
Aniping Documentation

Release 1

kuroshi

Feb 05, 2018

1	Quick Start Guide	3
1.1	Installation	3
1.2	Configuration	4
2	Tutorial	5
2.1	Logging In and Out	5
2.2	Searching for shows	5
2.3	Highlighting shows	5
2.4	Selecting a show	6
2.5	Editing a show	6
2.6	Deleting a show	6
3	Plugin Development	7
3.1	Pluggable Framework	7
3.2	Backend	7
3.3	Search	7
3.4	Scraper	8
3.5	DB	8
4	Indices and tables	9

[Aniping](#) is a front-end for [Sonarr](#) which collects the current season's airing shows from [Anilist](#) and makes it easy to select and download whatever shows you want to see.

Aniping is self-hosted, so you don't have to store your data in a random server somewhere, unless you want to. It doesn't even store a user account, instead it uses Sonarr's account system to handle users, with no special configuration required.

It uses the power of indexers to determine who is subtitling a show for you, to make it easy to decide which group to watch. You can even highlight shows if you don't want to start watching them just yet, to make them easier to find later.

If you just want to get going, head over to the [Quick Start Guide](#) to get up and running.

1.1 Installation

Right now, the fastest way to get going is using [Docker](#). If you're unfamiliar with Docker or don't want to set it up, head over to the [Tutorial](#) for instructions on setting up using a virtual environment.

There are plans to set up the repository to be able to deploy in [Heroku](#), but they aren't yet implemented.

To deploy using Docker, make sure you docker downloaded and installed according to the [Docker installation instructions](#). If you have a more complicated setup, using something like Kubernetes, Docker Swarm, or Mesos, great! Setting those up is outside the scope of this guide, unfortunately, but you should be able to figure out the settings you need based on this guide.

The run command is pretty straightforward:

```
docker run -d -v /etc/localtime:/etc/localtime:ro -v /etc/timezone:/etc/timezone:ro -  
→v $LOCAL_CONFIG:/app/config -p 80:80 kuroshi/aniping
```

Let's step through it:

- `-d` runs the container in a daemonized mode, as a service.
- `-v /etc/localtime:/etc/localtime:ro` maps your local time into the container as a read-only volume.
- `-v /etc/timezone:/etc/timezone:ro` maps your timezone into the container as a read-only volume.
- `-v $LOCAL_CONFIG:/app/config` Replace `$LOCAL_CONFIG` with the directory you will store the `config.yml` file, and that will be mapped in appropriately.
- `-p 80:80` maps port 80 in the container to port 80 outside the container.
- `kuroshi/aniping` is the container you're downloading from the dockerhub.

1.2 Configuration

Now that you've got the docker container running, you'll need to make a config file. In your `$LOCAL_CONFIG` directory, place a file called `config.yml` with the following:

```
SONARR:
  URL: https://my.sonarr.url
  API_KEY: my-sonarr-api-key
  QUALITY_PROFILE: HD - 720p/1080p
  LIBRARY_PATH: /path/to/sonarrs/library
DELETE_SHOWS: False
NYAA:
  FILTER: Trusted Only
  CATEGORY: Anime - English-translated
ANILIST:
  CLIENT_ID: anilist_client_id
  CLIENT_SECRET: anilist_client_secret
SQLITE:
  FILE: db/aniping.sqlite
  SCHEMA: schema.ddl
BACK_END: Sonarr
DATABASE: Sqlite
SEARCH:
  - Nyaa
SCRAPER:
  - Anilist
SECRET_KEY: this is a totally random string and can be whatever you want
IMAGE_CACHE: static/images/cache
```

Spacing is important! The file will not be read correctly if it is not indented, with spaces, like it is above. You will need to change the values above to suit your needs. Defaults you don't need to change but you may want to change include `BACK_END`, `DATABASE`, `SEARCH`, `SCRAPER`, `SQLITE`-> `FILE`, `SONARR`-> `QUALITY_PROFILE`, and both options under `NYAA`. You absolutely should not change `SQLITE`-> `SCHEMA` unless you are using a different database, nor should you change `IMAGE_CACHE` unless you really know what you're doing. Everything else you will need to change.

Once you have Aniping set up as in the [Quick Start Guide](#), or using a more advanced setup, using it is pretty simple.

Before you do anything else, open up aniping in your web browser. If you followed the [Quick Start Guide](#) exactly, you should be able to open `http://localhost` on the same machine as your running container. Otherwise, you'll need to get your running container's IP address, or assign it a hostname. Once you are on the aniping home, you'll see that it is trying to populate its database. This can take 5 to 10 minutes, so just sit back and let it do its thing.

It will refresh when it has data to show, but it might not be completely done at that point. After 10 minutes or so, refresh the page and you should see all of this season's shows.

2.1 Logging In and Out

At the top right, next to the search box, is a simple log in and out form. This may already say “log out” if you have authentication disabled on sonarr. If you do not and it says “log in”, simply log in with the same username and password you use for sonarr. If you were successful, it will replace the link with a “log out” button. When you press it, it will instantly log you out.

2.2 Searching for shows

You can search for shows currently airing by using the search box. The search box is not very advanced, and cannot correct typos or misspellings, but it does search titles, studios, descriptions, and genres. It will not search sonarr, it will only search what it has in its database.

2.3 Highlighting shows

On desktop, when you are logged in, hover over a card and click the “star” button. The background of that card will turn yellow. Click it again to unhighlight it. This can be used to mark shows you want to watch, but you are waiting for whatever reason to start it.

On mobile, the star will always be visible.

2.4 Selecting a show

To select a show, hover over a card and click the “+” button. This should take you to a page with a more detailed description of the show, this time coming directly from sonarr’s search. If sonarr cannot find the show, you will be redirected back to the home page.

Once on the show description page, simply select which release group you would like working on the show, and click “Add show”. This will automatically create a tag with a restriction in sonarr if necessary, and add the show to sonarr’s list.

On mobile, the “+” will always be visible.

2.5 Editing a show

Any show in the “watching” category will have an edit button instead of a “+” button when rolled over. Click this button and you will be given the same interface as the show selection button. Make the appropriate change and click “Edit show”, and the show will be updated.

On mobile, the “edit” button will always be visible.

2.6 Deleting a show

If you no longer wish to have sonarr tracking a show, you can delete it by clicking the “x” button. It will not delete anything sonarr has already gathered.

3.1 Pluggable Framework

Aniping is written as a pluggable framework. There are four major plugin categories: `back_end`, `search`, `scraper`, and `db`. Each one performs a different task, and aniping is basically the glue that puts them together. You can extend aniping pretty easily just by building one of these plugins.

This page currently just has a high-level explanation of what these plugins are, but building them is fairly easy. Most of the documentation is handled in the plugins module in this guide, and you can use the existing plugins as templates, though some are more complicated than they perhaps should be.

There are three things to do when developing plugins for aniping:

- Put them in the correct directory.
- Make sure they extend the correct class described in the plugins module.
- Ensure they override the functions defined in the extended class.

That's all you need to do - the functions in the extended class will cause an exception if they aren't overridden.

3.2 Backend

In the default install, the backend for aniping is `sonarr`. The backend is the plugin that handles downloading or watching of shows. These shows should not be licensed in your country, of course.

3.3 Search

In the default install, the search engine for aniping is `Nyaa`. The search engine is the plugin that handles finding the release groups for the show - other plugins may use it to find specific results, however.

3.4 Scraper

In the default install, the scraper for aniping is anilist. The scraper is a bit of a misnomer, it's actually the repository of information for specific shows airing this season. Another popular example is MyAnimeList.

3.5 DB

In the default install, the database for aniping is sqlite3. The database is where the information from the scraper is held, as well as session ids, watching information, and other details. It is a necessary component of aniping.

CHAPTER 4

Indices and tables

- API reference