
zippyshare-downloader

Release 0.0.21

Rahman Yusuf

Sep 29, 2021

CONTENTS

1	Command Line Interface (CLI) Usage	1
1.1	App names	1
1.2	Options	1
2	API Documentation	3
	Index	7

COMMAND LINE INTERFACE (CLI) USAGE

1.1 App names

There is few app names in zippyshare-downloader:

- zippyshare-dl
- zippyshare-downloader

Note: If none of above doesn't work use this

```
# For Windows
py -3 -m zippyshare_downloader

# For Linux
python3 -m zippyshare_downloader
```

1.2 Options

1.2.1 Global options

- ZIPPYSHARE_URL or FILE Zippyshare URL or file containing zippyshare urls
- --no-download No download file
- --verbose, -v Enable verbose
- --replace, -r Replace file if exist
- --silent No output

1.2.2 File / Folder related

- `--folder FOLDER` Store downloaded file in given folder
- `--filename FILENAME` Rewrite filename. Will be ignored if using multiple urls

1.2.3 Zip and unzip

- `--zip FILENAME, -z FILENAME` Zip all downloaded files
- `--unzip, -uz` Unzip all downloaded files, one by one

Warning: Option `--unzip` or `-uz` will only work if you are using multiple zippyshare urls

For example:

```
# This will work
$ zippyshare-dl "urls.txt" --unzip

# This will NOT work
$ zippyshare-dl "https://www.zippyshare.com/v/..." --unzip
```

1.2.4 Download related

- `--async` Run zippyshare-downloader in asynchronous mode
- `--fast` Enable fast download

Note: When you using `--async` option, the download is faster than without using `--async` one.

Warning: `--fast` option require `--async`. It will throw error if you specified `--fast` without `--async` option.

For example:

```
# This will work
$ zippyshare-dl "https://www.zippyshare.com/v/..." --async --fast

# This will NOT work
$ zippyshare-dl "https://www.zippyshare.com/v/..." --fast
```

API DOCUMENTATION

`zippyshare_downloader.extract_info(url, download=True, unzip=False, **kwargs)`

Extract all informations in Zippyshare url.

Parameters

- **url** (`str`) – Zippyshare url.
- **download** (`bool`) – Download given zippyshare url if True, default to True.
- **unzip** (`bool`) – Unzip downloaded file once finished (if given file is zip or tar format extract it, otherwise ignore it), default to False.
- ****kwargs** – These parameters will be passed to `File.download()`

Returns Zippyshare file

Return type `File`

`zippyshare_downloader.download(*urls, zip=None, unzip=False, **kwargs)`

Download multiple zippyshare urls

Parameters

- ***urls** – Zippyshare urls.
- **zip** (`str`) – Zip all downloaded files once finished. Zip filename will be taken from zip parameter, default to None. NOTE: You can't mix zip and unzip options together with value True, it will raise error.
- **unzip** (`bool`) – Unzip all downloaded files once finished (if given file is zip format extract it, otherwise ignore it), default to False. NOTE: You can't mix zip and unzip options together with value True, it will raise error.
- ****kwargs** – These parameters will be passed to `File.download()`, except for parameter filename.

Returns a list of Zippyshare files

Return type `List[File]`

`async zippyshare_downloader.extract_info_coro(url, download=True, unzip=False, **kwargs)`

Extract all informations in Zippyshare url.

Parameters

- **url** (`str`) – Zippyshare url.
- **download** (`bool`) – Download given zippyshare url if True, default to True.
- **unzip** (`bool`) – Unzip downloaded file once finished (if given file is zip or tar format extract it, otherwise ignore it), default to False.

- ****kwargs** – These parameters will be passed to [File.download\(\)](#)

Returns Zippyshare file

Return type [File](#)

async zippyshare_downloader.**download_coro**(*urls, zip=None, unzip=False, **kwargs)

“Coroutine Function”

Download multiple zippyshare urls

Parameters

- ***urls** ([str](#)) – Zippyshare urls.
- **zip** ([str](#)) – Zip all downloaded files once finished. Zip filename will be taken from zip, default to None. NOTE: You can’t mix zip and unzip options together with value True, it will raise error.
- **unzip** ([bool](#)) – Unzip all downloaded files once finished (if given file is zip format extract it, otherwise ignore it), default to False. NOTE: You can’t mix zip and unzip options together with value True, it will raise error.
- ****kwargs** – These parameters will be passed to [File.download\(\)](#), except for parameter filename.

Returns a list of Zippyshare files

Return type List[[File](#)]

class zippyshare_downloader.**File**(data)

property **date_uploaded**

Return date that this file uploaded.

download(progress_bar=True, replace=False, folder=None, filename=None)

Download this file

Parameters

- **progress_bar** ([bool](#)) – Enable/Disable progress bar, default to *True*
- **replace** ([bool](#)) – Replace file if exist, default to *False*
- **folder** ([str](#)) – Set a folder where to store downloaded file, default to *None*.
- **filename** ([str](#)) – Set a replacement filename, default to *None*.

Returns Zippyshare file downloaded

Return type Path

async **download_coro**(progress_bar=True, replace=False, folder=None, filename=None, fast=False)

Same like [File.download\(\)](#) but for asynchronous process

Parameters

- **progress_bar** ([bool](#)) – Enable/Disable progress bar, default to *True*
- **replace** ([bool](#)) – Replace file if exist, default to *False*
- **folder** ([str](#)) – Set a folder where to store downloaded file, default to *None*.
- **filename** ([str](#)) – Set a replacement filename, default to *None*.
- **fast** ([bool](#)) – Enable Fast download, default to *False*

Returns Zippyshare file downloaded

Return type Path

property download_url

Return downloadable url

property name

Return name of the file

property size

Return size of the file, in bytes.

to_JSON()

Return all zippyshare informations in JSON

to_dict()

Return all zippyshare informations in dict

INDEX

D

`date_uploaded` (*zippyshare_downloader.File* property), 4

`download()` (*in module zippyshare_downloader*), 3

`download()` (*zippyshare_downloader.File* method), 4

`download_coro()` (*in module zippyshare_downloader*), 4

`download_coro()` (*zippyshare_downloader.File* method), 4

`download_url` (*zippyshare_downloader.File* property), 5

E

`extract_info()` (*in module zippyshare_downloader*), 3

`extract_info_coro()` (*in module zippyshare_downloader*), 3

F

`File` (*class in zippyshare_downloader*), 4

N

`name` (*zippyshare_downloader.File* property), 5

S

`size` (*zippyshare_downloader.File* property), 5

T

`to_dict()` (*zippyshare_downloader.File* method), 5

`to_JSON()` (*zippyshare_downloader.File* method), 5