
Hootsweet

Release 0.6.5

Aug 22, 2020

Contents:

1	HootSweet Client	3
2	Indices and tables	7
	Python Module Index	9
	Index	11

CHAPTER 1

HootSweet Client

This module provides a client, HootSweet, to interact with the Hootsuite REST 1.0 API.

```
class hootsweet.api.HootSweet (client_id: str, client_secret: str, token: Dict[KT, VT] = None,  
                             redirect_uri: str = None, scope: str = 'offline', refresh_cb=None,  
                             **kwargs)
```

A client for interacting with the Hootsuite REST API.

Parameters

- **client_id** (*str*) – A Hootsuite client id.
- **client_secret** (*str*) – A Hootsuite client secret.
- **token** (*Dict*) – A token dictionary containing the *access_token*, *refresh_token*, *expires_on* and *expires_in* keys.
- **redirect_uri** (*str*) – The callback uri registered with Hootsuite.
- **scope** (*str*) – The OAuth2 scope.
- **refresh_cb** (*callable*) – A function to be called when a token is refreshed.

```
authorization_url (state: str = None, **kwargs) → Tuple[str, str]
```

Get a Hootsuite authorization url.

Parameters **state** (*str*) – An opaque value used by the client to maintain state between the request and callback.

```
fetch_token (code: str) → Dict[str, Any]
```

Fetch a Hootsuite OAuth2 token.

Parameters **code** (*str*) – The authorization code obtained from Hootsuite.

```
refresh_token () → Dict[str, Any]
```

Refresh the OAuth2 token and call token updater.

```
get_me () → Dict[KT, VT]
```

Retrieve the currently authenticated member.

get_me_organizations () → Dict[KT, VT]

Retrieve the organizations that the authenticated member is in.

get_me_social_profiles () → Dict[KT, VT]

Retrieve the social media profiles that the authenticated user has basic usage permissions on.

get_social_profiles () → Dict[KT, VT]

Retrieve the social profiles that the authenticated user has access to.

get_social_profile (*profile_id: int*) → Dict[KT, VT]

Retrieve a social profile.

Parameters *profile_id* (*int*) – The social profile id.

get_social_profile_teams (*profile_id: int*) → List[T]

Retrieve a list of team IDs with access to a social profile.

Parameters *profile_id* (*int*) – The social profile id.

get_member (*member_id: str*) → Dict[str, Any]

Retrieve a member.

Parameters *member_id* (*str*) – The Hootsuite member id.

create_member (*full_name: str, email: str, organization_ids: List[int], company_name: str = None, bio: str = None, timezone: str = 'Europe/London', language: str = 'en'*) → Dict[str, Any]

Create a member in a Hootsuite organization.

Parameters

- **full_name** (*str*) – The member's name.
- **email** (*str*) – The member's email.
- **organization_ids** (*List[int]*) – The organizations the member should be added to.
- **company_name** (*str*) – The member's company name.
- **bio** (*str*) – The member's bio.
- **timezone** (*str*) – The member's time zone. Defaults to "Europe/London"
- **language** (*str*) – The member's language. Defaults to "en"

get_member_organizations (*member_id: str*) → List[Dict[str, Any]]

Retrieve the organizations that the member is in.

Parameters *member_id* (*str*) – A Hootsuite member id.

schedule_message (*text: str, social_profile_ids: List[str], send_time: datetime.datetime, **kwargs*)

Schedule a message to send on one or more social profiles.

Parameters

- **text** (*str*) – The text of the message.
- **social_profile_ids** (*List[str]*) – A list of ids of social profiles that will publish the message.
- **send_time** (*datetime*) – Time to send the message in UTC time.

get_outbound_messages (*start_time: datetime.datetime, end_time: datetime.datetime, state: hootsweet.constants.MessageState = None, social_profile_ids: List[int] = None, limit: int = 50, include_unscheduled_review_messages: bool = None, **kwargs*) → List[Dict[str, Any]]

Retrieve a list of outbound messages.

Parameters

- **start_time** (*datetime*) – The start date range of the messages returned.
- **end_time** (*datetime*) – The end date range of the messages returned.
- **state** (*MessageState*) – The state of the messages returned.
- **social_profile_ids** (*List[int]*) – The ids of social profiles of the messages returned.
- **limit** (*int*) – Maximum number of messages to be returned. Defaults to 50.
- **include_unscheduled_review_messages** (*bool*) – Flag to retrieve unscheduled (Send Now) review messages on top of scheduled ones retrieved from time range query.

get_message (*message_id: str*) → Dict[str, Any]

Retrieve a message.

Parameters **message_id** (*str*) – The Hootsuite message id.

delete_message (*message_id: str*) → Dict[str, Any]

Delete a message.

Parameters **message_id** (*str*) – The Hootsuite message id.

approve_message (*message_id: str, sequence_number: int, reviewer_type: hootsweet.constants.Reviewer*)

Approve a message.

Parameters

- **message_id** (*str*) – A Hootsuite message id.
- **sequence_number** (*int*) – The sequence number of the message.
- **reviewer_type** (*Reviewer*) – The actor that will be approving the message.

reject_message (*message_id: str, reason: str, sequence: int, reviewer_type: hootsweet.constants.Reviewer = None, **kwargs*)

Reject a message.

Parameters

- **message_id** (*str*) – The Hootsuite message id.
- **reason** (*str*) – The rejection reason.
- **sequence** (*int*) – The sequence number of the message.
- **reviewer_type** (*Reviewer*) – The actor that will be rejecting the message.

get_message_review_history (*message_id: str*) → Dict[KT, VT]

Get a messages prescreening review history.

Parameters **message_id** (*str*) – The Hootsuite message id.

create_media_upload_url (*size_bytes: int, mime_type: str*)

Creates an Amazon S3 upload URL that can be used to transfer media to Hootsuite.

Parameters

- **size_bytes** (*int*) – Size in bytes of the media file to be uploaded.
- **mime_type** (*str*) – MIME type of the media to be uploaded. Supported
- **types** are **video/mp4**, **image/gif**, **image/jpeg**, **image/jpg**, **image/png**. (*media*) –

get_media_upload_status (*media_id*)

Retrieves the status of a media upload to Hootsuite.

Parameters **media_id** (*str*) – The Media ID to retrieve.

CHAPTER 2

Indices and tables

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

h

`hootsweet.api`, [1](#)

A

`approve_message()` (*hootsweet.api.HootSweet method*), 5

`authorization_url()` (*hootsweet.api.HootSweet method*), 3

C

`create_media_upload_url()` (*hootsweet.api.HootSweet method*), 5

`create_member()` (*hootsweet.api.HootSweet method*), 4

D

`delete_message()` (*hootsweet.api.HootSweet method*), 5

F

`fetch_token()` (*hootsweet.api.HootSweet method*), 3

G

`get_me()` (*hootsweet.api.HootSweet method*), 3

`get_me_organizations()` (*hootsweet.api.HootSweet method*), 3

`get_me_social_profiles()` (*hootsweet.api.HootSweet method*), 4

`get_media_upload_status()` (*hootsweet.api.HootSweet method*), 6

`get_member()` (*hootsweet.api.HootSweet method*), 4

`get_member_organizations()` (*hootsweet.api.HootSweet method*), 4

`get_message()` (*hootsweet.api.HootSweet method*), 5

`get_message_review_history()` (*hootsweet.api.HootSweet method*), 5

`get_outbound_messages()` (*hootsweet.api.HootSweet method*), 4

`get_social_profile()` (*hootsweet.api.HootSweet method*), 4

`get_social_profile_teams()` (*hootsweet.api.HootSweet method*), 4

`get_social_profiles()` (*hootsweet.api.HootSweet method*), 4

H

`HootSweet` (*class in hootsweet.api*), 3

`hootsweet.api` (*module*), 1

R

`refresh_token()` (*hootsweet.api.HootSweet method*), 3

`reject_message()` (*hootsweet.api.HootSweet method*), 5

S

`schedule_message()` (*hootsweet.api.HootSweet method*), 4