
tomputils Documentation

Release 1.12.17

Tom Parker

Jul 16, 2019

Contents:

1	tomputils.mattermost	3
2	tomputils.downloader	5
3	tomputils.message	7
4	tomputils.util	9
5	Command line tools	11
6	Indices and tables	15
	Python Module Index	17
	Index	19

A collection of utility modues I find useful.

CHAPTER 1

tomputils.mattermost

2.1 tomputils.downloader

A simple segmenting downloader.

license CC0 1.0 Universal <http://creativecommons.org/publicdomain/zero/1.0/>

`tomputils.downloader.fetch(req_url, output=None)`

Fetch a single URL using default settings.

Parameters `req_url` (*unicode or str*) – URL to request. File will be written to the current working directory.

class `tomputils.downloader.Downloader(max_retry=5, min_seg_size=16384, max_con=4)`

Bases: `future.types.newobject.newobject`

Download a file, possibly in segments.

Parameters

- **max_retry** (*int, optional*) – Maximum attempts that will be made to retrieve a segment.
- **min_seg_size** (*int, optional*) – Largest file size, in bytes, that will not trigger segmenting.
- **max_con** (*int, optional*) – Maximum number of concurrent connections to the remote server.

`fetch(req_url, output=None)`

Fetch a file.

Parameters

- **req_url** (*str*) – URL of the file to retrieve
- **output** (*str, optional*) – filename, possibly with path, of the downloaded file.
- **TODO** (*test can_segment == false*) –

3.1 tomputils.message

Interface with a RabbitMQ message broker.

license CC0 1.0 Universal <http://creativecommons.org/publicdomain/zero/1.0/>

CHAPTER 4

tomputils.util

5.1 downloader

Download a file of HTTP or HTTPS, in concurrent segments if supported by the remote server. Usage:

```
usage: downloader [-h] [-r RETRIES] [-n NUM_CON] [-s SEG_SIZE] [-v] url

Provides a console interface for downloading a file, possibly in segments.

positional arguments:
  url                  URL of file to download.

optional arguments:
  -h, --help            show this help message and exit
  -r RETRIES, --retries RETRIES
                        Maximum number of attempts to fulfill request
  -n NUM_CON, --num-con NUM_CON
                        Maximum number of concurrent requests to the remote
                        server
  -s SEG_SIZE, --seg-size SEG_SIZE
                        Largest file size, in bytes, that will not trigger
                        segmenting.
  -v, --verbose         Verbose logging
```

5.2 mattermost.py

Interact with a mattermost server. Usage:

```
usage: mattermost [-h] [-a ATTACHMENTS] [-r RETRIES] [-t TIMEOUT]
                  [--team-name TEAM_NAME] [--channel-name CHANNEL_NAME] [-v]
                  {post,getteams,getchannels}
```

(continues on next page)

(continued from previous page)

Interact **with** a Mattermost server. Not **all** possible combinations of arguments will make sense, avoid those that do **not** make sense. The message to post, **if** **any**, will be read **from** <STDIN>.

positional arguments:

{post, getteams, getchannels}
Command

optional arguments:

-h, --help show this help message **and** exit
 -a ATTACHMENTS, --attachments ATTACHMENTS
 File to attach. Argument may be repeated to attach multiple files.
 -r RETRIES, --retries RETRIES
 Maximum number of attempts to fulfill request
 -t TIMEOUT, --timeout TIMEOUT
 request timeout
 --team-name TEAM_NAME
 Mattermost team name. Will override MATTERMOST_TEAM_ID environment variable.
 --channel-name CHANNEL_NAME
 Mattermost channel name. Will override MATTERMOST_CHANNEL_ID environment variable.
 -v, --verbose Verbose logging

5.3 singleTimeout.sh

Kill a job started with single.py if it has been running too long. Usage:

```
Usage: ./singleTimeout.sh [-g] [-v] -t <timeout in seconds> [ -m <email addr>] [ -f
↪<lockfile> ] -c <command>"
```

Kill a long-running job started **with** single.py.

required arguments:

-c COMMAND Command **as** passed to single.py
 -t TIMEOUT Time, **in** seconds, job **is** allowed to run

optional arguments:

-f LOCKFILE Path to the lock file. If **not** specified, use the default. If a lockfile **is** provided **and** the job has been running too long, I will attempt to remove the lockfile after killing the job.
 -m ADDRESS Address to email when a job **is** killed
 -g Kill job by group **id** rather than process **id**
 -v Print more stuff

5.4 configupdater

Update a collection of configuration files stored in a remote repository. See the annotated config file for details. Usage:


```
usage: configupdater.py [-h] [--svnurl SVNURL | --url URL] [--user USER]
                        [--passwd PASSWD]
                        config
```

I look after a config file.

positional arguments:

config Local config path

optional arguments:

-h, --help show this help message **and** exit
--svnurl SVNURL Subversion URL of config file
--url URL URL of config file
--user USER Username
--passwd PASSWD password

CHAPTER 6

Indices and tables

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

t

`tomputils.downloader`, [5](#)
`tomputils.message`, [7](#)

D

`Downloader` (*class in `tomputils.downloader`*), [5](#)

F

`fetch()` (*in module `tomputils.downloader`*), [5](#)

`fetch()` (*`tomputils.downloader.Downloader` method*),
[5](#)

T

`tomputils.downloader` (*module*), [5](#)

`tomputils.message` (*module*), [7](#)