

# twitter (version 0.6)

[index](#)  
</Users/dewitt/svn/python-twitter/twitter.py>

A library that provides a python interface to the Twitter API

## Modules

[base64](#)  
[calendar](#)  
[os](#)

[rfc822](#)  
[simplejson](#)  
[sys](#)

[tempfile](#)  
[textwrap](#)  
[time](#)

[urllib](#)  
[urllib2](#)  
[urlparse](#)

## Classes

[builtin\\_.object](#)  
[Api](#)  
[DirectMessage](#)  
[Status](#)  
[User](#)  
[exceptions.Exception\(exceptions.BaseException\)](#)  
[TwitterError](#)

class **Api**([builtin\\_.object](#))

A python interface into the Twitter API

By default, the [Api](#) caches results for 1 minute.

Example usage:

To create an instance of the twitter.[Api](#) class, with no authentication:

```
>>> import twitter
>>> api = twitter.Api()
```

To fetch the most recently posted public twitter status messages:

```
>>> statuses = api.GetPublicTimeline()
>>> print [s.user.name for s in statuses]
[u'DeWitt', u'Kesuke Miyagi', u'ev', u'Buzz Andersen', u'Biz Stone'] #...
```

To fetch a single user's public status messages, where "user" is either a Twitter "short name" or their user id.

```
>>> statuses = api.GetUserTimeline(user)
>>> print [s.text for s in statuses]
```

To use authentication, instantiate the twitter.[Api](#) class with a username and password:

```
>>> api = twitter.Api(username='twitter user', password='twitter pass')
```

To fetch your friends (after being authenticated):

```
>>> users = api.GetFriends()
>>> print [u.name for u in users]
```

To post a twitter status message (after being authenticated):

```
>>> status = api.PostUpdate('I love python-twitter!')
>>> print status.text
I love python-twitter!
```

There are many other methods, including:

```
>>> api.PostUpdates(status)
>>> api.PostDirectMessage(user, text)
>>> api.GetUser(user)
>>> api.GetReplies()
>>> api.GetUserTimeline(user)
>>> api.GetStatus(id)
>>> api.DestroyStatus(id)
>>> api.GetFriendsTimeline(user)
>>> api.GetFriends(user)
>>> api.GetFollowers()
>>> api.GetFeatured()
>>> api.GetDirectMessages()
>>> api.PostDirectMessage(user, text)
>>> api.DestroyDirectMessage(id)
>>> api.DestroyFriendship(user)
>>> api.CreateFriendship(user)
>>> api.GetUserByEmail(email)
```

Methods defined here:

### **ClearCredentials(self)**

Clear the username and password for this instance

### **CreateFavorite(self, status)**

Favorites the status specified in the status parameter as the authenticating user. Returns the favorite status when successful.

The twitter.[Api](#) instance must be authenticated.

Args:

The twitter.[Status](#) instance to mark as a favorite.

Returns:

A twitter.[Status](#) instance representing the newly-marked favorite.

### **CreateFriendship(self, user)**

Befriends the user specified in the user parameter as the authenticating user.

The twitter.[Api](#) instance must be authenticated.

Args:

The ID or screen name of the user to befriend.

Returns:

A twitter.[User](#) instance representing the befriended user.

### **DestroyDirectMessage(self, id)**

Destroys the direct message specified in the required ID parameter.

The twitter.[Api](#) instance must be authenticated, and the authenticating user must be the recipient of the specified direct message.

Args:

id: The id of the direct message to be destroyed

Returns:

A twitter.[DirectMessage](#) instance representing the message destroyed

### **DestroyFavorite(self, status)**

Un-favorites the status specified in the ID parameter as the authenticating user.

Returns the un-favorited status in the requested format when successful.

The twitter.[Api](#) instance must be authenticated.

Args:

The twitter.[Status](#) to unmark as a favorite.

Returns:

A twitter.[Status](#) instance representing the newly-unmarked favorite.

### **DestroyFriendship(self, user)**

Discontinues friendship with the user specified in the user parameter.

The twitter.[Api](#) instance must be authenticated.

Args:

The ID or screen name of the user with whom to discontinue friendship.

Returns:

A twitter.[User](#) instance representing the discontinued friend.

### **DestroyStatus(self, id)**

Destroys the status specified by the required ID parameter.

The twitter.[Api](#) instance must be authenticated and the authenticating user must be the author of the specified status.

Args:

id: The numerical ID of the status you're trying to destroy.

Returns:

A twitter.[Status](#) instance representing the destroyed status message

### **GetDirectMessages(self, since=None, since\_id=None, page=None)**

Returns a list of the direct messages sent to the authenticating user.

The twitter.[Api](#) instance must be authenticated.

Args:

since:

Narrows the returned results to just those statuses created after the specified HTTP-formatted date. [optional]

since\_id:

Returns only public statuses with an ID greater than (that is, more recent than) the specified ID. [Optional]

Returns:

A sequence of twitter.[DirectMessage](#) instances

### **GetFeatured(self)**

Fetch the sequence of twitter.[User](#) instances featured on twitter.com

The twitter.[Api](#) instance must be authenticated.

Returns:

A sequence of twitter.[User](#) instances

### **GetFollowers(self, page=None)**

Fetch the sequence of twitter.[User](#) instances, one for each follower

The twitter.[Api](#) instance must be authenticated.

Returns:

A sequence of twitter.[User](#) instances, one for each follower

### **GetFriends(self, user=None, page=None)**

Fetch the sequence of twitter.[User](#) instances, one for each friend.

Args:

user: the username or id of the user whose friends you are fetching. If not specified, defaults to the authenticated user. [optional]

The twitter.[Api](#) instance must be authenticated.

Returns:

A sequence of twitter.[User](#) instances, one for each friend

**GetFriendsTimeline(self, user=None, count=None, since=None, since\_id=None)**

Fetch the sequence of twitter.[Status](#) messages for a user's friends

The twitter.[Api](#) instance must be authenticated if the user is private.

Args:

user:  
Specifies the ID or screen name of the user for whom to return the friends\_timeline. If unspecified, the username and password must be set in the twitter.[Api](#) instance. [Optional]

count:  
Specifies the number of statuses to retrieve. May not be greater than 200. [Optional]

since:  
Narrows the returned results to just those statuses created after the specified HTTP-formatted date. [Optional]

since\_id:  
Returns only public statuses with an ID greater than (that is, more recent than) the specified ID. [Optional]

Returns:

A sequence of twitter.[Status](#) instances, one for each message

**GetPublicTimeline(self, since\_id=None)**

Fetch the sequence of public twitter.[Status](#) message for all users.

Args:

since\_id:  
Returns only public statuses with an ID greater than (that is, more recent than) the specified ID. [Optional]

Returns:

An sequence of twitter.[Status](#) instances, one for each message

**GetReplies(self, since=None, since\_id=None, page=None)**

Get a sequence of status messages representing the 20 most recent replies (status updates prefixed with @username) to the authenticating user.

Args:

page:  
since:  
Narrows the returned results to just those statuses created after the specified HTTP-formatted date. [optional]

since\_id:  
Returns only public statuses with an ID greater than (that is, more recent than) the specified ID. [Optional]

Returns:

A sequence of twitter.[Status](#) instances, one for each reply to the user.

**GetStatus(self, id)**

Returns a single status message.

The twitter.[Api](#) instance must be authenticated if the status message is private.

Args:

id: The numerical ID of the status you're trying to retrieve.

Returns:

A twitter.[Status](#) instance representing that status message

### **GetUser(self, user)**

Returns a single user.

The twitter.[Api](#) instance must be authenticated.

Args:

user: The username or id of the user to retrieve.

Returns:

A twitter.[User](#) instance representing that user

### **GetUserByEmail(self, email)**

Returns a single user by email address.

Args:

email: The email of the user to retrieve.

Returns:

A twitter.[User](#) instance representing that user

### **GetUserTimeline(self, user=None, count=None, since=None, since\_id=None)**

Fetch the sequence of public twitter.[Status](#) messages for a single user.

The twitter.[Api](#) instance must be authenticated if the user is private.

Args:

user:

either the username (short\_name) or id of the user to retrieve. If not specified, then the current authenticated user is used. [optional]  
count: the number of status messages to retrieve [optional]

since:

Narrows the returned results to just those statuses created after the specified HTTP-formatted date. [optional]

since\_id:

Returns only public statuses with an ID greater than (that is, more recent than) the specified ID. [Optional]

Returns:

A sequence of twitter.[Status](#) instances, one for each message up to count

### **PostDirectMessage(self, user, text)**

Post a twitter direct message from the authenticated user

The twitter.[Api](#) instance must be authenticated.

Args:

user: The ID or screen name of the recipient user.

text: The message text to be posted. Must be less than 140 characters.

Returns:

A twitter.[DirectMessage](#) instance representing the message posted

### **PostUpdate(self, status, in\_reply\_to\_status\_id=None)**

Post a twitter status message from the authenticated user.

The twitter.[Api](#) instance must be authenticated.

Args:

**status:**  
The message text to be posted. Must be less than or equal to 140 characters.

**in\_reply\_to\_status\_id:**  
The ID of an existing status that the status to be posted is in reply to. This implicitly sets the `in_reply_to_user_id` attribute of the resulting status to the user ID of the message being replied to. Invalid/missing status IDs will be ignored. [Optional]

**Returns:**  
A twitter.[Status](#) instance representing the message posted.

### **PostUpdates(self, status, continuation=None, \*\*kwargs)**

Post one or more twitter status messages from the authenticated user.

Unlike `api.PostUpdate`, this method will post multiple status updates if the message is longer than 140 characters.

The twitter.[Api](#) instance must be authenticated.

**Args:**

**status:**  
The message text to be posted. May be longer than 140 characters.

**continuation:**  
The character string, if any, to be appended to all but the last message. Note that Twitter strips trailing `'...'` strings from messages. Consider using the unicode `\u2026` character (horizontal ellipsis) instead. [Defaults to None]

**\*\*kwargs:**  
See `api.PostUpdate` for a list of accepted parameters.

**Returns:**  
A of list twitter.[Status](#) instance representing the messages posted.

### **SetCache(self, cache)**

Override the default cache. Set to None to prevent caching.

**Args:**  
**cache:** an instance that supports the same API as the `twitter._FileCache`

### **SetCacheTimeout(self, cache\_timeout)**

Override the default cache timeout.

**Args:**  
**cache\_timeout:** time, in seconds, that responses should be reused.

### **SetCredentials(self, username, password)**

Set the username and password for this instance

**Args:**  
**username:** The twitter username.  
**password:** The twitter password.

### **SetSource(self, source)**

Suggest the "from source" value to be displayed on the Twitter web site.

The value of the 'source' parameter must be first recognized by the Twitter server. New source values are authorized on a case by case basis by the Twitter development team.

**Args:**  
**source:**  
The source name as a string. Will be sent to the server as the 'source' parameter.

**SetUrllib(self, urllib)**

Override the default urllib implementation.

Args:

urllib: an instance that supports the same API as the urllib2 module

**SetUserAgent(self, user\_agent)**

Override the default user agent

Args:

user\_agent: a string that should be send to the server as the [User-agent](#)

**SetXTwitterHeaders(self, client, url, version)**

Set the X-Twitter HTTP headers that will be sent to the server.

Args:

client:

The client name as a string. Will be sent to the server as the 'X-Twitter-Client' header.

url:

The URL of the meta.xml as a string. Will be sent to the server as the 'X-Twitter-Client-URL' header.

version:

The client version as a string. Will be sent to the server as the 'X-Twitter-Client-Version' header.

**\_\_init\_\_(self, username=None, password=None, input\_encoding=None, request\_headers=None)**

Instantiate a new twitter. [Api object](#).

Args:

username: The username of the twitter account. [optional]

password: The password for the twitter account. [optional]

input\_encoding: The encoding used to encode input strings. [optional]

request\_header: A dictionary of additional HTTP request headers. [optional]

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

Data and other attributes defined here:

**DEFAULT\_CACHE\_TIMEOUT = 60**

**class DirectMessage([\\_\\_builtin\\_\\_.object](#))**

A class representing the [DirectMessage](#) structure used by the twitter API.

The [DirectMessage](#) structure exposes the following properties:

```
direct_message.id
direct_message.created_at
direct_message.created_at_in_seconds # read only
direct_message.sender_id
direct_message.sender_screen_name
direct_message.recipient_id
direct_message.recipient_screen_name
```

```
direct_message.text
```

Methods defined here:

**AsDict(self)**

A dict representation of this twitter.[DirectMessage](#) instance.

The return value uses the same key names as the JSON representation.

Return:

A dict representing this twitter.[DirectMessage](#) instance

**AsJsonString(self)**

A JSON string representation of this twitter.[DirectMessage](#) instance.

Returns:

A JSON string representation of this twitter.[DirectMessage](#) instance

**GetCreatedAt(self)**

Get the time this direct message was posted.

Returns:

The time this direct message was posted

**GetCreatedAtInSeconds(self)**

Get the time this direct message was posted, in seconds since the epoch.

Returns:

The time this direct message was posted, in seconds since the epoch.

**GetId(self)**

Get the unique id of this direct message.

Returns:

The unique id of this direct message

**GetRecipientId(self)**

Get the unique recipient id of this direct message.

Returns:

The unique recipient id of this direct message

**GetRecipientScreenName(self)**

Get the unique recipient screen name of this direct message.

Returns:

The unique recipient screen name of this direct message

**GetSenderId(self)**

Get the unique sender id of this direct message.

Returns:

The unique sender id of this direct message

**GetSenderScreenName(self)**

Get the unique sender screen name of this direct message.

Returns:

The unique sender screen name of this direct message

**GetText(self)**



Get the text of this direct message.

Returns:

The text of this direct message.

### **SetCreatedAt(self, created\_at)**

Set the time this direct message was posted.

Args:

created\_at: The time this direct message was created

### **SetId(self, id)**

Set the unique id of this direct message.

Args:

id: The unique id of this direct message

### **SetRecipientId(self, recipient\_id)**

Set the unique recipient id of this direct message.

Args:

recipient id: The unique recipient id of this direct message

### **SetRecipientScreenName(self, recipient\_screen\_name)**

Set the unique recipient screen name of this direct message.

Args:

recipient\_screen\_name: The unique recipient screen name of this direct message

### **SetSenderId(self, sender\_id)**

Set the unique sender id of this direct message.

Args:

sender id: The unique sender id of this direct message

### **SetSenderScreenName(self, sender\_screen\_name)**

Set the unique sender screen name of this direct message.

Args:

sender\_screen\_name: The unique sender screen name of this direct message

### **SetText(self, text)**

Set the text of this direct message.

Args:

text: The text of this direct message

### **\_\_eq\_\_(self, other)**

### **\_\_init\_\_(self, id=None, created\_at=None, sender\_id=None, sender\_screen\_name=None, recipient\_id=None, recipient\_screen\_name=None, text=None)**

An [object](#) to hold a Twitter direct message.

This class is normally instantiated by the twitter.[Api](#) class and returned in a sequence.

Note: Dates are posted in the form "Sat Jan 27 04:17:38 +0000 2007"

Args:

id: The unique id of this direct message

created\_at: The time this direct message was posted

sender\_id: The id of the twitter user that sent this message

sender\_screen\_name: The name of the twitter user that sent this message  
 recipient\_id: The id of the twitter that received this message  
 recipient\_screen\_name: The name of the twitter that received this message  
 text: The text of this direct message

**\_\_ne\_\_**(self, other)

**\_\_str\_\_**(self)

A string representation of this twitter.[DirectMessage](#) instance.

The return value is the same as the JSON string representation.

Returns:

A string representation of this twitter.[DirectMessage](#) instance.

---

Static methods defined here:

**NewFromJsonDict**(data)

Create a new instance based on a JSON dict.

Args:

data: A JSON dict, as converted from the JSON in the twitter API

Returns:

A twitter.[DirectMessage](#) instance

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

**created\_at**

The time this direct message was posted.

**created\_at\_in\_seconds**

The time this direct message was posted, in seconds since the epoch

**id**

The unique id of this direct message.

**recipient\_id**

The unique recipient id of this direct message.

**recipient\_screen\_name**

The unique recipient screen name of this direct message.

**sender\_id**

The unique sender id of this direct message.

**sender\_screen\_name**

The unique sender screen name of this direct message.

**text**

The text of this direct message

## class **Status**([\\_\\_builtin\\_\\_](#).object)

A class representing the [Status](#) structure used by the twitter API.

The [Status](#) structure exposes the following properties:

```
status.created_at
status.created_at_in_seconds # read only
status.favorited
status.in_reply_to_screen_name
status.in_reply_to_user_id
status.in_reply_to_status_id
status.truncated
status.source
status.id
status.text
status.relative_created_at # read only
status.user
```

Methods defined here:

### **AsDict**(self)

A dict representation of this twitter.[Status](#) instance.

The return value uses the same key names as the JSON representation.

Return:

A dict representing this twitter.[Status](#) instance

### **AsString**(self)

A JSON string representation of this twitter.[Status](#) instance.

Returns:

A JSON string representation of this twitter.[Status](#) instance

### **GetCreatedAt**(self)

Get the time this status message was posted.

Returns:

The time this status message was posted

### **GetCreatedAtInSeconds**(self)

Get the time this status message was posted, in seconds since the epoch.

Returns:

The time this status message was posted, in seconds since the epoch.

### **GetFavorited**(self)

Get the favorited setting of this status message.

Returns:

True if this status message is favorited; False otherwise

### **GetId**(self)

Get the unique id of this status message.

Returns:

The unique id of this status message

### **GetInReplyToScreenName**(self)

### **GetInReplyToStatusId**(self)

**GetInReplyToUserId(self)****GetNow(self)**

Get the wallclock time for this status message.

Used to calculate relative\_created\_at. Defaults to the time the [object](#) was instantiated.

Returns:

Whatever the status instance believes the current time to be, in seconds since the epoch.

**GetRelativeCreatedAt(self)**

Get a human redable string representing the posting time

Returns:

A human readable string representing the posting time

**GetSource(self)****GetText(self)**

Get the text of this status message.

Returns:

The text of this status message.

**GetTruncated(self)****GetUser(self)**

Get a twitter.[User](#) representing the entity posting this status message.

Returns:

A twitter.[User](#) representing the entity posting this status message

**SetCreatedAt(self, created\_at)**

Set the time this status message was posted.

Args:

created\_at: The time this status message was created

**SetFavorited(self, favorited)**

Set the favorited state of this status message.

Args:

favorited: boolean True/False favorited state of this status message

**SetId(self, id)**

Set the unique id of this status message.

Args:

id: The unique id of this status message

**SetInReplyToScreenName(self, in\_reply\_to\_screen\_name)****SetInReplyToStatusId(self, in\_reply\_to\_status\_id)****SetInReplyToUserId(self, in\_reply\_to\_user\_id)****SetNow(self, now)**

Set the wallclock time for this status message.

Used to calculate `relative_created_at`. Defaults to the time the [object](#) was instantiated.

Args:

`now`: The wallclock time for this instance.

**SetSource**(self, source)

**SetText**(self, text)

Set the text of this status message.

Args:

`text`: The text of this status message

**SetTruncated**(self, truncated)

**SetUser**(self, user)

Set a twitter.[User](#) representing the entity posting this status message.

Args:

`user`: A twitter.[User](#) representing the entity posting this status message

**\_\_eq\_\_**(self, other)

**\_\_init\_\_**(self, created\_at=None, favorited=None, id=None, text=None, user=None, in\_reply\_to\_screen\_name=None, in\_reply\_to\_user\_id=None, in\_reply\_to\_status\_id=None, truncated=None, source=None, now=None)

An [object](#) to hold a Twitter status message.

This class is normally instantiated by the twitter.[Api](#) class and returned in a sequence.

Note: Dates are posted in the form "Sat Jan 27 04:17:38 +0000 2007"

Args:

`created_at`: The time this status message was posted

`favorited`: Whether this is a favorite of the authenticated user

`id`: The unique id of this status message

`text`: The text of this status message

`relative_created_at`:

A human readable string representing the posting time

`user`:

A twitter.[User](#) instance representing the person posting the message

`now`:

The current time, if the client choses to set it. Defaults to the wall clock time.

**\_\_ne\_\_**(self, other)

**\_\_str\_\_**(self)

A string representation of this twitter.[Status](#) instance.

The return value is the same as the JSON string representation.

Returns:

A string representation of this twitter.[Status](#) instance.

---

Static methods defined here:

**NewFromJsonDict**(data)

Create a new instance based on a JSON dict.

Args:  
    data: A JSON dict, as converted from the JSON in the twitter API  
Returns:  
    A twitter.[Status](#) instance

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

**created\_at**

The time this status message was posted.

**created\_at\_in\_seconds**

The time this status message was posted, in seconds since the epoch

**favorited**

The favorited state of this status message.

**id**

The unique id of this status message.

**in\_reply\_to\_screen\_name**

**in\_reply\_to\_status\_id**

**in\_reply\_to\_user\_id**

**now**

The wallclock time for this status instance.

**relative\_created\_at**

Get a human readable string representing the posting time

**source**

**text**

The text of this status message

**truncated**

**user**

A twitter.[User](#) representing the entity posting this status message

class **TwitterError**([exceptions.Exception](#))

Base class for Twitter errors

Method resolution order:

[TwitterError](#)

[exceptions.Exception](#)

[exceptions.BaseException](#)

[\\_\\_builtin\\_\\_.object](#)


---

Data descriptors defined here:

[\\_\\_weakref\\_\\_](#)

list of weak references to the object (if defined)

**message**

Returns the first argument used to construct this error.

---

Methods inherited from [exceptions.Exception](#):

[\\_\\_init\\_\\_](#)(...)

x.[\\_\\_init\\_\\_](#)(...) initializes x; see x.[\\_\\_class\\_\\_](#).[\\_\\_doc\\_\\_](#) for signature

---

Data and other attributes inherited from [exceptions.Exception](#):

[\\_\\_new\\_\\_](#) = <built-in method [\\_\\_new\\_\\_](#) of type object at 0x1e8a40>

T.[\\_\\_new\\_\\_](#)(S, ...) -> a new [object](#) with type S, a subtype of T

---

Methods inherited from [exceptions.BaseException](#):

[\\_\\_delattr\\_\\_](#)(...)

x.[\\_\\_delattr\\_\\_](#)('name') <==> del x.name

[\\_\\_getattr\\_\\_](#)(...)

x.[\\_\\_getattr\\_\\_](#)('name') <==> x.name

[\\_\\_getitem\\_\\_](#)(...)

x.[\\_\\_getitem\\_\\_](#)(y) <==> x[y]

[\\_\\_getslice\\_\\_](#)(...)

x.[\\_\\_getslice\\_\\_](#)(i, j) <==> x[i:j]

Use of negative indices is not supported.

[\\_\\_reduce\\_\\_](#)(...)[\\_\\_repr\\_\\_](#)(...)

x.[\\_\\_repr\\_\\_](#)() <==> repr(x)

[\\_\\_setattr\\_\\_](#)(...)

x.[\\_\\_setattr\\_\\_](#)('name', value) <==> x.name = value

[\\_\\_setstate\\_\\_](#)(...)[\\_\\_str\\_\\_](#)(...)

x.[\\_\\_str\\_\\_](#)() <==> str(x)

---

Data descriptors inherited from [exceptions.BaseException](#):

[\\_\\_dict\\_\\_](#)**args**

**class User([\\_\\_builtin\\_\\_](#).object)**

A class representing the [User](#) structure used by the twitter API.

The [User](#) structure exposes the following properties:

```
user.id
user.name
user.screen_name
user.location
user.description
user.profile_image_url
user.profile_background_tile
user.profile_background_image_url
user.profile_sidebar_fill_color
user.profile_background_color
user.profile_link_color
user.profile_text_color
user.protected
user.utc_offset
user.time_zone
user.url
user.status
user.statuses_count
user.followers_count
user.friends_count
user.favourites_count
```

Methods defined here:

**AsDict(self)**

A dict representation of this twitter.[User](#) instance.

The return value uses the same key names as the JSON representation.

Return:

A dict representing this twitter.[User](#) instance

**AsJsonString(self)**

A JSON string representation of this twitter.[User](#) instance.

Returns:

A JSON string representation of this twitter.[User](#) instance

**GetDescription(self)**

Get the short text description of this user.

Returns:

The short text description of this user

**GetFavouritesCount(self)**

Get the number of favourites for this user.

Returns:

The number of favourites for this user.

**GetFollowersCount(self)**

Get the follower count for this user.

Returns:

The number of users following this user.

**GetFriendsCount(self)**



Get the friend count for this user.

Returns:

The number of users this user has befriended.

### **GetId(self)**

Get the unique id of this user.

Returns:

The unique id of this user

### **GetLocation(self)**

Get the geographic location of this user.

Returns:

The geographic location of this user

### **GetName(self)**

Get the real name of this user.

Returns:

The real name of this user

### **GetProfileBackgroundColor(self)**

### **GetProfileBackgroundImageUrl(self)**

### **GetProfileBackgroundTile(self)**

Boolean for whether to tile the profile background image.

Returns:

True if the background is to be tiled, False if not, None if unset.

### **GetProfileImageUrl(self)**

Get the url of the thumbnail of this user.

Returns:

The url of the thumbnail of this user

### **GetProfileLinkColor(self)**

### **GetProfileSidebarFillColor(self)**

### **GetProfileTextColor(self)**

### **GetProtected(self)**

### **GetScreenName(self)**

Get the short username of this user.

Returns:

The short username of this user

### **GetStatus(self)**

Get the latest twitter.[Status](#) of this user.

Returns:

The latest twitter.[Status](#) of this user

### **GetStatusesCount(self)**

Get the number of status updates for this user.

Returns:

The number of status updates for this user.

### **GetTimeZone(self)**

Returns the current time zone string for the user.

Returns:

The descriptive time zone string for the user.

### **GetUrl(self)**

Get the homepage url of this user.

Returns:

The homepage url of this user

### **GetUtcOffset(self)**

### **SetDescription(self, description)**

Set the short text description of this user.

Args:

description: The short text description of this user

### **SetFavouritesCount(self, count)**

Set the favourite count for this user.

Args:

count: The number of favourites for this user.

### **SetFollowersCount(self, count)**

Set the follower count for this user.

Args:

count: The number of users following this user.

### **SetFriendsCount(self, count)**

Set the friend count for this user.

Args:

count: The number of users this user has befriended.

### **SetId(self, id)**

Set the unique id of this user.

Args:

id: The unique id of this user.

### **SetLocation(self, location)**

Set the geographic location of this user.

Args:

location: The geographic location of this user

### **SetName(self, name)**

Set the real name of this user.

Args:

name: The real name of this user

### **SetProfileBackgroundColor(self, profile\_background\_color)**

**SetProfileBackgroundImageUrl**(self, profile\_background\_image\_url)

**SetProfileBackgroundTile**(self, profile\_background\_tile)

Set the boolean flag for whether to tile the profile background image.

Args:

profile\_background\_tile: Boolean flag for whether to tile or not.

**SetProfileImageUrl**(self, profile\_image\_url)

Set the url of the thumbnail of this user.

Args:

profile\_image\_url: The url of the thumbnail of this user

**SetProfileLinkColor**(self, profile\_link\_color)

**SetProfileSidebarFillColor**(self, profile\_sidebar\_fill\_color)

**SetProfileTextColor**(self, profile\_text\_color)

**SetProtected**(self, protected)

**SetScreenName**(self, screen\_name)

Set the short username of this user.

Args:

screen\_name: the short username of this user

**SetStatus**(self, status)

Set the latest twitter.[Status](#) of this user.

Args:

status: The latest twitter.[Status](#) of this user

**SetStatusesCount**(self, count)

Set the status update count for this user.

Args:

count: The number of updates for this user.

**SetTimeZone**(self, time\_zone)

Sets the user's time zone string.

Args:

time\_zone: The descriptive time zone to assign for the user.

**SetUrl**(self, url)

Set the homepage url of this user.

Args:

url: The homepage url of this user

**SetUtcOffset**(self, utc\_offset)

**\_\_eq\_\_**(self, other)

**\_\_init\_\_**(self, id=None, name=None, screen\_name=None, location=None, description=None, profile\_image\_url=None, profile\_background\_tile=None, profile\_background\_image\_url=None, profile\_sidebar\_fill\_color=None, profile\_background\_color=None, profile\_link\_color=None, profile\_text\_color=None, protected=None, utc\_offset=None, time\_zone=None, followers\_count=None,

friends\_count=None, statuses\_count=None, favourites\_count=None, url=None, status=None)

**\_\_ne\_\_**(self, other)

**\_\_str\_\_**(self)

A string representation of this twitter.[User](#) instance.

The return value is the same as the JSON string representation.

Returns:

A string representation of this twitter.[User](#) instance.

---

Static methods defined here:

**NewFromJsonDict**(data)

Create a new instance based on a JSON dict.

Args:

data: A JSON dict, as converted from the JSON in the twitter API

Returns:

A twitter.[User](#) instance

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

**description**

The short text description of this user.

**favourites\_count**

The number of favourites for this user.

**followers\_count**

The number of users following this user.

**friends\_count**

The number of friends for this user.

**id**

The unique id of this user.

**location**

The geographic location of this user.

**name**

The real name of this user.

**profile\_background\_color**

**profile\_background\_image\_url**

The url of the profile background of this user.

**profile\_background\_tile**

Boolean for whether to tile the background image.

**profile\_image\_url**

The url of the thumbnail of this user.

**profile\_link\_color****profile\_sidebar\_fill\_color****profile\_text\_color****protected****screen\_name**

The short username of this user.

**status**

The latest twitter.Status of this user.

**statuses\_count**

The number of updates for this user.

**time\_zone**

Returns the current time zone string for the user.

Returns:

The descriptive time zone string for the user.

**url**

The homepage url of this user.

**utc\_offset**

## Functions

**md5** = openssl\_md5(...)

Returns a md5 hash [object](#); optionally initialized with a string

## Data

**CHARACTER\_LIMIT** = 140

**\_\_author\_\_** = 'dewitt@google.com'

**\_\_version\_\_** = '0.6'

## Author

dewitt@google.com