Fip Music Documentation

Release 0.2.5

Matthieu Berjon

Feb 27, 2018

Contents

Contents:

FIP music

Read the current and next track played on FIP radio

WARNING: at this early stage of the development, don't even think about any backward compatibility.

- Free software: Apache Software License 2.0
- Documentation: https://fip.readthedocs.io.

1.1 Usage

```
# Get the information on the current song played
$ fip --current-song

# Get the information on the next song being played
$ fip --next-song

# Get the help
$ fip --help
```

For more information, you can refer to the documentation of the project.

Installation

2.1 Stable release

To install FIP music, run this command in your terminal:

```
$ pip install fip
```

This is the preferred method to install FIP music, 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.

2.2 From sources

The sources for FIP music can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/mattberjon/fip
```

Or download the tarball:

```
$ curl  -OL https://github.com/mattberjon/fip/tarball/master
```

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

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

Usage

To use FIP music in a project:

```
import fip
```

3.1 Available keys

This is subject to change without any warning from Fip developers.

Key	Comment
uuid	Unique ID of the song
stepId	
title	Title of the song
start	Beginning of the song (timestamp)
end	End of the song (timestamp)
fatherStep	
stationId	Number of the station (Fip=7)
embededId	(default=song)
depth	
diskJockey	(default=null)
authors	Authors of the song
titleSlug	Slug of the song
anneeEditionMusique	Edition year of the song
visual	Link to the album cover
titreAlbum	Album title
label	Label name
releaseId	Unique release ID
coverUuid	Unique cover ID

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/mattberjon/fip_music/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/mattberjon/fip_music/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 :)

4.2 Get Started!

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

1. Fork the *fip_music* repo on GitHub.

2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/fip_music.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 fip_music
   $ cd fip_music/
   $ 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 fip_music 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.

4.3 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/mattberjon/fip_music/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_fip_music
```

Credits

5.1 Development Lead

- Matthieu Berjon <matthieu@berjon.net>

5.2 Contributors

None yet. Why not be the first?