anime-downloader Documentation

Release 3.5.0

Vishnunarayan K I

Sep 12, 2022

Contents:

1 Features **3**

2 Supported Sites **5**
- 2.1 Installation . . . 6
- 2.2 `anime dl` . . . 9
- 2.3 `anime test` . . . 10
- 2.4 `watch` . . . 10
- 2.5 Config . . . 12
- 2.6 anime config . . . 12
- 2.7 config.json . . . 14
- 2.8 Site Configs . . . 16
- 2.9 Library usage . . . 17
- 2.10 Writing your own custom site class . . . 17
- 2.11 Base classes . . . 18
- 2.12 Helper functions . . . 20

Python Module Index **23**

Index **25**

Anime-Downloader helps you download your favorite anime

CHAPTER 1

Features

- Download or stream any episode or episode range of any anime.
- Specify the quality you want to stream or download.
- Search and download.
- Save yourselves from those malicious ads.
- Download using external downloader ([aria2](https://aria2.github.io/) recommended).
- Configurable using *config.json*. See [doc](https://github.com/anime-dl/anime-downloader/wiki/Config).

CHAPTER 2

Supported Sites

- 4Anime
- AnimeBinge
- Animedaisuki
- Animeflix
- Animeflv
- Animefreak
- AnimeKisa
- AnimeOnline360
- animeout
- Animerush
- Animesimple
- Animevibe
- AnimeTake
- AniTube
- Animixplay
- Anistream
- Darkanime
- Dbanimes
- EraiRaws
- EgyAnime - usually m3u8 (good for streaming, not so much for downloading)
- FastAni
- GurminderBoparai (AnimeChameleon)

- itsaturday
- Justdubs
- Kickassanime
- Kissanimefree
- KissanimeX
- Kisscartoon - requires Selenium
- Nyaa.si
- PutLockers
- RyuAnime
- SubsPlease
- twist.moe - requires Node.js
- tenshi.moe
- Vidstream
- Voiranime
- Vostfree

NOTE: To download from sites marked *[cloudflare]*, anime-downloader has to be installed with cloudflare support(See below);

2.1 Installation

This document provides complete installation instructions for the following systems: Windows, Mac, Linux, Android, iOS and Windows Choco. Please contact the dev team if you find any errors or inconsistencies in this document.

2.1.1 Windows

Please see this video: https://www.youtube.com/watch?v=gC2tctOL5I8

2.1.2 Automatic Windows install

Add the following to a file named install.bat and then run it as Administrator;

```
::This installs choco, as found on https://chocolatey.org/docs/installation
@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -
↪InputFormat None -ExecutionPolicy Bypass -Command " [System.Net.
↪ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.
↪WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH
↪%;%ALLUSERSPROFILE%\chocolatey\bin"

choco install -y git mpv python3 aria2 nodejs
refreshenv && pip3 install -U git+https://github.com/anime-dl/anime-downloader.git &&␣
↪echo Testing providers, the install is done && anime test
```

2.1.3 Windows via `choco`

Contributed by @CodaTheOtaku

NOTE: Ensure the Command Prompt (cmd) is being ran as Administrator.

- Install Chocolatey Package manager.
- Using the Chocolatey Package Manager;

```
choco install -y git mpv python3 aria2 nodejs
```

- Once these are installed;

```
pip3 install -U git+https://github.com/anime-dl/anime-downloader.git
```

- Then, the commands to view a show would be;

```
anime dl "showname" --provider *Insert provider name* --new --play *mpv or vlc*
```

2.1.4 Mac

Anime-Downloader is available from brew via the following command;

```
brew install anime-downloader
```

2.1.5 Linux

If you are using Linux, Python is probably already installed. Type `pip --version` into your terminal. If the command returns python2, replace all the following `pip` with `pip3`.

- Install aria2.
- Install Anime-Downloader via the following command;

```
pip3 install anime-downloader
```

- To install the bleeding-edge version of Anime-Downloader use this alternative command;:

  pip3 install -U git+https://github.com/anime-dl/anime-downloader.git

- Enjoy.

2.1.6 Android

This does not require a rooted device to work.

- Install Termux or similar Terminal Emulator.
- Install Aria2c via the following command if using Termux;

```
pkg install aria2
```

- Install Python via the following command if using Termux;

```
pkg install python
```

- Install git via the following command if using Termux;

```
pkg install git
```

- Install Anime-Downloader via the following command after python and git are installed;

```
pip3 install -U git+https://github.com/anime-dl/anime-downloader.git
```

- The usage commands should now match the commands used on PC.

2.1.7 iOS (Jailbreak Required)

A jailbroken iPhone is required. Visit r/jailbreak on Reddit for the latest jailbreak news and information.

Tinkering May be Required.

- Install the following packages from the mcapollo repo (https://mcapollo.github.io/Public/) using your favorite package manager;

```
Aria2, Python@3.7, git, nano (for File editing) and a Terminal Emulator (NewTerm␣
↪is an example of this).
```

- Open your Terminal Emulator and type in the following command;

```
python3 -m ensurepip
```

- Pip and setuptools should now be installed.

The following steps install Anime-Downloader; - To install Anime-Downloader before the PR which moves an unsupported module into extras follow along, if not, skip to **(continue)**;

- Firstly, clone the repository via this command;

```
git clone https://github.com/anime-dl/anime-downloader.git
```

- Next, change your directory into the cloned repo. To do so, use the following case-sensitive command;

```
cd anime-downloader
```

- Following this, type in the following command to start editing the file;

```
nano setup.py
```

- Navigate to the following line using the arrow keys;

```
'pycryptodome>=3.8.2',
```

- Delete the highlighted line as to match the image below;

- Press ctrl+o then enter then press ctrl+X.
- If all the steps were performed correctly then you should be back to the command line.
- **(continue)** Type the following command to install the project;

```
python3 setup.py install
```

- Wait for the line to finish, then the program functions the same as the PC version.

2.2 `anime dl`

Note: It is recommended to use the external downloader functionality with aria2c because it will be faster than the internal downloader: Use the argument `-xd "{aria2}"`

Note: Run `anime dl --help` to get the latest cli options.

2.2.1 Search and download

- To search and download all episodes use the following command;

```
anime dl 'code geass'
```

To search on animepahe,

```
anime dl 'code geass' --provider animepahe
```

Run `anime dl --help` for help using the `dl` subcommand.

2.2.2 Download directly

- To download Fullmetal Alchemist: Brotherhood all episodes;

```
anime dl 'https://animepahe.com/anime/fullmetal-alchemist-brotherhood'
```

- To download Fullmetal Alchemist: Brotherhood episode 1;

```
anime dl 'https://animepahe.com/anime/fullmetal-alchemist-brotherhood' --episodes 1
```

- To download Fullmetal Alchemist: Brotherhood episode 1 to 20;

```
anime dl 'https://animepahe.com/anime/fullmetal-alchemist-brotherhood' --episodes 1:21
```

- To get stream url of Fullmetal Alchemist: Brotherhood episode 1;

```
anime dl 'https://animepahe.com/anime/fullmetal-alchemist-brotherhood' --url --
↪episodes 1
```

- To play using vlc. (On windows use path to exe);

```
anime dl 'https://animepahe.com/anime/fullmetal-alchemist-brotherhood' --play vlc --
↪episodes 1
```

2.3 anime test

You can check which provider has your favorite anime and check which provider is currrently working.

2.3.1 Usage

- To check the providers with code geass

2.4 watch

Note: You need mpv to use this subcommand currently. Work is being done to add VLC as a player also, but will have reduced functionality.

Fig. 1: anime watch in action

`anime watch` can be a all in one solution for your anime needs. Anime watch contains exactly the same providers used in anime dl so you will not be missing anything!

`watch` will locally track your progress with an anime list making your time with anime watch easy and enjoyable. Anime lists from MyAnimeList can also be imported into your list, so you need not worry about losing a complete tracker!

Fig. 2: Revamped watch command with an imported MyAnimeList list!!

```
CONTROLS IN MPV:
> : Next episode
< : Previous episode
q : Quit
```

```
$ anime watch --help
Usage: anime watch [OPTIONS] [ANIME_NAME]

  With watch you can keep track of any anime you watch.

  Using anime watch without any options:
     Using anime watch without any options will return the complete tracking list.␣
↪Allowing for quick access to the list.

  Available Commands inside a list:
     swap  : Swap the list
     add   : Add new anime

  Available Commands after selection of an anime:
    set    : Set episodes_done, provider,title, anime_status and stars. Ex: set␣
↪episodes_done=3
    remove : Remove selected anime from watch list.
    update : Update the episodes of the currrent anime, brilliant for currently␣
↪airing series.
    watch  : Watch selected anime.
    download : Download episodes of selected anime.
    back : Return back to the list.

Options:
  -n, --new                         Add a new entry to the list.
  -l, --list  [all,completed,watching,dropped,planned]
                                    Return a list of the anime from the tracker, sorted␣
↪by anime_status.
  -r, --remove                      Remove a specified anime from the list.
  --mal_import [FILEPATH]                    Give a .xml MAL file to populate the␣
↪list with MAL.
  -q, --quality [360p|480p|720p|1080p]
                                    Specify the quality of episodes.
  -ll, --log-level [DEBUG|INFO|WARNING|ERROR]
                                    Sets the level of logger.
  --help                            Show this message and exit.
```

2.4.1 anime watch --new

This command adds an anime to your watch list.

If you run `anime watch --new`, you will be prompted to enter a search term. If you already know what to search for, use `anime watch <search term> --new`. You can then select an anime to be added to the watch list.

If you are familiar with the dl side of anime downloader then you can also specify your choice of provider at this point with the `--provider` flag.

```
$ anime watch 'code geass' --new
 1: Code Geass: Nunnally in Wonderland          OVA
 2: Code Geass: Fukkatsu no Lelouch             Preview | Ep 2 Previ
 3: Code Geass: Lelouch of the Rebellion (Du    DUB | Ep 25/25
 4: Code Geass: Hangyaku no Lelouch Recaps      Special | Ep 2/2
 5: Code Geass: Lelouch of the Rebellion R2     Ep 25/25
 6: Code Geass: Hangyaku no Lelouch III - Ou    Preview
 7: Code Geass: Hangyaku no Lelouch I - Koud    Movie
 8: Code Geass: Hangyaku no Lelouch II - Han    Preview
 9: Code Geass: Lelouch of the Rebellion        Ep 25/25
10: Code Geass: Lelouch of the Rebellion R2     DUB | Ep 25/25

Enter the anime no:  [1]: 9
INFO:Selected Code Geass: Lelouch of the Rebellion
INFO:Extracting episode info from page
INFO:Added Code Geass: Lelouch of the Rebellion to watch list.
```

2.4.2 anime watch --list

This command lists your watch list.

```
$ anime watch --list
 SlNo |                Name                 |   Eps    |    Type
-------------------------------------------------------------------
    1 | Code Geass: Lelouch of the Rebellio |    0/25  | TV Series
```

You can select an anime from this list and perform an action on it. ‘‘‘

Once you select an anime from the table a new enviroment for you to use appears, this has the following options;

- set: Update information about the anime on the list. Episodes, title and provider changes go here.
- remove: Remove an anime from the list.
- update: Update the episode range of the anime.
- watch: Watch an episode of the anime and then return back to this enviroment.
- download: Download an episode of the anime.
- back: Return back to the list

2.5 Config

You can configure the tool to suit your needs by using `anime config` or by editing the config.json.

2.6 anime config

This command lets you change the configuration for anime dl from within your terminal.

```
+--------+------------+
|      4 | watch      |
|      3 | siteconfig |
|      2 | ezdl       |
|      1 | dl         |
|--------+------------|
|   SlNo | settings   |
+--------+------------+
Select Option [1]:
```

This lists all options in the config, and lets you change their values by entering them. If you select 1 (one) then it will bring all the sub-keys for that selected choice.

```
+--------+-------------------------------------+
|     16 | url                                 |
|     15 | speed_limit                         |
|     14 | skip_download                       |
|     13 | selescrape_driver_binary_path       |
|     12 | selescrape_browser_executable_path  |
|     11 | selescrape_browser                  |
|     10 | quality                             |
|      9 | provider                            |
|      8 | player                              |
|      7 | force_download                      |
|      6 | file_format                         |
|      5 | fallback_qualities                  |
|      4 | external_downloader                 |
|      3 | download_dir                        |
|      2 | chunk_size                          |
|      1 | aria2c_for_torrents                 |
|--------+-------------------------------------|
|   SlNo | dl settings                         |
+--------+-------------------------------------+
Select Option [1]:
```

Selecting a setting that has no sub-keys will get you to the value input mode.

```
+--------+-------------------------------------+
|     16 | url                                 |
|     15 | speed_limit                         |
|     14 | skip_download                       |
|     13 | selescrape_driver_binary_path       |
|     12 | selescrape_browser_executable_path  |
|     11 | selescrape_browser                  |
|     10 | quality                             |
|      9 | provider                            |
|      8 | player                              |
|      7 | force_download                      |
|      6 | file_format                         |
|      5 | fallback_qualities                  |
|      4 | external_downloader                 |
|      3 | download_dir                        |
|      2 | chunk_size                          |
|      1 | aria2c_for_torrents                 |
|--------+-------------------------------------|
|   SlNo | dl settings                         |
+--------+-------------------------------------+
```

(continues on next page)

(continued from previous page)

```
Select Option [1]:
Current value: False
Input new value for aria2c_for_torrents: True
```

2.7 config.json

If you want you can directly edit the config.json. It can be found under:

- `~/.config/anime-downloader` on Linux
- `%appdata%\anime downloader` on Windows
- `~/Library/Application Support/anime downloader` on MacOS

The default config file is given below.

```
{
    "dl": {
        "aria2c_for_torrents": false,
        "chunk_size": "10",
        "download_dir": ".",
        "external_downloader": "{aria2}",
        "fallback_qualities": [
            "720p",
            "480p",
            "360p"
        ],
        "file_format": "{anime_title}/{anime_title}_{ep_no}",
        "force_download": false,
        "player": null,
        "provider": "twist.moe",
        "quality": "1080p",
        "skip_download": false,
        "url": false
    },
    "siteconfig": {
        "animedaisuki": {
            "servers": [
                "official"
            ]
        },
        "animeflix": {
            "fallback_servers": [
                "FastStream"
            ],
            "server": "AUEngine",
            "version": "sub"
        },
        "animeflv": {
            "server": "natsuki",
            "version": "subbed"
        },
        "animekisa": {
            "fallback_servers": [
                "mp4upload",
```

(continues on next page)

(continued from previous page)

```
            "vidstream"
        ],
        "server": "gcloud"
    },
    "animerush": {
        "fallback_servers": [
            "MP4Upload",
            "Mp4upload Video",
            "Youruploads Video"
        ],
        "server": "Mp4uploadHD Video"
    },
    "animesimple": {
        "server": "trollvid",
        "version": "subbed"
    },
    "anistream.xyz": {
        "version": "subbed"
    },
    "dreamanime": {
        "server": "trollvid",
        "version": "subbed"
    },
    "dubbedanime": {
        "servers": [
            "vidstream",
            "mp4upload",
            "trollvid"
        ],
        "version": "dubbed"
    },
    "gogoanime": {
        "server": "cdn"
    },
    "kickass": {
        "ext_fallback_servers": [
            "Mp4Upload",
            "Vidcdn",
            "Vidstreaming"
        ],
        "fallback_servers": [
            "ORIGINAL-QUALITY-V2",
            "HTML5-HQ",
            "HTML5",
            "A-KICKASSANIME",
            "BETAPLAYER",
            "KICKASSANIME",
            "DEVSTREAM"
        ],
        "server": "A-KICKASSANIME"
    },
    "nineanime": {
        "server": "mp4upload"
    },
    "ryuanime": {
        "server": "trollvid",
        "version": "subbed"
```

(continues on next page)

(continued from previous page)

```
        },
        "vidstream": {
            "servers": [
                "vidstream",
                "gcloud",
                "mp4upload",
                "cloud9",
                "hydrax"
            ]
        },
        "watchmovie": {
            "fallback_servers": [
                "fembed",
                "yourupload",
                "mp4upload"
            ],
            "server": "gcloud"
        }
    },
    "watch": {
        "fallback_qualities": [
            "720p",
            "480p",
            "360p"
        ],
        "log_level": "INFO",
        "provider": "twist.moe",
        "quality": "1080p"
    }
}
```

Note:

- For the key `file_format`, you can set `anime_title`(which refers to the title of the anime) and `ep_no` which is the number of the episode.
- If you set `player` in `dl`, the tool will never download, only play in the set player.
- If you set `force_download`, it will redownload even if the episode has already been downloaded.

2.7.1 Choosing preferred server

`server` contains the preferred server.

`servers` contains a list of servers, used in order. Set the preferred server by changing the order of the list.

`fallback_servers` contains a list of servers to use if the primary server is not found, used in order.

2.8 Site Configs

The following are the sites with configs. These can be configured using the config file. The keys and corresponding possible values are mentioned below;

class anime_downloader.sites.animeflv.**Animeflv**(*url=None*, *quality='720p'*, *fallback_qualities=None*, *_skip_online_data=False*, *subbed=None*)

Nice things

version: subbed or latin subbed for subbed latin for Spanish

server: one of below natsuki, streamango

class anime_downloader.sites.anistream.**Anistream**(*url=None*, *quality='720p'*, *fallback_qualities=None*, *_skip_online_data=False*, *subbed=None*)

Site: http://anistream.xyz

version: One of ['subbed', 'dubbed] Selects the version of audio of anime.

2.9 Library usage

Anime Downloader also be used as a library.

The following code searches for 'one punch' from twist.moe;

get_anime_class() can be used to import specific sites using the url one of *sitenames*.

```
from anime_downloader.sites import get_anime_class

Twist = get_anime_class('twist.moe')
search = Twist.search('one punch')
print(search[0].title)

# You can directly import twist too
from anime_downloader.sites.twistmoe import TwistMoe
anime = TwistMoe(search[0].url)
print(anime)
print(len(anime))

# Get first episodes url
print(anime[0].source().stream_url)
```

In the above example, *TwistMoe* is a concrete implementation of *anime_downloader.sites.anime.Anime*. Search results is a list of *anime_downloader.sites.anime.SearchResult*.

2.10 Writing your own custom site class

anime_downloader is built with easy extensibility in mind.

Each of the sites (in the tool) can roughly be classfied into two catagories;

- Sites which don't use Cloudflare DDoS protection: Ex: NineAnime
- Sites which use Cloudflare DDoS protection: Ex: KissAnime

All sites have the base class *Anime*. There are helper functions defined by :py:func: *anime_downloader.sites.helpers.request.get* and *anime_downloader.sites.helpers.request.post()*. You are expected to use these functions to perform any network requests. There is also a helper function anime_downloader.sites.helpers.request.soupfiy() for making *BeautifulSoup* s out of requests.

All requests are cached. So don't worry about making requests twice.

2.11 Base classes

anime.py contains the base classes required for other anime classes.

anime_downloader.sites.init.**get_anime_class**(*url*)

Get anime class corresposing to url or name. See `anime_downloader.sites.ALL_ANIME_SITES` to get the possible anime sites.

Parameters **url** (*string*) – URL of the anime.

Returns Concrete implementation of *anime_downloader.sites.anime.Anime*

Return type *anime_downloader.sites.anime.Anime*

anime_downloader.commands.dl.**sitenames = ['anitube', 'animtime', 'animebinge', 'animeflv',**

Built-in mutable sequence.

If no argument is given, the constructor creates a new empty list. The argument must be an iterable if specified.

class anime_downloader.sites.anime.**Anime**(*url=None*, *quality='720p'*, *fallback_qualities=None*, *_skip_online_data=False*, *subbed=None*)

Base class for all anime classes.

Parameters

- **url** (*string*) – URL of the anime.
- **quality** (*One of ['360p', '480p', '540p', '720p', '1080p']*) – Quality of episodes
- **fallback_qualities** (*list*) – The order of fallback.

sitename

name of the site

Type str

title

Title of the anime

Type str

meta

metadata about the anime. [Can be empty]

Type dict

QUALITIES

Possible qualities for the site

Type list

_scrape_metadata()

_scrape_metadata is function which has to be overridden by the base classes to scrape the metadata of anime from the web page.

Parameters **soup** (`bs4.BeautifulSoup`) – soup is the html of the anime url after passing through BeautifulSoup.

get_data()
get_data is called inside the __init__ of BaseAnime. It is used to get the necessary data about the anime and it's episodes.

This function calls _scrape_episodes and _scrape_metadata

TODO: Refactor this so that classes which need not be soupified don't have to overload this function.

Returns

A list of tuples of episodes containing episode name and episode url. Ex:

```
[('1', 'https://9anime.is/.../...', ...)]
```

Return type list

classmethod search(*query*)
Search searches for the anime using the query given.

Parameters **query** (*str*) – query is the query keyword to be searched.

Returns List of *SearchResult*

Return type list

class anime_downloader.sites.anime.**AnimeEpisode**(*url*, *parent: anime_downloader.sites.anime.Anime = None*, *ep_no=None*)
Base class for all Episode classes.

Parameters

- **url** (*string*) – URL of the episode.
- **quality** (*One of ['360p', '480p', '540p', '720p', '1080p']*) – Quality of episode
- **fallback_qualities** (*list*) – The order of fallback.

sitename
name of the site

Type str

title
Title of the anime

Type str

meta
metadata about the anime. [Can be empty]

Type dict

ep_no
Episode number/title of the episode

Type string

pretty_title
Pretty title of episode in format <animename>-<ep_no>

Type string

headers
Headers the downloader should use, used to bypass downloading restrictions.

Type dict

download(*force=False*, *path=None*, *format='{anime_title}_{ep_no}'*, *range_size=None*)
Downloads episode. This might be removed in a future release.

Parameters

- **force** (*bool*) – Whether to force download or not.
- **path** (*string*) – Path to the directory/file where the file should be downloaded to.
- **format** (*string*) – The format of the filename if not provided.

source(*index=0*)
Get the source for episode

Returns Extractor depending on the source.

Return type *anime_downloader.extractors.base_extractor.BaseExtractor*

class anime_downloader.sites.anime.**SearchResult**(*title*, *url*, *poster=''*, *meta=''*, *meta_info={}*)
SearchResult class holds the search result of a search done by an Anime class

Parameters

- **title** (*str*) – Title of the anime.
- **url** (*str*) – URL of the anime
- **poster** (*str*) – URL for the poster of the anime.
- **meta** (*dict*) – Additional metadata regarding the anime.

title
Title of the anime.

Type str

url
URL of the anime

Type str

poster
URL for the poster of the anime.

Type str

meta
Additional metadata regarding the anime.

Type dict

meta_info
Metadata regarding the anime. Not shown in the results, used to match with MAL

Type dict

2.12 Helper functions

anime_downloader.sites.helpers.request.**get**(*url: str*, *cf: bool = False*, *sel: bool = False*, *referer: str = None*, *cache: bool = True*, *headers=None*, ***kwargs*)
get performs a get request

Parameters

- **url** (*str*) – url is the url of the request to be performed
- **cf** (*bool*) – cf if True performs the request through cfscrape. For cloudflare protected sites.
- **sel** (*bool*) – sel if True perfroms the request through selescrape (selenium).
- **referer** (*str*) – a url sent as referer in request headers

anime_downloader.sites.helpers.request.**post**(*url: str*, *cf: bool = False*, *sel: bool = False*, *referer: str = None*, *cache: bool = True*, *headers=None*, ***kwargs*)

post performs a post request

Parameters

- **url** (*str*) – url is the url of the request to be performed
- **cf** (*bool*) – cf if True performs the request through cfscrape. For cloudflare protected sites.
- **sel** (*bool*) – sel if True perfroms the request through selescrape (selenium).
- **referer** (*str*) – a url sent as referer in request headers

anime_downloader.sites.helpers.request.**soupify**(*res*)

soupify Beautiful soups response object of request

Parameters **res** – res is *request.response*

Returns

Return type BeautifulSoup.Soup

Python Module Index

a

anime_downloader.sites.anime, 18
anime_downloader.sites.helpers.request, 20

Index

Symbols

_scrape_metadata() (*anime_downloader.sites.anime.Anime method*), 18

A

Anime (*class in anime_downloader.sites.anime*), 18
anime_downloader.sites.anime (*module*), 18
anime_downloader.sites.helpers.request (*module*), 20
AnimeEpisode (*class in anime_downloader.sites.anime*), 19
Animeflv (*class in anime_downloader.sites.animeflv*), 16
Anistream (*class in anime_downloader.sites.anistream*), 17

D

download() (*anime_downloader.sites.anime.AnimeEpisode method*), 20

E

ep_no (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19

G

get() (*in module anime_downloader.sites.helpers.request*), 20
get_anime_class() (*in module anime_downloader.sites.init*), 18
get_data() (*anime_downloader.sites.anime.Anime method*), 18

H

headers (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19

M

meta (*anime_downloader.sites.anime.Anime attribute*), 18
meta (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19
meta (*anime_downloader.sites.anime.SearchResult attribute*), 20
meta_info (*anime_downloader.sites.anime.SearchResult attribute*), 20

P

post() (*in module anime_downloader.sites.helpers.request*), 21
poster (*anime_downloader.sites.anime.SearchResult attribute*), 20
pretty_title (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19

Q

QUALITIES (*anime_downloader.sites.anime.Anime attribute*), 18

S

search() (*anime_downloader.sites.anime.Anime class method*), 19
SearchResult (*class in anime_downloader.sites.anime*), 20
sitename (*anime_downloader.sites.anime.Anime attribute*), 18
sitename (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19
sitenames (*in module anime_downloader.commands.dl*), 18
soupify() (*in module anime_downloader.sites.helpers.request*), 21
source() (*anime_downloader.sites.anime.AnimeEpisode method*), 20

T

title (*anime_downloader.sites.anime.Anime attribute*), 18

title (*anime_downloader.sites.anime.AnimeEpisode attribute*), 19
title (*anime_downloader.sites.anime.SearchResult attribute*), 20

U

url (*anime_downloader.sites.anime.SearchResult attribute*), 20