
utilpy Documentation

Release 0.0.1

AI2

Dec 07, 2018

Contents

1	Install	3
2	Utilspy Examples	5
3	Examples:	7
3.1	Utilpy Examples	7
4	Utilpy Packages:	9
4.1	Utilpy Packages	9
5	Indices and tables	13
	Python Module Index	15

Collection of python utils.

CHAPTER 1

Install

`pip install utilpy`

CHAPTER 2

Utilspy Examples

```
import time

from utilspy import log, files, decorator

# logger funcation
logs = log.Logger()

logs.log("< started >")
time.sleep(2)
logs.log("running..")
time.sleep(1)
logs.log("</>")

# timeout
@decorator.timeout(5)
def add(x, y):
    return x+y

# other useful utils
files = utils.walk_directory('.')
```

Examples:

3.1 Utilpy Examples

3.1.1 Install

pip install utilpy

3.1.2 Examples

```
import time

from utilpy import log, files, decorator

# logger funcation
logs = log.Logger()

logs.log("< started >")
time.sleep(2)
logs.log("running..")
time.sleep(1)
logs.log("</>")

# timeout
@decorator.timeout(5)
def add(x, y):
    return x+y

# other useful utils
files = utils.walk_directory('.')
```


4.1 Utilpy Packages

4.1.1 utilpy.utils

`utilpy.utils.clean_str(string)`

Tokenization/string cleaning for all datasets except for SST. Original taken from https://github.com/yoonkim/CNN_sentence/blob/master/process_data.py

Args:

string:str string of sentences

Returns: Lowercase cleaned string:str

`utilpy.utils.download_file(url, save_dir='tmp/', filename=None)`

download and save the file

Args:

url:str url of file

save_dir:str directory to save

filename:str Name of file to save as

`utilpy.utils.download_files(urls, save_dir='tmp/')`

download all files from urls

Args:

urls:list urls of file

save_dir:str directory to save

`utilpy.utils.format_filename(s)`

Take a string and return a valid filename constructed from the string. Uses a whitelist approach: any characters not present in `valid_chars` are removed. Also spaces are replaced with underscores.

Args:

s:str string

Returns: Valid filename : str

`utilpy.utils.logger(logger_name='utilpy.utils', filename='log.log', level=10)`
logger for logging

Args:

logger_name:str name of logger

filename:str path and file name

level:str logging level

Return: logger function for logging:logger

`utilpy.utils.page_soup(url)`
Make BeautifulSoup of page

Args:

url:str url of page

Returns: BeautifulSoup of page:soup

`utilpy.utils.read_config(filename)`
Parse config yaml file

Args:

filename:str path of config file

`utilpy.utils.top_n(lis, n)`
Get index of top n item from list with value

Args:

lis:list list

n:int number of top value

Return: dict of top value and index: dict

`utilpy.utils.walk_directory(dir_path, files_only=True)`
Get list of all file in directory recursive

Args:

dir_path:str parent directory to start

files_only:bool get list of files only other

Return: If file only than list of all files otherwise files and list of directories

4.1.2 utilpy.decorator

`utilpy.decorator.profile(fnc)`
A decorator that uses cProfile to profile a function And print the result

`utilpy.decorator.threading_d(func)`
A decorator to run function in background on thread

Args:

func: function Function with args

Return: background_thread: Thread

`utilpy.decorator.timeout (seconds=10, error_message='Timer expired')`

Decorator to throw timeout error, if function doesnt complete in certain time

Args:

seconds: int No of seconds to wait

error_message: str Error message

4.1.3 utilpy.log

Logger function that print time required in each session of logger

class `utilpy.log.Logger`

Bases: object

Print the log message with time required between each session New session should start with “<” and end with “>” session ends with “</>”

log (*message*)

log message new session starts with < and ends with > session end with </>

4.1.4 utilspy.files

`utilpy.files.delete_dirs (folder_names)`

delete folder and all its contents

Args: ‘

folder_names: str‘ or **list** string or list of folders

`utilpy.files.make_dirs (folder_names)`

make folder if it doesnot exist

Args:

folder_names: str or list string or list of folders

`utilpy.files.move_file (filename, out_dir)`

move file/dir

Args:

filename: str filename of file to be moved or name of dir to be moved

out_dir: str output directory

`utilpy.files.parse_path (path, fn_only=False, ext=False, al=True)`

get the directory from filename,

Args:

path: str path of file

fn_only: bool get the file name only from path

ext: bool split file name into name and extension

al: bool get list of dir and file name

Returns: list of value: list

`utilpy.files.unzip_file(filename, save_dir='data')`
unzip all file

Args:

filename:str filename of zip file

sav_dir:str parent directory to zip

Warning: for rar file type, install rar and unrar

```
apt install rar && apt install unrar
```

`utilpy.files.zip_folder(dir_path, filename=None, exclude=None, include=None)`
zip all file in folder,

Args:

dir_path:str parent directory to zip

filename:str filename of zip file

exclude:list list of file to exclude

include:list list of file to include

CHAPTER 5

Indices and tables

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

u

- `utilpy.decorator`, [10](#)
- `utilpy.files`, [11](#)
- `utilpy.log`, [11](#)
- `utilpy.utils`, [9](#)

C

`clean_str()` (in module `utilpy.utils`), 9

D

`delete_dirs()` (in module `utilpy.files`), 11

`download_file()` (in module `utilpy.utils`), 9

`download_files()` (in module `utilpy.utils`), 9

F

`format_filename()` (in module `utilpy.utils`), 9

L

`log()` (`utilpy.log.Logger` method), 11

`Logger` (class in `utilpy.log`), 11

`logger()` (in module `utilpy.utils`), 10

M

`make_dirs()` (in module `utilpy.files`), 11

`move_file()` (in module `utilpy.files`), 11

P

`page_soup()` (in module `utilpy.utils`), 10

`parse_path()` (in module `utilpy.files`), 11

`profile()` (in module `utilpy.decorator`), 10

R

`read_config()` (in module `utilpy.utils`), 10

T

`threading_d()` (in module `utilpy.decorator`), 10

`timeout()` (in module `utilpy.decorator`), 11

`top_n()` (in module `utilpy.utils`), 10

U

`unzip_file()` (in module `utilpy.files`), 12

`utilpy.decorator` (module), 10

`utilpy.files` (module), 11

`utilpy.log` (module), 11

`utilpy.utils` (module), 9

W

`walk_directory()` (in module `utilpy.utils`), 10

Z

`zip_folder()` (in module `utilpy.files`), 12