – cursor for forward pagination
- **extension_id** (*str*) – If this is specified, the returned URL points to an analytics report for just the specified extension.
- **first** (*int*) – Maximum number of objects returned, range 1 to 100, default 20
- **ended_at** (*datetime.datetime*) – Ending date/time for returned reports, if this is provided, *started_at* must also be specified.
- **started_at** (*datetime.datetime*) – Starting date/time for returned reports, if this is provided, *ended_at* must also be specified.
- **report_type** (*AnalyticsReportType*) – Type of analytics report that is returned

Return type dict

Raises `twitchAPI.types.UnauthorizedException`, `twitchAPI.types.MissingScopeException`, `ValueError`

get_extension_transactions (*extension_id: str, transaction_id: Optional[str] = None, after: Optional[str] = None, first: int = 20*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-extension-transactions>

Parameters

- **extension_id** – str
- **transaction_id** – optional str
- **after** – optional str

- **first** – optional int in range 1 to 100

Return type dict

get_game_analytics (*after: Optional[str] = None, first: int = 20, game_id: Optional[str] = None, ended_at: Optional[datetime.datetime] = None, started_at: Optional[datetime.datetime] = None, report_type: Optional[twitchAPI.types.AnalyticsReportType] = None*) → dict

Requires User authentication with scope ANALYTICS_READ_GAMES

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-game-analytics>

Parameters

- **after** – optional str
- **first** – optional int in range 1 to 100
- **game_id** – optional str
- **ended_at** – optional datetime
- **started_at** – optional datetime
- **report_type** – optional *twitchAPI.types.AnalyticsReportType*

Return type dict

get_games (*game_ids: Optional[List[str]] = None, names: Optional[List[str]] = None*) → dict

Requires no authentication. In total, only 100 game ids and names can be fetched at once.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-games>

Parameters

- **game_ids** – optional list of str
- **names** – optional list of str

Return type dict

get_hype_train_events (*broadcaster_id: str, first: Optional[int] = 1, id: Optional[str] = None, cursor: Optional[str] = None*) → dict

Requires App or User authentication with AuthScope.CHANNEL_READ_HYPE_TRAIN

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-hype-train-events>

Parameters

- **broadcaster_id** – str
- **first** – optional int, default 1, max 100
- **id** – optional str
- **cursor** – optional str

Return type dict

get_moderator_events (*broadcaster_id: str, user_id: Optional[str] = None*) → dict

Requires User authentication with scope MODERATION_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-moderator-events>

Parameters

- **broadcaster_id** – str
- **user_id** – optional str

Return type dict

get_moderators (*broadcaster_id: str, user_id: Optional[str] = None, after: Optional[str] = None*)
→ dict

Requires User authentication with scope MODERATION_READ

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-moderators>

Parameters

- **broadcaster_id** – str
- **user_id** – optional str
- **after** – optional str

Return type dict

get_stream_key (*broadcaster_id: str*) → dict

Requires User authentication with AuthScope.CHANNEL_READ_STREAM_KEY

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-stream-key>

Parameters **broadcaster_id** – str

Return type dict

get_stream_markers (*user_id: str, video_id: str, after: Optional[str] = None, before: Optional[str] = None, first: int = 20*) → dict

Requires User authentication with scope USER_READ_BROADCAST

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-stream-markers>

Parameters

- **user_id** – str
- **video_id** – str
- **after** – optional str
- **before** – optional str
- **first** – optional int in range 1 to 100

Return type dict

get_stream_tags (*broadcaster_id: str*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-stream-tags>

Parameters **broadcaster_id** – str, id of streamer

Return type dict

get_streams (*after: Optional[str] = None, before: Optional[str] = None, first: int = 20, game_id: Optional[str] = None, language: Optional[List[str]] = None, user_id: Optional[List[str]] = None, user_login: Optional[List[str]] = None*) → dict

Requires no authentication.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-streams>

Parameters

- **after** – optional str
- **before** – optional str
- **first** – optional int in range 1 to 100

- **game_id** – optional str
- **language** – optional list of str with max length of 100
- **user_id** – optional list of str with max length of 100
- **user_login** – optional list of str with max length of 100

Return type dict

get_streams_metadata (*after: Optional[str] = None, before: Optional[str] = None, first: int = 20, game_id: Optional[List[str]] = None, language: Optional[List[str]] = None, user_id: Optional[List[str]] = None, user_login: Optional[List[str]] = None*) → dict

Requires no authentication.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-streams-metadata>

Parameters

- **after** – optional str
- **before** – optional str
- **first** – optional int in range between 1 and 100
- **game_id** – optional list of str with maximum 100 entries
- **language** – optional list of str with maximum 100 entries
- **user_id** – optional list of str with maximum 100 entries
- **user_login** – optional list str with maximum 100 entries

Return type dict

get_top_games (*after: Optional[str] = None, before: Optional[str] = None, first: int = 20*) → dict

Requires no authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-top-games>

Parameters

- **after** – optional str
- **before** – optional str
- **first** – optional int in range 1 to 100

Return type dict

get_user_active_extensions (*user_id: Optional[str] = None*) → dict

Requires no authentication.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-user-active-extensions>

Parameters **user_id** – optional str

Return type dict

get_user_auth_token () → Optional[str]

Returns the current user auth token, None if no user Authentication is set

Returns current user auth token

Return type str, None

get_user_extensions () → dict

Requires User authentication with scope `USER_READ_BROADCAST`

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-user-extensions> :rtype: dict

get_users (*user_ids: Optional[List[str]] = None, logins: Optional[List[str]] = None*) → dict

Requires no authentication.

You have to either provide `user_ids` or `logins` or both. The maximum combined entries should not exceed 100.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-users>

Parameters

- **user_ids** – optional list of str
- **logins** – optional list of str

Return type dict

get_users_follows (*after: Optional[str] = None, first: int = 20, from_id: Optional[str] = None, to_id: Optional[str] = None*) → dict

Requires no authentication.

You have to use at least one of the following fields: `from_id`, `to_id` For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-users-follows>

Parameters

- **after** – optional str
- **first** – optional int between 1 and 100
- **from_id** – optional str
- **to_id** – optional str

Return type dict

get_videos (*ids: Optional[List[str]] = None, user_id: Optional[str] = None, game_id: Optional[str] = None, after: Optional[str] = None, before: Optional[str] = None, first: int = 20, language: Optional[str] = None, period: twitchAPI.types.TimePeriod = <TimePeriod.ALL: 'all'>, sort: twitchAPI.types.SortMethod = <SortMethod.TIME: 'time'>, video_type: twitchAPI.types.VideoType = <VideoType.ALL: 'all'>*) → dict

Requires no authentication.

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-videos>

Parameters

- **ids** – optional list of str
- **user_id** – optional str
- **game_id** – optional str
- **after** – optional str
- **before** – optional str
- **first** – optional int in range 1 to 100
- **language** – optional str
- **period** – optional *TimePeriod*

- **sort** – optional *SortMethod*
- **video_type** – optional *VideoType*

Return type dict

get_webhook_subscriptions (*first: Optional[str] = None, after: Optional[str] = None*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#get-webhook-subscriptions>

Parameters

- **first** – optional str
- **after** – optional str

Return type dict

modify_channel_information (*broadcaster_id: str, game_id: Optional[str] = None, broadcaster_language: Optional[str] = None, title: Optional[str] = None*) → bool

Requires User authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#modify-channel-information>

Parameters

- **broadcaster_id** – str
- **game_id** – optional str
- **broadcaster_language** – optional str
- **title** – optional str

Return type bool

redeem_code (*code: List[str], user_id: int*) → dict

Requires App authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#redeem-code>

Parameters

- **code** – list of str, maximum of 20 entries
- **user_id** – int

Return type dict

replace_stream_tags (*broadcaster_id: str, tag_ids: List[str]*) → dict

Requires User authentication with scope USER_EDIT_BROADCAST

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#replace-stream-tags>

Parameters

- **broadcaster_id** – str, id of streamer
- **tag_ids** – list of str, tag ids to be set

Returns empty dict

Return type dict

search_categories (*query: str, first: Optional[int] = None, after: Optional[str] = None*) → dict
Requires App or User authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#search-categories>

Parameters

- **query** – str, does NOT need to be URI encoded
- **first** – optional int
- **after** – optional str

Return type dict

search_channels (*query: str, first: Optional[int] = None, after: Optional[str] = None, live_only: Optional[bool] = False*) → dict
Requires App or User authentication

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#search-channels>

Parameters

- **query** – str, does NOT need to be URI encoded
- **first** – optional int
- **after** – optional str
- **live_only** – optional bool, default false

Return type dict

set_user_authentication (*token: str, scope: List[twitchAPI.types.AuthScope], refresh_token: Optional[str] = None*) → None

Set a user token to be used.

Parameters

- **token** – the generated user token
- **scope** – List of *AuthScope* that the given user token has
- **refresh_token** – str, has to be provided if `auto_refresh_user_auth` is True

Returns None

Raises ValueError

start_commercial (*broadcaster_id: str, length: int*) → dict
Requires User authentication with `AuthScope.CHANNEL_EDIT_COMMERCIAL`

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#start-commercial>

Parameters

- **broadcaster_id** – str
- **length** – int, one of these: [30, 60, 90, 120, 150, 180]

Return type dict

update_user (*description: str*) → dict
Requires User authentication with scope `USER_EDIT`

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#update-user>

Parameters **description** – str

Return type dict

update_user_extensions () → dict

“Requires User authentication with scope `USER_EDIT_BROADCAST`

For detailed documentation, see here: <https://dev.twitch.tv/docs/api/reference#update-user-extensions>

Return type dict

2.2 twitchAPI.webhook

class twitchAPI.webhook.**TwitchWebHook** (*callback_url: str, api_client_id: str, port: int*)

Webhook integration for the Twitch Helix API.

Parameters

- **callback_url** – The full URL of the webhook.
- **api_client_id** – The id of your API client
- **port** – the port on which this webhook should run

Variables

- **secret** – A random secret string. Set this for added security.
- **callback_url** – The full URL of the webhook.

authenticate (*auth_token: str*) → None

Set authentication for the Webhook. Can be either a app or user token.

Parameters **auth_token** – str

Return type None

Raises Exception

renew_subscription (*uuid: uuid.UUID*) → bool

Renew existing topic subscription

Parameters **uuid** – UUID of the subscription to renew

Return type bool

Returns True if renewal worked. Note that you still need to wait for the handshake to make sure its renewed.

start ()

Starts the Webhook

Return type None

stop ()

Stops the Webhook

Please make sure to unsubscribe from all subscriptions!

Return type None

subscribe_channel_ban_change_events (*broadcaster_id: str, user_id: Optional[str], callback_func: Callable[[uuid.UUID, dict], None]*) → Tuple[bool, uuid.UUID]

Subscribe to Channel Ban Change Events

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-channel-ban-change-events> for documentation

Parameters

- **broadcaster_id** – str
- **user_id** – str or None
- **callback_func** – function for callback

Return type bool, UUID

subscribe_extension_transaction_created (*extension_id: str, callback_func: Optional[Callable[[uuid.UUID, dict], None]]*) → Tuple[bool, uuid.UUID]

Subscribe to Extension transaction topic

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-extension-transaction-created> for documentation

Parameters

- **extension_id** – str
- **callback_func** – function for callback

Return type bool, UUID

subscribe_hype_train_events (*broadcaster_id: str, callback_func: Callable[[uuid.UUID, dict], None]*) → Tuple[bool, uuid.UUID]

Subscribe to Hype Train Events

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-hype-train-event> for documentation

Parameters

- **broadcaster_id** – str
- **callback_func** – function for callback

Return type bool, UUID**subscribe_least_seconds = 864000**

The duration in seconds for how long you want to subscribe to webhooks.

subscribe_moderator_change_events (*broadcaster_id: str, user_id: Optional[str], callback_func: Callable[[uuid.UUID, dict], None]*) → Tuple[bool, uuid.UUID]

Subscribe to Moderator Change Events topic

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-moderator-change-events> for documentation

Parameters

- **broadcaster_id** – str
- **user_id** – str or None
- **callback_func** – function for callback

Return type bool, UUID

subscribe_stream_changed (*user_id: str, callback_func: Optional[Callable[[uuid.UUID, dict], None]]*) → Tuple[bool, uuid.UUID]

Subscribe to stream changed topic

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-stream-changed> for documentation

Parameters

- **user_id** – str

- **callback_func** – function for callback

Return type bool, UUID

subscribe_subscription_events (*broadcaster_id: str, callback_func: Callable[[uuid.UUID, dict], None], user_id: Optional[str] = None, gifter_id: Optional[str] = None, gifter_name: Optional[str] = None*) → Tuple[bool, uuid.UUID]

Subscribe to Subscription Events Topic

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-subscription-events> for documentation

Parameters

- **broadcaster_id** – str
- **callback_func** – function for callback
- **user_id** – optional str
- **gifter_id** – optional str
- **gifter_name** – optional str

Return type bool, UUID

subscribe_user_changed (*user_id: str, callback_func: Optional[Callable[[uuid.UUID, dict], None]]*) → Tuple[bool, uuid.UUID]

Subscribe to subscription event topic

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-user-changed> for documentation

Parameters

- **user_id** – str
- **callback_func** – function for callback

Return type bool, UUID

subscribe_user_follow (*from_id: Optional[str], to_id: Optional[str], callback_func: Optional[Callable[[uuid.UUID, dict], None]]*) → Tuple[bool, uuid.UUID]

Subscribe to user follow topic.

Set only from_id if you want to know if User with that id follows someone.

Set only to_id if you want to know if someone follows User with that id.

Set both if you only want to know if from_id follows to_id.

See <https://dev.twitch.tv/docs/api/webhooks-reference#topic-user-follows> for documentation

Parameters

- **from_id** – str or None
- **to_id** – str or None
- **callback_func** – function for callback

Return type bool, UUID

unsubscribe_all (*twitch: twitchAPI.twitch.Twitch*) → None

Unsubscribe from all active Webhooks

Parameters **twitch** – App authorized instance of twitchAPI.twitch.Twitch

Return type None

wait_for_subscription_confirm = True

Set this to false if you dont want to wait for a subscription confirm.

wait_for_subscription_confirm_timeout = 30

Max time in seconds to wait for a subscription confirmation. Only used if wait_for_subscription_confirm is set to True

2.3 twitchAPI.oauth

User OAuth Authenticator and helper functions

class twitchAPI.oauth.UserAuthenticator (*twitch: twitchAPI.twitch.Twitch, scopes: List[twitchAPI.types.AuthScope], force_verify: bool = False*)

Simple to use client for the Twitch User authentication flow.

Parameters

- **twitch** (*twitchAPI.twitch.Twitch*) – A twitch instance
- **scopes** (*[twitchAPI.types.AuthScope]*) – List of the desired Auth scopes
- **force_verify** (*bool*) – If this is true, the user will always be prompted for authorization by twitch, default False

authenticate (*callback_func=None*)

Start the user authentication flow

If callback_func is not set, authenticate will wait till the authentication process finnishd and then return the access_token and the refresh_token

Parameters **callback_func** – Function to call once the authentication finnishd.

Returns None if callback_func is set, otherwise access_token and refresh_token

Return type None or (str, str)

stop()

Manually stop the flow

Return type None

twitchAPI.oauth.refresh_access_token (*refresh_token: str, app_id: str, app_secret: str*)

Simple helper function for refreshing a user access token.

Parameters

- **refresh_token** (*str*) – the current refresh_token
- **app_id** (*str*) – the id of your app
- **app_secret** (*str*) – the secret key of your app

Returns access_token, refresh_token

Return type (str, str)

2.4 twitchAPI.types

class twitchAPI.types.AnalyticsReportType

Enum of all Analytics report types

Variables

- **V1** –
- **V2** –

class twitchAPI.types.AuthScope

Enum of Authentication scopes

Variables

- **ANALYTICS_READ_EXTENSION** –
- **ANALYTICS_READ_GAMES** –
- **BITS_READ** –
- **CHANNEL_READ_SUBSCRIPTIONS** –
- **CHANNEL_READ_STREAM_KEY** –
- **CHANNEL_EDIT_COMMERCIAL** –
- **CHANNEL_READ_HYPE_TRAIN** –
- **CLIPS_EDIT** –
- **USER_EDIT** –
- **USER_EDIT_BROADCAST** –
- **USER_READ_BROADCAST** –
- **USER_READ_EMAIL** –
- **USER_EDIT_FOLLOWS** –
- **CHANNEL_MODERATE** –
- **CHAT_EDIT** –
- **CHAT_READ** –
- **WHISPERS_READ** –
- **WHISPERS_EDIT** –
- **MODERATION_READ** –

class twitchAPI.types.AuthType

Type of authentication required. Only internal use

Variables

- **NONE** – No auth required
- **USER** – User auth required
- **APP** – app auth required

class twitchAPI.types.CodeStatus

Enum Code Status, see <https://dev.twitch.tv/docs/api/reference#get-code-status> for more documentation

Variables

- **SUCCESSFULLY_REDEEMED** –
- **ALREADY_CLAIMED** –
- **EXPIRED** –

- **USER_NOT_ELIGIBLE** –
- **NOT_FOUND** –
- **INACTIVE** –
- **UNUSED** –
- **INCORRECT_FORMAT** –
- **INTERNAL_ERROR** –
- **UNKNOWN_VALUE** –

class twitchAPI.types.**HypeTrainContributionMethod**
Enum of valid Hype Train contribution types

Variables

- **BITS** –
- **SUBS** –
- **UNKNOWN** –

exception twitchAPI.types.**MissingScopeException**
authorization is missing scope

class twitchAPI.types.**ModerationEventType**
Enum of moderation event types

Variables

- **BAN** –
- **UNBAN** –
- **UNKNOWN** –

class twitchAPI.types.**SortMethod**
Enum of valid sort methods

Variables

- **TIME** –
- **TRENDING** –
- **VIEWS** –

class twitchAPI.types.**TimePeriod**
Enum of valid Time periods

Variables

- **ALL** –
- **DAY** –
- **WEEK** –
- **MONTH** –
- **YEAR** –

exception twitchAPI.types.**UnauthorizedException**
Not authorized to use this

class twitchAPI.types.VideoType

Enum of valid video types

Variables

- **ALL** –
- **UPLOAD** –
- **ARCHIVE** –
- **HIGHLIGHT** –
- **UNKNOWN** –

2.5 twitchAPI.helper

twitchAPI.helper.build_scope(scopes: list) → str

Builds a valid scope string from list

Parameters **scopes** – list of *AuthScope*

Return type str

twitchAPI.helper.build_url(url: str, params: dict, remove_none=False, split_lists=False) → str

Build a valid url string

Parameters

- **url** – base URL
- **params** – dictionary of URL parameter
- **remove_none** – optional bool, if set all params that have a None value get removed
- **split_lists** – optional bool, if set all params that are a list will be split over multiple url parameter with the same name

Returns URL

Return type str

twitchAPI.helper.fields_to_enum(data: Union[dict, list], fields: List[str], _enum: Type[enum.Enum], default: Optional[enum.Enum]) → Union[dict, list]

Iterates a dict or list and tries to replace every dict entry with key in fields with the correct Enum value

Parameters

- **data** – dict or list
- **fields** – list of keys to be replaced
- **_enum** – Type of Enum to be replaced
- **default** – The default value if _enum does not contain the field value

Return type dict or list

twitchAPI.helper.get_json(request: aiohttp.web_request.Request) → Union[list, dict, None]

Tries to retrieve the json object from the body

Parameters **request** – the request

Returns the object in the body or None

twitchAPI.helper.**get_uuid**()

Returns a random UUID

Return type UUID

twitchAPI.helper.**make_fields_datetime** (*data: Union[dict, list], fields: List[str]*)

Iterates over dict or list recursively to replace string fields with datetime

Parameters

- **data** – dict or list
- **fields** – list of keys to be replaced

Return type dict or list

t

- `twitchAPI.helper`, [22](#)
- `twitchAPI.oauth`, [19](#)
- `twitchAPI.twitch`, [5](#)
- `twitchAPI.types`, [19](#)
- `twitchAPI.webhook`, [16](#)

A

AnalyticsReportType (class in twitchAPI.types), 19

authenticate() (twitchAPI.oauth.UserAuthenticator method), 19

authenticate() (twitchAPI.webhook.TwitchWebHook method), 16

authenticate_app() (twitchAPI.twitch.Twitch method), 5

AuthScope (class in twitchAPI.types), 20

AuthType (class in twitchAPI.types), 20

auto_refresh_auth (twitchAPI.twitch.Twitch attribute), 5

B

build_scope() (in module twitchAPI.helper), 22

build_url() (in module twitchAPI.helper), 22

C

check_automod_status() (twitchAPI.twitch.Twitch method), 5

CodeStatus (class in twitchAPI.types), 20

create_clip() (twitchAPI.twitch.Twitch method), 6

create_entitlement_grants_upload_url() (twitchAPI.twitch.Twitch method), 6

create_stream_marker() (twitchAPI.twitch.Twitch method), 6

create_user_follows() (twitchAPI.twitch.Twitch method), 6

D

delete_user_follows() (twitchAPI.twitch.Twitch method), 6

F

fields_to_enum() (in module twitchAPI.helper), 22

G

get_all_stream_tags() (twitchAPI.twitch.Twitch method), 7

get_app_token() (twitchAPI.twitch.Twitch method), 7

get_banned_events() (twitchAPI.twitch.Twitch method), 7

get_banned_users() (twitchAPI.twitch.Twitch method), 7

get_bits_leaderboard() (twitchAPI.twitch.Twitch method), 8

get_broadcaster_subscriptions() (twitchAPI.twitch.Twitch method), 8

get_channel_information() (twitchAPI.twitch.Twitch method), 8

get_cheermotes() (twitchAPI.twitch.Twitch method), 8

get_clips() (twitchAPI.twitch.Twitch method), 8

get_code_status() (twitchAPI.twitch.Twitch method), 9

get_extension_analytics() (twitchAPI.twitch.Twitch method), 9

get_extension_transactions() (twitchAPI.twitch.Twitch method), 9

get_game_analytics() (twitchAPI.twitch.Twitch method), 10

get_games() (twitchAPI.twitch.Twitch method), 10

get_hype_train_events() (twitchAPI.twitch.Twitch method), 10

get_json() (in module twitchAPI.helper), 22

get_moderator_events() (twitchAPI.twitch.Twitch method), 10

get_moderators() (twitchAPI.twitch.Twitch method), 11

get_stream_key() (twitchAPI.twitch.Twitch method), 11

get_stream_markers() (twitchAPI.twitch.Twitch method), 11

get_stream_tags() (twitchAPI.twitch.Twitch method), 11

get_streams() (twitchAPI.twitch.Twitch method), 11

get_streams_metadata() (twitchAPI.twitch.Twitch method), 12

get_top_games() (*twitchAPI.twitch.Twitch method*), 12
 get_user_active_extensions() (*twitchAPI.twitch.Twitch method*), 12
 get_user_auth_token() (*twitchAPI.twitch.Twitch method*), 12
 get_user_extensions() (*twitchAPI.twitch.Twitch method*), 12
 get_users() (*twitchAPI.twitch.Twitch method*), 13
 get_users_follows() (*twitchAPI.twitch.Twitch method*), 13
 get_uuid() (*in module twitchAPI.helper*), 22
 get_videos() (*twitchAPI.twitch.Twitch method*), 13
 get_webhook_subscriptions() (*twitchAPI.twitch.Twitch method*), 14

H

HypeTrainContributionMethod (*class in twitchAPI.types*), 21

M

make_fields_datetime() (*in module twitchAPI.helper*), 23
 MissingScopeException, 21
 ModerationEventType (*class in twitchAPI.types*), 21
 modify_channel_information() (*twitchAPI.twitch.Twitch method*), 14

R

redeem_code() (*twitchAPI.twitch.Twitch method*), 14
 refresh_access_token() (*in module twitchAPI.oauth*), 19
 renew_subscription() (*twitchAPI.webhook.TwitchWebHook method*), 16
 replace_stream_tags() (*twitchAPI.twitch.Twitch method*), 14

S

search_categories() (*twitchAPI.twitch.Twitch method*), 14
 search_channels() (*twitchAPI.twitch.Twitch method*), 15
 set_user_authentication() (*twitchAPI.twitch.Twitch method*), 15
 SortMethod (*class in twitchAPI.types*), 21
 start() (*twitchAPI.webhook.TwitchWebHook method*), 16
 start_commercial() (*twitchAPI.twitch.Twitch method*), 15
 stop() (*twitchAPI.oauth.UserAuthenticator method*), 19

stop() (*twitchAPI.webhook.TwitchWebHook method*), 16
 subscribe_channel_ban_change_events() (*twitchAPI.webhook.TwitchWebHook method*), 16
 subscribe_extension_transaction_created() (*twitchAPI.webhook.TwitchWebHook method*), 17
 subscribe_hype_train_events() (*twitchAPI.webhook.TwitchWebHook method*), 17
 subscribe_least_seconds (*twitchAPI.webhook.TwitchWebHook attribute*), 17
 subscribe_moderator_change_events() (*twitchAPI.webhook.TwitchWebHook method*), 17
 subscribe_stream_changed() (*twitchAPI.webhook.TwitchWebHook method*), 17
 subscribe_subscription_events() (*twitchAPI.webhook.TwitchWebHook method*), 18
 subscribe_user_changed() (*twitchAPI.webhook.TwitchWebHook method*), 18
 subscribe_user_follow() (*twitchAPI.webhook.TwitchWebHook method*), 18

T

TimePeriod (*class in twitchAPI.types*), 21
 Twitch (*class in twitchAPI.twitch*), 5
 twitchAPI.helper (*module*), 22
 twitchAPI.oauth (*module*), 19
 twitchAPI.twitch (*module*), 5
 twitchAPI.types (*module*), 19
 twitchAPI.webhook (*module*), 16
 TwitchWebHook (*class in twitchAPI.webhook*), 16

U

UnauthorizedException, 21
 unsubscribe_all() (*twitchAPI.webhook.TwitchWebHook method*), 18
 update_user() (*twitchAPI.twitch.Twitch method*), 15
 update_user_extensions() (*twitchAPI.twitch.Twitch method*), 15
 UserAuthenticator (*class in twitchAPI.oauth*), 19

V

VideoType (*class in twitchAPI.types*), 21

W

`wait_for_subscription_confirm`
(*twitchAPI.webhook.TwitchWebHook* *at-tribute*), [18](#)

`wait_for_subscription_confirm_timeout`
(*twitchAPI.webhook.TwitchWebHook* *at-tribute*), [19](#)