playlistfromsong Documentation

Release 2.1.1

Zack

Jul 06, 2017

Contents

Contents:

CHAPTER 1

playlistfromsong

Generate an offline playlist from a single song.

Features

- Similar song matching using last.fm or Spotify
- Automatic downloading of songs
- Builtin music server for webhooks

Quickstart

Install with `pip`:

```
pip install playlistfromsong
```

Download a playlist of 5 songs similar to Miles Davis' *Blue In Green*:

```
playlistfromsong --song 'Miles Davis Blue In Green' --num 5 -f /path/to/save
```

Use a bearer token `--bearer` to use Spotify to find suggestions:

```
playlistfromsong --song 'Miles Davis Blue In Green' --num 5 -f /path/to/save -b
↪'BEARER'
```

For more complete usage, see the docs.

CHAPTER 2

Installation

Stable release

To install playlistfromsong, run this command in your terminal:

```
$ pip install playlistfromsong
```

This is the preferred method to install playlistfromsong, as it will always install the most recent stable release.

If you don’t have pip installed, this Python installation guide can guide you through the process.

From sources

The sources for playlistfromsong can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/schollz/playlistfromsong
```

Or download the tarball:

```
$ curl  -OL https://github.com/schollz/playlistfromsong/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

Generate a playlist from song

Download a playlist from a song by specifying the artist and the song:

```
playlistfromsong -s 'Miles Davis Blue In Green'
```

By default, three songs are downloaded (the original song plus 2 that are similar), but you can change this with `-n`:

```
playlistfromsong -s 'Miles Davis Blue In Green' -n 30
```

By default, the similar songs are found using last.fm, but you can choose to use Spotify instead, by providing a bearer token. Obtain a bearer token by going to https://developer.spotify.com/web-api/console/get-track/ and click "Get OAUTH_TOKEN". Then apply your token::

```
playlistfromsong -s 'Miles Davis Blue In Green' -n 30 -b 'TOKEN'
```

Finally, you can specify a specific place to store the files by using the `-f` flag:

```
playlistfromsong -s 'Miles Davis Blue In Green' -f /music
```

Simple music server

There is a built-in simple music server that you can use to play your music, but also includes an API for webhooks for automatically generating playlists from songs.

Star the server using:

```
playlistfromsong --serve -f /path/to/music
```

The default port is 5000, and you should be able to see your server at http://localhost:5000. You can also specify the port with `--port X`.

There are routes for directly downloading songs. For instance, to generate a playlist in the current folder, just open:

```
http://localhost:5000/download/10/Miles Davis Blue In Green
```

This is very effective for using with IFTTT to automatically download playlists based on songs that are liked on Youtube / Spotify.

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/schollz/playlistfromsong/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

playlistfromsong could always use more documentation, whether as part of the official playlistfromsong docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/schollz/playlistfromsong/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *playlistfromsong* for local development.

1. Fork the *playlistfromsong* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/playlistfromsong.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv playlistfromsong
   $ cd playlistfromsong/
   $ python setup.py develop
   ```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

   ```
   $ flake8 playlistfromsong tests
   $ python setup.py test or py.test
   $ tox
   ```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/schollz/playlistfromsong/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test tests.test_playlistfromsong
```

CHAPTER 5

Credits

Development Lead

- Zack <hypercube.platforms@gmail.com>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

2.0.0 (2017-07-04)

- New `--server` option for starting a music server

1.0.0 (2017-06-25)

- Move to cookiecutter for improved packaging and tests

0.21.0 (2017-03-29)

- First stable release

CHAPTER 7

Indices and tables

- genindex
- modindex
- search