
anime-downloader Documentation

Release 3.5.0

Vishnunarayan K I

Oct 31, 2019

Contents:

1	Features	3
2	Supported Sites	5
2.1	Installation	5
2.2	anime dl	6
2.3	watch	7
2.4	Config	9
2.5	Site Configs	10
2.6	Library usage	10
2.7	Writing your own custom site class	10
2.8	Base classes	11
2.9	Helper functions	13
	Python Module Index	15
	Index	17

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.
- Add any anime to your watch list using *anime watch* and let anime downloader take care of everything for you.
- Download using external downloader ([aria2](https://aria2.github.io/) recommended).
- Configurable using *config.json*. See [doc](https://github.com/vn-ki/anime-downloader/wiki/Config).

CHAPTER 2

Supported Sites

- 9anime
- twist.moe
- KissAnime [cloudflare]
- Masterani.me [cloudflare]
- KissCartoon [cloudflare]
- Gogoanime
- AnimePahe [cloudflare]

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

2.1 Installation

The following are extended installation instructions for Windows and Linux users. (For windows choco users, scroll to last)

2.1.1 Windows via choco

Contributed by @CodaTheOtaku

Note: make sure you are running the Command Prompt in “Run as Administrator” mode

- 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/vn-ki/anime-downloader.git
```

- then the commands to view a show would be:

```
anime watch --provider *Insert provider name* --new
```

2.1.2 Mac

Anime downloader is available from brew.:

```
brew install anime-downloader
```

2.1.3 Linux

If you are using linux, you most probably already have python installed.

Type `pip --version` into your terminal. If it says python2, replace all the following `pip` with `pip3`.

- Install aria2.
- Install anime-downloader

```
pip3 install anime-downloader
```

- To install master branch:

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

- Enjoy.

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.

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

To search on kissanime,

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

Run `anime dl --help` for help text on `dl` subcommand.

2.2.2 Download directly

- To download Fullmetal Alchemist: Brotherhood all episodes

```
anime dl 'https://www7.9anime.is/watch/fullmetal-alchemist-brotherhood.0r7/j69y93'
```

- To download Fullmetal Alchemist: Brotherhood episode 1

```
anime dl 'https://www7.9anime.is/watch/fullmetal-alchemist-brotherhood.0r7/j69y93' --
↳episodes 1
```

- To download Fullmetal Alchemist: Brotherhood episode 1 to 20

```
anime dl 'https://www7.9anime.is/watch/fullmetal-alchemist-brotherhood.0r7/j69y93' --
↳episodes 1:21
```

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

```
anime dl 'https://www7.9anime.is/watch/fullmetal-alchemist-brotherhood.0r7/j69y93' --
↳url --episodes 1
```

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

```
anime dl 'https://www7.9anime.is/watch/fullmetal-alchemist-brotherhood.0r7/j69y93' --
↳play vlc --episodes 1
```

2.3 watch

Note: You need `mpv` to use this subcommand.

Fig. 1: anime watch in action

`anime watch` can be a all in one solution for your anime needs. Currently only supports 9anime.

`watch` will track your progress through an anime and thus make your life easier.

```
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.

(continues on next page)

(continued from previous page)

```

Available Commands after selection of an anime:
  set      : set episodes_done and title. Ex: set episodes_done=3
  remove  : remove selected anime from watch list
  update  : Update the episodes of the current anime
  watch   : Watch selected anime
  download : Download episodes of selected anime

Options:
-n, --new          Create a new anime to watch
-l, --list        List all animes in watch list
-r, --remove      Remove the specified anime
-q, --quality [360p|480p|720p|1080p]
                  Specify the quality of episode.
-ll, --log-level [DEBUG|INFO|WARNING|ERROR]
                  Sets the level of logger
--help           Show this message and exit.

```

2.3.1 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.

```

$ 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.

```

2.3.2 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. “ Available Commands after selection of an anime: set : set episodes_done and title. Ex: set episodes_done=3 remove : remove selected anime fro

2.4 Config

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

This can be found under:

- `~/.config/anime-downloader` on Linux
- `C:\Users\[Username]\AppData\Roaming\anime downloader` on Windows

You can override the settings in the `config.json`, with command line arguments.

The default config file is given below.

```
{
  "dl": {
    "download_dir": ".",
    "external_downloader": "{aria2}",
    "fallback_qualities": [
      "720p",
      "480p",
      "360p"
    ],
    "file_format": "{anime_title}/{anime_title}_{ep_no}",
    "force_download": false,
    "player": null,
    "provider": "9anime",
    "quality": "720p",
    "skip_download": false,
    "url": false
  },
  "siteconfig": {
    "animeflv": {
      "server": "natsuki",
      "version": "subbed"
    },
    "anistream.xyz": {
      "version": "subbed"
    },
    "nineanime": {
      "server": "mp4upload"
    }
  },
  "watch": {
    "log_level": "INFO",
    "provider": "9anime",
    "quality": "720p"
  }
}
```

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.

2.5 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)
```

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)
```

Site: <http://anistream.xyz>

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

```
anime_downloader.sites.nineanime.NineAnime
    alias  of  anime_downloader.sites.helpers.util.not_working.<locals>.wrapper.
    <locals>.NotWorking
```

2.6 Library usage

Anime downloader can be used as a library too.

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 list of *anime_downloader.sites.anime.SearchResult*.

2.7 Writing your own custom site class

anime_downloader is built with easy extensibility in mind.

Each of the site (in the tool) can roughly be classified into two.

- 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 *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.

2.8 Base classes

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

anime_downloader.sites.init.get_anime_class(url)

Get anime class corresponding 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 = ['9anime', 'gogoanime', 'kissanime', 'kisscartoon']
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*)

Base class for all anime classes.

Parameters

- **url** (*string*) – URL of the anime.
- **quality** (*One of ['360p', '480p', '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', '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

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=""*)

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

2.9 Helper functions

anime_downloader.sites.helpers.request.get (*url: str, cf: bool = True, referer: str = None, 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 cfscape. For cloudflare protected sites.
- **referer** (*str*) – a url sent as referer in request headers

`anime_downloader.sites.helpers.request.post` (*url: str, cf: bool = True, referer: str = None, 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 cfscape. For cloudflare protected sites.
- **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

a

`anime_downloader.sites.anime`, 11
`anime_downloader.sites.helpers.request`,
13

Symbols

`_scrape_metadata()`

(*anime_downloader.sites.anime.Anime* method), 11

A

Anime (class in *anime_downloader.sites.anime*), 11

anime_downloader.sites.anime (module), 11

anime_downloader.sites.helpers.request (module), 13

AnimeEpisode (class in *anime_downloader.sites.anime*), 12

Animeflv (class in *anime_downloader.sites.animeflv*), 10

Anistream (class in *anime_downloader.sites.anistream*), 10

D

`download()` (*anime_downloader.sites.anime.AnimeEpisode* method), 13

E

`ep_no` (*anime_downloader.sites.anime.AnimeEpisode* attribute), 12

G

`get()` (in module *anime_downloader.sites.helpers.request*), 13

`get_anime_class()` (in module *anime_downloader.sites.init*), 11

`get_data()` (*anime_downloader.sites.anime.Anime* method), 12

M

`meta` (*anime_downloader.sites.anime.Anime* attribute), 11

`meta` (*anime_downloader.sites.anime.AnimeEpisode* attribute), 12

`meta` (*anime_downloader.sites.anime.SearchResult* attribute), 13

N

NineAnime (in module *anime_downloader.sites.nineanime*), 10

P

`post()` (in module *anime_downloader.sites.helpers.request*), 14

`poster` (*anime_downloader.sites.anime.SearchResult* attribute), 13

`pretty_title` (*anime_downloader.sites.anime.AnimeEpisode* attribute), 12

Q

QUALITIES (*anime_downloader.sites.anime.Anime* attribute), 11

S

`search()` (*anime_downloader.sites.anime.Anime* class method), 12

SearchResult (class in *anime_downloader.sites.anime*), 13

`sitename` (*anime_downloader.sites.anime.Anime* attribute), 11

`sitename` (*anime_downloader.sites.anime.AnimeEpisode* attribute), 12

`sitenames` (in module *anime_downloader.commands.dl*), 11

`soupify()` (in module *anime_downloader.sites.helpers.request*), 14

`source()` (*anime_downloader.sites.anime.AnimeEpisode* method), 13

T

`title` (*anime_downloader.sites.anime.Anime* attribute), 11

title (*anime_downloader.sites.anime.AnimeEpisode*
attribute), 12

title (*anime_downloader.sites.anime.SearchResult* at-
tribute), 13

U

url (*anime_downloader.sites.anime.SearchResult*
attribute), 13