tildee

Oct 04, 2019

Contents

This is Tildee, a Python 3 library for interacting with the https://tildes.net API. Note that this API is not stable and not actually intended for external use, so this could break at any moment.

Source code and issue tracker on Dingenskirchen Systems Git. Available on PyPI.

CHAPTER 1

Capabilities

Currently Tildee can parse posts and their comments, create comments, topics and messages, parse new messages and notifications, edit comment and topic contents, edit topic metadata, delete, remove and lock topics and comments.

CHAPTER 2

Missing features

See the relevant label on the issue tracker.

CHAPTER 3

Development

To install dependencies, run `poetry install`. You can run a python shell in the environment using `poetry run python` (I'd recommend using `ipython`, too). Format your code before committing by running `black .`.

CHAPTER 4

License

The project is licensed under the MIT license.

4.1 Client Usage

4.1.1 General

All methods of the `TildesClient` raise if any exception occurs, including unsuccessful HTTP responses from the site.

4.1.2 Logging in

You can connect to Tildes and log in by creating a new `TildesClient` instance and passing the login data to the constructor. If you're not connecting to tildes.net, override the `base_url` argument. If you're connecting to a local development instance, which uses self-signed SSL certificates, also pass `verify_ssl = False` to avoid errors. The automatic ratelimiting is set to 0.5 seconds between each request, but you can adjust it using the `ratelimit` parameter. Keep in mind that some actions, like posting topics or comments, are additionally ratelimited, see the Tildes Source Code.

class tildee.**TildesClient**(*username: str*, *password: str*, *totp_code: Optional[str] = None*, *base_url: str = 'https://tildes.net'*, *verify_ssl: bool = True*, *ratelimit: int = 500*, *user_agent: str = ''*)

Initializes client and logs in.

Parameters

- **username** (*str*) – The username to log in with.
- **password** (*str*) – The password to log in with.
- **totp_code** (*Optional[str]*) – The 2FA code to use.
- **base_url** (*str*) – The site to log in to.

- **verify_ssl** (*bool*) – Whether to check SSL certificate validity.
- **ratelimit** (*int*) – The default cooldown between each request, in milliseconds.
- **user_agent** (*str*) – What to include in the UserAgent string.

4.1.3 Posting topics and comments

You can use the `create_topic()` and `create_comment()` methods for these tasks. They both return the new item's id36, which you can use for further operations or just ... discard.

`tildee.TildesClient.`**`create_topic`**(*self, group: str, title: str, tags: Union[str, List[str]], **kwargs*) → Union[str, List[str]]

Post a topic into a group, returns new topic's id36. Either a link or markdown must be passed. If this method returns a list instead of a string, there have been previous topics with this link and posting was therefore canceled. Pass the `confirm_repost=True` to force posting anyway.

Parameters

- **group** (*str*) – The group to post in, without a ~ in front.
- **title** (*str*) – The topic's title.
- **tags** (*str or List[str]*) – Comma separated string or list of tags.
- **markdown** (*str*) – The topic's content as markdown.
- **link** (*str*) – The topic's link.
- **confirm_repost** (*bool*) – Whether to submit the topic even if one with the same link was posted before.

Return type Union[str, List[str]]

Returns New topic's id36.

`tildee.TildesClient.`**`create_comment`**(*self, parent_id36: str, markdown: str, top_level: bool = True*) → str

Post a comment, returns new comment's id36.

Parameters

- **markdown** (*str*) – The comment's content as markdown.
- **parent_id36** (*str*) – The parent entity's id36. Can be a topic or comment.
- **top_level** (*bool*) – Set this to False if the comment's a reply to another comment.

Return type str

Returns The new comment's id36.

4.1.4 Fetching topics and comments

If you want to use data from comments or topics, you can use Tildee to fetch them and extract their data from the raw HTML the site returns. For more information and examples on these representations see the *Model Classes* page.

`tildee.TildesClient.`**`fetch_topic`**(*self, topic_id36: str*) → tildee.models.TildesTopic

Fetches, parses and returns a topic as an object for further processing.

Parameters **topic_id36** (*str*) – The id36 of the topic to fetch.

Return type *TildesTopic*

Returns The requested topic.

tildee.TildesClient.**fetch_comment**(*self*, *comment_id36: str*) → tildee.models.TildesComment
Fetches, parses and returns a single comment as an object for further processing.

This endpoint doesn't include a comments' children or the applied labels.

Parameters **comment_id36** (*str*) – The id36 of the comment to fetch.

Return type *TildesComment*

Returns The requested comment.

4.1.5 Fetching topic listings

If you either want to fetch topic listings for tags and/or groups or search for a phrase, use the `fetch_filtered_topic_listing` or the `fetch_search_topic_listing` method. Note that they only return the limited data the topic listing page offers.

tildee.TildesClient.**fetch_filtered_topic_listing**(*self*, *group: str = ''*, *tag: str = ''*, ***kwargs*)
Fetches a filtered list of topics. Automatically adds `per_page=100` to the query string.

Parameters

- **group** (*str*) – The group to filter for. Leave empty to search all subscribed groups.
- **tag** (*str*) – The tag to filter for, Tildes currently only supports filtering for one tag.

Return type List[*TildesPartialTopic*]

Returns Up to 100 topics matching the filters.

tildee.TildesClient.**fetch_search_topic_listing**(*self*, *query: str = ''*, *group: str = ''*, ***kwargs*)
Fetches a search result's list of topics. Automatically adds `per_page=100` to the query string.

Parameters

- **query** (*str*) – The string to search for.
- **group** (*str*) – The group to search in, don't pass the argument to search in all groups.

Return type List[*TildesPartialTopic*]

Returns Up to 100 topics matching the query string.

4.1.6 Interacting with groups

To see what groups your account is subscribed to and which others are available, use `fetch_groups`, it returns a list of `TildesGroup` instances. For details on these see the *Model Classes* page. To (un)subscribe from a group, use the `set_group_subscription` method.

tildee.TildesClient.**fetch_groups**(*self*)
Fetches groups as a list of objects.

Return type List[*TildesGroup*]

Returns All groups on the site as a list.

tildee.TildesClient.**set_group_subscription**(*self*, *group: str*, *subscribe: bool = True*)
(Un)subscribes to/from a group.

Parameters

- **group** (*str*) – The group to target.
- **subscribe** (*bool*) – Whether to add (True) or remove (False) your subscription.

4.1.7 Interacting with topics

You can use Tildee to edit a topic's metadata, e.g. its tags or title (assuming your account has the permissions for this), edit or delete your account's own topics, and remove or lock topics if your account has admin permissions.

tildee.TildesClient.**edit_topic**(*self*, *topic_id36: str*, ***kwargs*)
Interact with a topic in nearly any way possible; permission limits still apply, obviously.

Parameters

- **topic_id36** (*str*) – The id36 of the topic to act on.
- **tags** (*str or List[str]*) – Comma separated string or list of tags.
- **old_tags** – Optional (Comma separated string or list): Tags to pass to the overwrite protection, if not passed, ignores overwrite protection.
- **group** (*str*) – The new group for the topic, without ~ in front.
- **title** (*str*) – The new title for the topic.
- **link** (*str*) – The new link for the topic.
- **content** (*str*) – The new markdown for the topic. Account must be topic author.
- **vote** (*bool*) – Boolean, vote/unvote this topic.
- **bookmark** (*bool*) – Boolean, bookmark/unbookmark this topic.

tildee.TildesClient.**delete_topic**(*self*, *topic_id36: str*)
Delete a topic. Account must be topic author.

Parameters **topic_id36** (*str*) – The id36 of the topic to delete.

tildee.TildesClient.**moderate_topic**(*self*, *topic_id36: str*, ***kwargs*)
Moderate a topic, setting its locked/removed status. Account must be admin.

Parameters

- **topic_id36** (*str*) – The id36 of the topic to act on.
- **lock** (*bool*) – Boolean, lock/unlock comments.
- **remove** (*bool*) – Boolean, remove/unremove this topic.

4.1.8 Interacting with comments

You can use Tildee to interact with comments, i.e. editing and deleting your account's own, voting on and labeling other's, bookmarking, and — given admin permissions — removing them.

tildee.TildesClient.**edit_comment**(*self*, *comment_id36: str*, ***kwargs*)
Interact with a comment in nearly any way possible; permission limits still apply, obviously.

Parameters

- **comment_id36** (*str*) – The id36 of the comment to act on.
- **content** (*str*) – The new markdown for the comment. Account must be comment author.
- **vote** (*bool*) – Boolean, vote/unvote this comment.

- **bookmark** (*bool*) – Boolean, bookmark/unbookmark this comment.

tildee.TildesClient.**edit_comment_labels**(*self*, *comment_id36: str*, ***kwargs*)
Change which labels the account applies to a comment.

Parameters

- **comment_id36** (*str*) – The id36 of the comment to edit.
- **str] labelnames** (*Union[bool,*) – Keyword Arguments: For `offtopic`, `noise`, `joke`, pass a Boolean to set/unset the label. For `exemplary` and `malice`, pass a string to set the label and `False` to unset it.

tildee.TildesClient.**delete_comment**(*self*, *comment_id36: str*)
Delete a comment. Account must be comment author.

Parameters **comment_id36** (*str*) – The id36 of the comment to delete.

tildee.TildesClient.**moderate_comment**(*self*, *comment_id36: str*, ***kwargs*)
Moderate a comment, setting its removed status. Account must be admin.

Parameters

- **comment_id36** (*str*) – The id36 of the comment to act on.
- **remove** (*bool*) – Boolean, remove/unremove comment.

4.1.9 Notifications

Notifications are created by comments when they're in reply to one of your comments/topics or your account is @-mentioned in their text. You can fetch unread notifications as `TildesNotification` objects using `fetch_unread_notifications()`, for more on these see the *Model Classes* page. Use the `mark_notification_as_read()` method to mark a notification as read.

tildee.TildesClient.**fetch_unread_notifications**(*self*) → List[tildee.models.TildesNotification]
Fetches, parses and returns a list of unread notifications as objects for further processing.

Return type List[*TildesNotification*]

Returns The list of unread notifications.

tildee.TildesClient.**mark_notification_as_read**(*self*, *subject_id36: str*)
Marks a notification as read.

Parameters **subject_id36** (*str*) – The notification subject's id36 to mark.

4.1.10 Sending messages

You can send messages in existing conversations using `create_message()` or create conversations using the aptly named `create_conversation()`.

tildee.TildesClient.**create_message**(*self*, *convo_id36: str*, *markdown: str*)
Creates a message in an existing conversation.

Parameters

- **convo_id36** (*str*) – The target conversation's id36.
- **markdown** (*str*) – The message's content as markdown.

tildee.TildesClient.**create_conversation**(*self*, *username: str*, *subject: str*, *markdown: str*)
Creates a new conversation with a user.

Parameters

- **username** (*str*) – The username of the recipient.
- **subject** (*str*) – The conversation's subject.
- **markdown** (*str*) – The first message's content as markdown.

4.1.11 Fetching messages and conversations

To check for new messages, use the `fetch_unread_message_ids()` method which returns a list of message id36s. You can then use `fetch_conversation()` to fetch and process them individually. A conversation is represented by a `TildesConversation` object, which has a list of children `TildesMessage` objects. For details on these see the *Model Classes* page.

tildee.TildesClient.**fetch_unread_message_ids**(*self*) → List[str]
Fetches IDs of unread messages.

Return type List[str]

Returns The list of unread conversations' id36s.

tildee.TildesClient.**fetch_conversation**(*self*, *convo_id36: str*) → tildee.models.TildesConversation
Fetches, parses and returns a conversation as TildesConversation object for further processing.

Parameters **convo_id36** (*str*) – The target conversation's id36.

Return type *TildesConversation*

Returns The requested conversation.

4.1.12 Interacting with group wikis

You can fetch a group wiki page using `fetch_wiki_page`, and, if you have editing permissions, its underlying markdown code using `fetch_wiki_page_markdown`. Fetch a list of all wiki pages in a group using `fetch_wiki_page_list` and edit/create new pages using `edit_wiki_page` and `create_wiki_page`, respectively.

tildee.TildesClient.**fetch_wiki_page_list**(*self*, *group: str*)
Fetches the slugs of all wiki pages in a group.

Parameters **group** (*str*) – The group to target.

Return type List[str]

Returns The list of slugs of this group's wiki pages.

tildee.TildesClient.**fetch_wiki_page**(*self*, *group: str*, *slug: str*)
Fetches a single group wiki page.

Parameters

- **group** (*str*) – The group to target.
- **slug** (*str*) – The wiki site's slug.

Return type *TildesWikiPage*

Returns The wiki page.

tildee.TildesClient.**fetch_wiki_page_markdown**(*self*, *group: str*, *slug: str*)
Fetches a single group wiki page's markdown. This requires edit permissions.

Parameters

- **group** (*str*) – The group to target.
- **slug** (*str*) – The wiki page's slug.

Return type str

Returns The wiki page's content as markdown.

tildee.TildesClient.**edit_wiki_page**(*self*, *group: str*, *slug: str*, *markdown: str*, *commit: str*)
Updates a wiki page's markdown.

Parameters

- **group** (*str*) – The group to target.
- **slug** (*str*) – The wiki page's slug.
- **markdown** (*str*) – The new markdown for the page.
- **commit** (*str*) – The commit message/edit summary.

tildee.TildesClient.**create_wiki_page**(*self*, *group: str*, *title: str*, *markdown: str*)
Creates a wiki page with starting content.

Parameters

- **group** (*str*) – The group to create a page in.
- **title** (*str*) – The new wiki page's title.
- **markdown** (*str*) – The new wiki page's content.

4.2 Model Classes

These are all representations of "things" on Tildes, created from HTML from the site and provide easy access to the interesting parts of that HTML.

4.2.1 Topics

class tildee.models.**TildesTopic**(*text*)
Represents a single topic on Tildes, generated from the entire page.

Variables

- **tags** (*List[str]*) – List of tags this topic has.
- **group** (*str*) – The group this topic was posted in.
- **title** (*str*) – The title of this topic.
- **id36** (*str*) – The id36 of the topic.
- **status** (TildesAccessStatus) – Status of this topic. If DELETED or REMOVED, content_html, link, author, timestamp, comments, log, num_votes and num_comments are unavailable.
- **content_html** (*Optional[str]*) – The text of this topic as rendered by the site.

- **link** (*Optional[str]*) – The link of this topic.
- **author** (*str*) – The topic author's username.
- **timestamp** (*str*) – The topic's creation timestamp.
- **num_votes** (*int*) – The amount of votes this topic has received.
- **num_comments** (*int*) – The amount of comments on this topic.
- **log** (*List[TildesTopicLogEntry]*) – The associated topic log in chronological order.
- **is_locked** (*bool*) – Whether or not the topic is currently locked.
- **comments** (*List[TildesComment]*) – Top level comments in this topic.

class tildee.models.**TildesPartialTopic**(*text*)

Represents a topic on Tildes as fetched from the topic listings, generated from its surrounding `<article>` tag. Has only reduced information available.

Variables

- **id36** (*str*) – The topic's id36.
- **title** (*str*) – The topic's title.
- **group** (*Optional[str]*) – The topic's group, if provided in the listing.
- **author** (*str*) – The topic's author.
- **link** (*Optional[str]*) – The topic's link, if available.
- **content_html** (*Optional[str]*) – The text of this topic as rendered by the site, if available.
- **num_votes** (*int*) – The amount of votes this topic has received.
- **num_comments** (*int*) – The amount of comments on this topic.
- **tags** (*List[str]*) – The tags on this topic. Can be incomplete or missing if you have the "Show topic tags in listing pages" setting disabled.
- **timestamp** (*str*) – The topic's timestamp.

class tildee.models.**TildesTopicLogEntry**(*text*)

Represents a single entry from a topic's log, generated from its surrounding `<li>` tag.

Variables

- **user** (*str*) – The responsible curator's username.
- **timestamp** (*str*) – The entry's timestamp.
- **kind** (TildesTopicLogEntryKind) – The kind of log entry.
- **data** (*Optional[Dict]*) – The data associated with this log entry. See *below* for structure.

class tildee.models.**TildesTopicLogEntryKind**

Enum representing the possible kinds of topic log entry. Documentation includes structure for `TildesTopicLogEntry`'s data attribute.

LINK_EDIT = 4

Link edit, data contains old and new link.
data: `{"old": str, "new": str}`

LOCK = 6

Comments locked, no data.
data: `None`

MOVE = 5

Group move, data contains old and new group path.
data: `{"old": str, "new": str}`

REMOVE = 8

Topic removed, no data.
data: `None`

TAG_EDIT = 2

Tag edit, data contains added and removed tags.
data: `{"added": List[str], "removed": List[str]}`

TITLE_EDIT = 3

Title edit, data contains old and new title. If the program can't decide what is part of the titles and what isn't, `certain` will be set to `False`.
data: `{"old": str, "new": str, "certain": bool}`

UNKNOWN = 1

Default option if the entry is unrecognized, no data.
data: `None`

UNLOCK = 7

Comments unlocked, no data.
data: `None`

UNREMOVE = 9

Topic unremoved, no data.
data: `None`

class tildee.models.**TildesAccessStatus**

Enum representing the possible visibility statuses of comments or topics.

DELETED = 3
Comment/Topic deleted by its author, only reduced data available.

FULL = 1
Full content is available.

REMOVED = 2
Comment/Topic removed by site admin, only reduced data available.

4.2.2 Comments

class tildee.models.**TildesComment**(*text*)
Represents a single comment on Tildes, generated from its surrounding <article> tag.

Variables

- **id36** (*str*) – The id36 of this comment.
- **status** (TildesAccessStatus) – Status of this comment. If DELETED or REMOVED, no data beyond id36 are available.
- **content_html** (*str*) – This comment's content as rendered by the site.
- **applied_labels** (*List[str]*) – The labels the account has applied to this comment.
- **author** (*str*) – The comment author's username.
- **timestamp** (*str*) – The comment's creation timestamp.
- **num_votes** (*int*) – The amount of votes this comment has received.
- **children** (*List[TildesComment]*) – Top level replies to this comment.

class tildee.models.**TildesNotification**(*text*)
Represents a single notification on Tildes, generated from its surrounding <li> tag.

Variables

- **subject** (*str*) – The id36 of the comment that triggered the notification.
- **kind** (TildesNotificationKind) – The kind of notification.

class tildee.models.**TildesNotificationKind**
Enum representing the possible kinds of notification.

COMMENT_REPLY = 4

MENTION = 2

TOPIC_REPLY = 3

UNKNOWN = 1

4.2.3 Groups

class tildee.models.**TildesGroup**(*text*)
Data about a Tildes Group, generated from it's <tr> tag on the group listing page.

Variables

- **name** (*str*) – The name of the group.
- **desc** (*str*) – The group's description.
- **num_subscribers** (*int*) – How many users have subscribed to the group.
- **subscribed** (*bool*) – If the current user is subscribed to the group.

class tildee.models.**TildesWikiPage**(*text*)
Represents a single group wiki page, generated from its entire page.

Variables

- **title** (*str*) – The page's title.

- **slug** (*str*) – The page's slug, i.e. an URL-safe transformation of the title.
- **group** (*str*) – The group the page is in.
- **content_html** (*str*) – The content of the page as rendered by the site.

4.2.4 Messages

class tildee.models.**TildesConversation**(*text*)
Represents a conversation on Tildes, generated from the entire page.

Variables

- **title** (*str*) – The Subject of this conversation.
- **entries** (*List[TildesMessage]*) – The messages in this conversation in order.

class tildee.models.**TildesMessage**(*text*)
Represents a message in a conversation on Tildes, generated from its surrounding <article> tag.

Variables

- **author** (*str*) – The message author's username.
- **timestamp** (*str*) – The message's creation timestamp.
- **content_html** (*str*) – The message's content as rendered by the site.

Index

T

U