
coo Documentation

Release 0.1.0

wilfredinni

Aug 14, 2021

Contents

1	User Guide	3
1.1	Schedule Twitter Updates	3
1.2	Schedule a list of strings	5
1.3	The Twitter API	7
1.4	Change Log	8
1.5	Indices and tables	9

Coo is an easy to use Python library for scheduling Twitter updates. To use it, you need to first apply for a developer account in the [Twitter Developers Platform](#) and generate the Keys and Access Tokens.

```
from coo import Coo

at = Coo(
    "consumer_key",
    "consumer_secret",
    "access_token",
    "access_token_secret"
)

tweets = [
    ("2030-12-05 16:30", template, "Awesome Twitter update."),
    ("2030-10-28 18:50", template, "Another awesome Twitter update."),
    ("2030-10-29 18:15", template2, "One more update."),
    ("2030-11-01 13:45", None, "Twitter update without a template."),
]

at.schedule(tweets, time_zone="America/Santiago")
```

Or you can use a list of strings and add a delay, interval and template:

```
tweets = [
    "My first awesome Twitter Update",
    "My second awesome Twitter Update",
    "My third awesome Twitter Update",
    "My fourth awesome Twitter Update",
    "My fifth awesome Twitter Update",
    "My sixth awesome Twitter Update",
]

at.tweet(tweets, delay="13:45", interval="four_hours", template=my_template)
```


1.1 Schedule Twitter Updates

Schedule updates with *datetime* strings or integers and use custom *Templates* if needed.

```
Coo.schedule(updates, time_zone)
```

Full example:

```
from coo import Coo

at = Coo(
    "consumer_key",
    "consumer_secret",
    "access_token",
    "access_token_secret"
)

tweets = [
    # datetime with and without templates
    ("2030-10-28 18:50", template, "My Twitter update with a template."),
    ("2030-10-29 18:15", template2, "Update with a different template."),
    ("2030-11-01 13:45", None, "Twitter update without a template."),

    # date with and without templates
    ("2030-12-25", template3, "Merry christmas!"),
    ("2031-01-01", None, "And a happy new year!"),

    # time with and without templates
    ("18:46", template2, "Will be post today at 18:46."),
    ("23:00", None, "A tweet for today at 23:00."),

    # integer (seconds) with and without templates
    (3600, template, "This tweet will be posted in an hour."),
```

(continues on next page)

(continued from previous page)

```
(86400, None, "This one, tomorrow at the same hour."),
]

at.schedule(tweets, time_zone="America/Santiago")
```

1.1.1 Parsing DateTime strings

Note:

- If a time zone is not specified, it will set to *local*.
- The time will be set to 00:00:00 if it's not specified.
- When passing only time information the date will default to today.
- A future date is needed, otherwise a *ScheduleError* is raised.

Here you can find all the [Time Zones](#).

1.1.2 Media Files

There are two ways to add media files to your tweets. The first and easiest is to use one global file for all the updates:

```
at.schedule(tweets, time_zone="America/Santiago", media="path/to/file.png")
```

Also, an individual file can be set for each one of the updates:

```
tweets = [
    ("2030-10-28 18:50", template, "Update with an image.", "pics/owl.png"),
    ("2030-10-29 18:15", template, "Update with other media.", "videos/funny_video.mp4
↪"),
    ("2030-11-01 13:45", template, "Tweet without media."),
]
```

Finally, it is possible to combine these two ways. For example, if most of the tweets are gonna use the same media and just a few will have a different or none:

```
tweets = [
    ("2030-11-01 13:45", template, "Tweet with global media."),
    ("2030-11-02 13:45", template, "Tweet with global media."),
    ("2030-11-03 13:45", template, "Tweet with global media."),
    ("2030-11-04 13:45", template, "Tweet with global media."),
    ("2030-11-05 13:45", template, "Tweet with global media."),
    ("2030-11-06 13:45", template, "Tweet with global media."),
    ("2030-11-07 13:45", template, "Tweet with global media."),
    ("2030-11-08 13:45", template, "Tweet without media.", None),
    ("2030-11-09 13:45", template, "Tweet without media.", None),
    ("2030-12-10 18:50", template, "Update with an image.", "pics/owl.png"),
    ("2030-12-11 18:15", template, "Update with other media.", "videos/funny_video.mp4
↪"),
]

at.schedule(tweets, time_zone="America/Santiago", media="path/to/global_media.png")
```

1.1.3 Templates

You can set different templates for each one of your updates, or none.

```
tweets = [
    # datetime with and without templates
    ("2030-10-28 18:50", template, "My Twitter update with a template."),
    ("2030-10-29 18:15", template2, "Update with a different template."),
    ("2030-11-01 13:45", None, "Twitter update without a template."),
]

at.schedule(tweets, time_zone="America/Santiago")
```

Templates are very simple, just use a multiline string and add a *\$message* where you want your message to appear.

```
template = """My awesome header

$message

#python #coding #coo
"""
```

1.2 Schedule a list of strings

Post ordered updates with *Delay*, *Interval*, and a *Template* if needed.

```
Coo.tweet(updates, delay, interval, template, time_zone)
```

```
from coo import Coo

at = Coo(
    "consumer_key",
    "consumer_secret",
    "access_token",
    "access_token_secret"
)

tweets = [
    "My first awesome Twitter Update",
    "My second awesome Twitter Update",
    "My third awesome Twitter Update",
    "My fourth awesome Twitter Update",
    "My fifth awesome Twitter Update",
    "My sixth awesome Twitter Update",
]

# post the twitter updates
at.tweet(tweets)
```

1.2.1 Delay

You can use datetime, date and time strings, integers as seconds and some *Keywords*: *half_hour*, *one_hour*, *one_day* and *one_week* between others to delay the post of your first update.

```
# datetime, date and time strings
at.tweet(tweets, delay="2030-11-24 13:45", time_zone="America/Santiago")
at.tweet(tweets, delay="2030-11-24", time_zone="Australia/Sydney")
at.tweet(tweets, delay="13:45", time_zone="America/New_York")

# "keywords"
at.tweet(tweets, delay="one_week")

# integer
at.tweet(tweets, delay=604800)
```

Note: When parsing DateTime strings:

- If a time zone is not specified, it will set to *local*.
 - The time will be set to 00:00:00 if it's not specified.
 - When passing only time information the date will default to today.
 - A future date is needed, otherwise a *ScheduleError* is raised.
-

Here you can find all the [Time Zones](#).

1.2.2 Interval

Use integers as seconds or some strings as *Keywords*: `half_hour`, `one_hour`, `one_day` and `one_week` between others.

```
# "keywords"
at.tweet(tweets, interval="four_hours")

# integers
at.tweet(tweets, interval=14400)
```

1.2.3 Media files

Use one media file for all of your updates:

```
at.tweet(tweets, media="path/to/media.jpeg")
```

1.2.4 Random updates

To tweet your updates randomly:

```
at.tweet(tweets, aleatory=True)
```

1.2.5 Keywords

Keyword	Seconds
now	0
half_hour	1800
one_hour	3600
two_hours	7200
four_hours	14400
six_hours	21600
eight_hours	28800
ten_hours	36000
twelve_hours	43200
fourteen_hours	50400
sixteen_hours	57600
eighteen_hours	64800
twenty_hours	72000
twenty_two_hours	79200
one_day	86400
two_days	172800
three_days	259200
four_days	345600
five_days	432000
six_days	518400
one_week	604800

1.2.6 Template

You can also set one template for each one of the updates.

```
at.tweet(tweets, template=template)
```

Templates are very simple, just use a multiline string and add a *\$message* where you want your message to appear.

```
template = """My aswesome header

$message

#python #coding #coo
"""
```

1.3 The Twitter API

Coo is written using the [Python Twitter](#) wrapper, and through *Coo.api* you gain access to all of his models:

```
# get your followers
followers = at.api.GetFollowers()

# get your direct messages
d_messages = at.api.GetDirectMessages()
```

(continues on next page)

(continued from previous page)

```
# favorited tweets
favorites = at.api.GetFavorites()

# mentions
mentions = at.api.GetMentions()

# retweets
retweets = at.api.GetRetweets()
```

And a lot more. If you are interested, check their [documentation](#).

1.4 Change Log

1.4.1 0.1.3 - 2018-12-13

Added

- Added support to post updates randomly on `Coo.tweet(aleartory=True)`.
- Added support for updates with a single media file for all tweets on `Coo.tweet()`.
- Added support for updates with a single media file for all tweets on `Coo.schedule()`.
- Added support for updates with a different media file for each tweet on `Coo.schedule()`.

1.4.2 0.1.2 - 2018-11-29

Added

- Added changes to CHANGELOG.md.
- Added changes to the documentation.
- Added releases.

Fixed

- Fixed template overwriting the tweets when `$message` is not provided.
- Fixed lots of typos.

1.4.3 0.1.1 - 2018-11-21

Fixed

- Fixed typos and README_Prst for PyPI.

1.4.4 0.1.0 - 2018-11-21

Initial Release

1.5 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)