pygnusocial Documentation

Release 4.0.0

dtluna

Aug 05, 2017

Contents

pygnusocial uses Requests and Requests-OAuthlib libraries to provide an easy-to-use Python interface for building clients for GNU Social.

CHAPTER 1

Installation

pygnusocial can be installed with pip:

```
$  pip install gnusocial
```

CHAPTER 2

Contents:

API Reference

Accounts

accounts.**verify_credentials**(*server_url*, **args*, ***kwargs*)

Tests if supplied user credentials are valid.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/accounts.html#get--api-account-verify_credentials.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

accounts.**register**(*server_url*, **args*, ***kwargs*)

Register a new user.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/accounts.html#post--api-account-register.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

accounts.**update_profile**(*server_url*, **args*, ***kwargs*)

Updates the authenticating user's profile.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/accounts.html#post--api-account-update_profile.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

accounts.**update_profile_image**(*server_url*, **args*, ***kwargs*)
Updates the authenticating user's profile image.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/accounts.html#post--api-account-update_profile_image.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

ActivityStreams

activity_streams.**public**(*server_url*, **args*, ***kwargs*)
Returns the most recent notices, including repeats if they exist, from non-protected users.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-public_timeline.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

activity_streams.**home**(*server_url*, **args*, ***kwargs*)
Returns the most recent notices, including repeats if they exist, posted by the authenticating user and the users they follow.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-home_timeline.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

activity_streams.**friends**(*server_url*, **args*, ***kwargs*)
Home timeline for the specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-friends_timeline.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

activity_streams.**user**(*server_url*, **args*, ***kwargs*)
Returns the most recent notices posted by the authenticating user. It is also possible to request another user's timeline by using the `screen_name` or `user_id` parameter. The other users timeline will only be visible if they are not protected, or if the authenticating user's follow request was accepted by the protected user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-user_timeline.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

activity_streams.**mentions**(*server_url*, **args*, ***kwargs*)
Returns the most recent mentions (notices containing @username) for the authenticating user or specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-mentions.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

activity_streams.**replies**(*server_url*, **args*, ***kwargs*)
Returns the most recent mentions (notices containing @username) for the authenticating user or specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/activity_streams.html#get--api-statuses-replies.as

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *ActivityStream dict*

Blocking

blocks.**create**(*server_url*, **args*, ***kwargs*)
Blocks the specified user from following the authenticating user. In addition the blocked user will not show in the authenticating users mentions or timeline (unless repeated by another user). If a follow or friend relationship exists it is destroyed.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/blocks.html#post--api-blocks-create.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

blocks.**destroy**(*server_url*, **args*, ***kwargs*)
Un-blocks the specified user from following the authenticating user. If relationships existed before the block was instated, they will not be restored.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/blocks.html#post--api-blocks-destroy.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

Configuration

config.**config**(*server_url*, **args*, ***kwargs*)

Returns server configuration.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/config.html#get--api-statusnet-config.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *Config dict*

Data structures

OAuth dict

A dict to pass as the `oauth` argument to any function except ones in *oauth* module to use for authentication.

Dicts of this structure are returned by *oauth.request_token()* and *oauth.access_token()*.

```
{
    'resource_owner_key': '66b3c0ce86e231fb0b014dcba5c51c47',
    'resource_owner_secret': 'e261a7e968c9156972e7b1dc677686c4',
}
```

User dict

A dict describing a GNU Social user. Returned by functions in *accounts*, *blocks*, *users* modules and also *friendships.create()* and *friendships.destroy()*.

```
{
    'background_image': str, # URL to background image or False if none
    'backgroundcolor': str, # background color in hex or False if default
    'cover_photo': str, # URL to cover image or False if none
    'created_at': str, # the date of user registration
    'description': str, # user profile description or False if none
    'favourites_count': int, # the number of notices favorited by user
    'followers_count': int,
    'following': bool, # True if authenticating user is following the user
    'friends_count':int, # the number of users user is following
    'groups_count': int, # the number of group user is member of
    'id': int,
    'is_local': bool, # True if user is from server_url
    'is_sandboxed': bool,
    'is_silenced': bool,
    'linkcolor': str, # link color in hex or False if default
    'location': str, # user's location or False if none
    'name': str, # full name associated with the profile
```

```
    'notifications': bool, # True if authenticating user is getting
                           # notifications from user
    'ostatus_uri': str, # URL to user profile
    'profile_background_color':str, # same as backgroundcolor
    'profile_banner_url': str, # same as cover_photo
    'profile_image_url': str, # URL to 48x48 avatar image
    'profile_image_url_https': str. # same as profile_image_url, but with
                                    # HTTPS
    'profile_image_url_original': str, # URL to avatar image in original
                                       # resolution
    'profile_image_url_profile_size': str, # URL to 96x96 avatar image
    'profile_link_color': str, # same as linkcolor
    'protected': bool,
    'rights': # a dict of what user can do
    {
        'delete_others_notice': bool,
        'delete_user': bool,
        'sandbox': bool,
        'silence': bool
    },
    'screen_name': str, # user's handle
    'status': dict, # user's latest status
    'statuses_count': int,
    'statusnet_blocking': bool,
    'statusnet_profile_url': str,
    'time_zone': str,
    'url': str, # URL associated with the profile or False if none
    'utc_offset': str
}
```

Status dict

```
{
    'attachments':
    [
        {
            'height': str,
            'id': str,
            'large_thumb_url': str,
            'mimetype': str,
            'oembed': bool,
            'size': str, # size of attacment in kilobytes
            'thumb_url': str,
            'url': str,
            'version': str,
            'width': str
        }
    ],
    'attentions':
    [
        {
            'fullname': str,
            'id': int,
            'ostatus_uri': str,
            'profileurl': str,
            'screen_name': str
```

```
        }
    ],
    'created_at': str,
    'external_url': str,
    'fave_num': int, # number of users favorited the notice
    'favorited': bool, # True if favorited by authenticated user
    'geo': str,
    'id': int,
    'in_reply_to_ostatus_uri': str,
    'in_reply_to_profileurl': str,
    'in_reply_to_screen_name': str,
    'in_reply_to_status_id': str,
    'in_reply_to_user_id': int,
    'is_local': bool,
    'is_post_verb': bool,
    'repeat_num': int,
    'repeated': bool, # True if repeated by authenticated user
    'repeated_id': int,
    'retweeted_status': dict, # repeated status dict
    'source': str,
    'statusnet_conversation_id': int,
    'statusnet_html': str, # HTML contents of the notice
    'statusnet_in_groups': bool,
    'text': str, # plain text contents of the notice
    'truncated': bool,
    'uri': str,
    'user': dict # user dict
}
```

Group dict

```
{
    'admin_count': int,
    'blocked': bool,
    'created': str,
    'description': str,
    'fullname': str,
    'homepage': str,
    'homepage_logo': str,
    'id': int.
    'location': str,
    'member': bool,
    'member_count': int,
    'mini_logo': str,
    'modified': str,
    'nickname': str,
    'original_logo': str,
    'stream_logo': str,
    'url': str
}
```

Relationship dict

```
{
    'relationship':
    {
        'source':
        {
            'blocking': bool, # True if source user is blocking target user
            'followed_by': bool, # True if source user is followed by
                                 # target user
            'following': bool, # True if source user is following target
                               # user
            'id': int,
            'notifications_enabled': bool, # If notifications about target
                                           # user are enabled for source
                                           # user
            'screen_name': str
        },
        'target': dict # same as source
    }
}
```

Direct message dict

A dict returned by functions in *direct_messages* module.

```
{
    'relationship':
    {
        'source':
        {
            'blocking': bool, # True if source user is blocking target user
            'followed_by': bool, # True if source user is followed by
                                 # target user
            'following': bool, # True if source user is following target
                               # user
            'id': int,
            'notifications_enabled': bool, # If notifications about target
                                           # user are enabled for source
                                           # user
            'screen_name': str
        },
        'target': dict # same as source
    }
}
```

Config dict

A dict describing configuration of a GNU Social instance. Returned by *config.config()*.

```
{
    'attachments':
    {
        'file_quota': int, # maximum size of attachment in bytes
        'uploads': bool # True if users are allowed to upload files
```

```
    },
    'group':
    {
        'desclimit': int
    },
    'integration':
    {
        'source': str
    },
    'license':
    {
        'image': str,
        'owner': str,
        'title': str,
        'type': str,
        'url': str
    },
    'nickname':
    {
        'featured': list
    },
    'notice':
    {
        'contentlimit': int
    },
    'profile':
    {
        'biolimit': int
    },
    'site':
    {
        'broughtby': str,
        'broughtbyurl': str,
        'closed': bool,
        'email': str,
        'fancy': str,
        'inviteonly': bool,
        'language': str,
        'logo': str,
        'name': str,
        'path': str,
        'private': bool,
        'server': str,
        'ssl': str,
        'sslserver': str,
        'textlimit': str,
        'theme': str,
        'timezone': str
    },
    'throttle':
    {
        'count': int,
        'enabled': bool,
        'timespan': int
    },
    'url':
    {
        'maxnoticelength': int,
```

```
        'maxurllength': int
    },
    'xmpp':
    {
        'enabled': bool,
        'port': int,
        'server': str,
        'user': str
    }
}
```

ActivityStream dict

A dict returned by functions in *activity_streams* module.

```
{
  'generator': str, # string describing the server software
  'totalItems': int,
  'title': str, # string describing the result dict
  'links':
  [
      {
          'rel': 'alternate',
          'type': 'text/html',
          'url': str # URL with corresponding content
      }
  ],
  'items':
  [
      {
          'actor':
          {
              'displayName': str,
              'id': str, # URL to profile
              'image':
              {
                  'height': int,
                  'rel': 'avatar',
                  'type': str, # MIME type
                  'url': str,
                  'width': int
              },
              'objectType': 'person',
              'portablecontacts_net':
              {
                  'displayName': str,
                  'note': str, # user profile description
                  'preferredUsername': str, # screen name
                  'urls':
                  [
                      {
                          'primary': 'true',
                          'type': 'homepage',
                          'value': str, # URL to the homepage
                      }
                  ]
              },
```

```
            'status_net':
            {
                'avatarLinks':
                [
                    {
                        'height': int,
                        'rel': 'avatar',
                        'type': str, # MIME type
                        'url': str,
                        'width': int
                    }
                ],
                'profile_info':
                {
                    'local_id': str, # ID of the user on their home server
                }
            },
            'summary': str, # user profile description
            'url': str,
        },
        'content': str, # status in HTML
        'generator':
        {
            'id': str, # Example: 'tag:gs.smuglo.li,2016-11-
↪16:noticeId=1028155:objectType=comment'
            'objectType': 'application',
            'status_net':
            {
                'source_code': 'ostatus'
            }
        },
        'id': str, #
        'object':
        {
            'content': str, # status in HTML
            'id': str, # Example: 'tag:gs.smuglo.li,2016-11-
↪16:noticeId=1028155:objectType=comment'
            'inReplyTo':
            {
                'id': str, # Example: 'tag:gs.smuglo.li,2016-11-
↪16:noticeId=1028155:objectType=comment'
                'objectType': 'note',
                'url': str,
            },
            'objectType': 'comment',
            'status_net':
            {
                'notice_id': int,
            },
            'url': str,
        },
        'provider':
        {
            'displayName': str,
            'objectType': 'service',
            'url': str,
        },
        'published': str, # Example: '2016-11-16T20:42:18+00:00',
```

```
            'status_net':
            {
                'conversation': str, # Example:tag:social.heldscal.la,2016-11-
↪15:objectType=thread:nonce=c53d340f4850ca73 ,
                'notice_info':
                {
                    'local_id': str,
                    'source': 'ostatus'
                }
            },
            'to': # users, to whom the reply is addressed
            [
                {
                    'id': str, # URL to profile
                    'objectType': 'http://activitystrea.ms/schema/1.0/person'
                },
                {
                    'id': 'http://activityschema.org/collection/public',
                    'objectType': 'http://activitystrea.ms/schema/1.0/collection'
                }
            ],
            'url': str,
            'verb': str, # Example: 'post'
        }
    ]
}
```

Direct messages

`direct_messages.`**`new`**(*server_url*, **args*, ***kwargs*)

Sends a new direct message to the specified user from the authenticating user.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/direct-messages.html#post--api-direct_messages-new.json

Parameters **`server_url`** (*str*) – URL of the server

Return type dict

Returns *Direct message dict*

`direct_messages.`**`sent`**(*server_url*, **args*, ***kwargs*)

Returns the 20 most recent direct messages sent by the authenticating user. You can request up to 200 direct messages per call, and only the most recent 200 DMs will be available using this endpoint.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/direct-messages.html#get--api-direct_messages-sent.json

Parameters **`server_url`** (*str*) – URL of the server

Return type list

Returns a list of *direct message dicts*

direct_messages.**received**(*server_url*, **args*, ***kwargs*)

Returns the 20 most recent direct messages sent to the authenticating user. You can request up to 200 direct messages per call, and only the most recent 200 DMs will be available using this endpoint.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/direct-messages.html#get--api-direct_messages.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *direct message dicts*

Exceptions

exception ServerURLError(*server_url*)

Exception class for invalid server URLs.

Parameters **server_url** (*str*) – URL of the server

Friendships

friendships.**create**(*server_url*, **args*, ***kwargs*)

Subscribe to status updates from specified user.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/friendships.html#post--api-friendships-create.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

friendships.**destroy**(*server_url*, **args*, ***kwargs*)

Unsubscribe to status updates from specified user.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/friendships.html#post--api-friendships-destroy.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

friendships.**exists**(*server_url*, **args*, ***kwargs*)

Shows if `source_user` follows `target_user`.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/friendships.html#get--api-friendships-exists.json

Parameters **server_url** (*str*) – URL of the server

Return type bool

Returns `True` if `source_user` follows `target_user`. `False` otherwise.

friendships.**show**(*server_url*, **args*, ***kwargs*)
Returns detailed information about the relationship between two users.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/friendships.html#get--api-friendships-show.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *Relationship dict*

Groups

groups.**timeline**(*server_url*, *group*, **args*, ***kwargs*)
Shows a group's timeline. Similar to other timeline resources.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-timeline-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type list

Returns a list of *status dicts*

groups.**join**(*server_url*, *group*, **args*, ***kwargs*)
Join a group.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#post--api-statusnet-groups-join-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type dict

Returns *Group dict*

groups.**leave**(*server_url*, *group*, **args*, ***kwargs*)
Leave a group.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#post--api-statusnet-groups-leave-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type dict

Returns *Group dict*

groups.**create**(*server_url*, **args*, ***kwargs*)
Create a new group.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#post--api-statusnet-groups-create.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *Group dict*

groups.**show**(*server_url*, *group*, **args*, ***kwargs*)
Returns details about the group.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-show-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type dict

Returns *Group dict*

groups.**local_groups**(*server_url*, **args*, ***kwargs*)
List local groups.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-list_all.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *group dicts*

groups.**user_groups**(*server_url*, **args*, ***kwargs*)
Show the groups a given user is a member of.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-list.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *group dicts*

groups.**members**(*server_url*, *group*, **args*, ***kwargs*)
List the members of a given group.

Accepts the same parameters as `requests.get()` except `url`.

https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-membership-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type list

Returns a list of *user dicts*

groups.**is_member**(*server_url*, *group*, **args*, ***kwargs*)

Show is the specified user is a member of the group.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-is_member.json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type dict

Returns dict with `is_member` key set to `True` if the user is a member of the group, `False` otherwise

groups.**admins**(*server_url*, *group*, **args*, ***kwargs*)

List the admins of a given group.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/groups.html#get--api-statusnet-groups-admins-(int-id)(string-nickname).json

Parameters

- **server_url** (*str*) – URL of the server
- **group** – name or ID of the group

Return type list

Returns a list of *user dicts*

Media uploads

media.**upload**(*server_url*, **args*, ***kwargs*)

Uploads a file to server.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/media.html#post--api-media-upload.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns a dict with detals of the uploaded file

OAuth

oauth.**request_token**(*server_url*, **args*, ***kwargs*)

Returns resource owner key and secret for given consumer key and secret.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/oauth.html#post--api-oauth-request_token

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *OAuth dict*

oauth.**authorize_url**(*server_url*, *resource_owner_key*)
Returns a URL used to obtain user authorization for application access.

Parameters

- **server_url** (*str*) – URL of the server
- **resource_owner_key** (*str*) – a key used by user to authorize

Return type str

Returns a URL used to obtain user authorization for application access

oauth.**access_token**(*server_url*, **args*, ***kwargs*)
Returns resource owner key and secret for given consumer key and secret.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/oauth.html#post--api-oauth-access_token

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *OAuth dict*

oauth.**make_oauth_object**(*consumer_key*, *consumer_secret*, *resource_owner_key=None*, *resource_owner_secret=None*, *secret_key=None*)
Creates an `requests_oauthlib.OAuth1` object using given data.

Parameters

- **consumer_key** (*str*) – a value used by the application to identify itself to the GNU Social server
- **consumer_secret** (*str*) – a secret used by the application to establish ownership of the consumer key
- **resource_owner_key** (*str*) – a key used by user to authorize
- **resource_owner_secret** (*str*) – a secret used by the application to establish ownership of a given token
- **secret_key** (*str*) – a key given to you by GNU Social to get the access token

Return type `requests_oauthlib.OAuth1`

Search

search.**search**(*server_url*, *query*, **args*, ***kwargs*)
Returns a collection of statuses matching a specified query.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/search-api.html#get--api-search.json

Parameters

- **server_url** (*str*) – URL of the server

- **query** (*str*) – UTF-8 encoded query

Return type list

Returns a list of *status dicts*

Statuses

statuses.**update**(*server_url*, **args*, ***kwargs*)
Updates the authenticating user's current status.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/statuses.html#post--api-statuses-update.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *Status dict*

statuses.**show**(*server_url*, *status_id*, **args*, ***kwargs*)
Returns a specified status.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/statuses.html#get--api-statuses-show-(int-status_id).json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *Status dict*

statuses.**destroy**(*server_url*, *status_id*, **args*, ***kwargs*)
Deletes the status specified by the required ID parameter. The authenticating user must be the author of the specified status. Returns the destroyed status if successful.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/statuses.html#post--api-statuses-destroy-(int-status_id).json

Parameters

- **server_url** (*str*) – URL of the server
- **status_id** (*int*) – ID of the status to delete

Return type dict

Returns *Status dict*

statuses.**repeat**(*server_url*, *status_id*, **args*, ***kwargs*)
Repeats a status. Returns the original status with repeat details embedded.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/statuses.html#post--api-statuses-retweet-(int-status_id).json

Parameters

- **server_url** (*str*) – URL of the server
- **status_id** (*int*) – ID of the status to repeat

Return type dict

Returns *Status dict*

statuses.**conversation**(*server_url*, *conversation_id*, **args*, ***kwargs*)
Returns statuses that have been posted in the conversation

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/statuses.html#get--api-statuses-conversation-(int-conversation_id).json

Parameters

- **server_url** (*str*) – URL of the server
- **conversation_id** (*int*) – ID of the conversation to show

Return type list

Returns a list of *status dicts*

statuses.**favorite**(*server_url*, *status_id*, **args*, ***kwargs*)
Favorites the status specified in the ID parameter as the authenticating user. Returns the liked status when successful.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/favorites.html

Parameters

- **server_url** (*str*) – URL of the server
- **status_id** (*int*) – ID of the status to favorite

Return type dict

Returns *Status dict*

statuses.**unfavorite**(*server_url*, *status_id*, **args*, ***kwargs*)

Unfavorites the status specified in the ID parameter as the authenticating user. Returns the unliked status when successful.

Accepts the same parameters as `requests.post()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/favorites.html#post--api-favorites-destroy-(int-status_id).json

Parameters

- **server_url** (*str*) – URL of the server
- **status_id** (*int*) – ID of the status to unfavorite

Return type dict

Returns *Status dict*

statuses.**favorites**(*server_url*, **args*, ***kwargs*)
Returns recent notices favorited by the authenticating or specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/favorites.html#get--api-favorites.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

Timelines

`timelines.`**`public`**(*server_url*, **args*, ***kwargs*)

Returns the most recent notices, including repeats if they exist, from non-protected users.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-public_timeline.json

Parameters **`server_url`** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

`timelines.`**`home`**(*server_url*, **args*, ***kwargs*)

Returns the most recent notices, including repeats if they exist, posted by the authenticating user and the users they follow.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-home_timeline.json

Parameters **`server_url`** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

`timelines.`**`friends`**(*server_url*, **args*, ***kwargs*)

Home timeline for the specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-friends_timeline.json

Parameters **`server_url`** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

`timelines.`**`user`**(*server_url*, **args*, ***kwargs*)

Returns the most recent notices posted by the authenticating user. It is also possible to request another user's timeline by using the `screen_name` or `user_id` parameter. The other users timeline will only be visible if they are not protected, or if the authenticating user's follow request was accepted by the protected user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-user_timeline.json

Parameters **`server_url`** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

`timelines.`**`mentions`**(*server_url*, **args*, ***kwargs*)

Returns the most recent mentions (notices containing @username) for the authenticating user or specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-mentions_timeline.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

timelines.**replies**(*server_url*, **args*, ***kwargs*)

Returns the most recent mentions (notices containing @username) for the authenticating user or specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/timelines.html#get--api-statuses-replies.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *status dicts*

Users

users.**friends**(*server_url*, **args*, ***kwargs*)

Returns a collection of user objects for users followed by the specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/users.html#get--api-statuses-friends.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *user dicts*

users.**followers**(*server_url*, **args*, ***kwargs*)

Unsubscribe to status updates from specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/users.html#get--api-statuses-followers.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of *user dicts*

users.**show**(*server_url*, **args*, ***kwargs*)

Returns detailed information about the specified user.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/users.html#get--api-users-show.json

Parameters **server_url** (*str*) – URL of the server

Return type dict

Returns *User dict*

users.**friends_ids**(*server_url*, **args*, ***kwargs*)

Returns IDs of users the auntenticated or specified user is following (otherwise known as their "friends").

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/users.html#get--api-friends-ids.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of user IDs

users.**followers_ids**(*server_url*, **args*, ***kwargs*)

Returns IDs of users the auntenticated or specified user is followed by.

Accepts the same parameters as `requests.get()` except `url`.

Implements https://twitter-api.readthedocs.io/en/latest/users.html#get--api-followers-ids.json

Parameters **server_url** (*str*) – URL of the server

Return type list

Returns a list of user IDs

CHAPTER 3

Indices and tables

- genindex
- modindex

Python Module Index

Index

S

T

U

V