Nose-Watcher Documentation

Release 0.1.1

Chris Trotman

February 07, 2015

Contents

CHAPTER 1

Nose-Watcher

A nose plugin to watch for changes within the local directory.

- Free software: BSD license
- Documentation: http://nose-watcher.readthedocs.org.

Inspired by the nose-watch nose plugin.

Note: nose-watcher will only run on linux, due to the depenency on *python-inotify* and *inotify*.

1.1 Features

- Watches for changes in the local directory, then runs nosetests with the specified command line options.
- Doesn't run the tests multiple times if you're using vim, Unlike the similar plugin *nose-watch*.
- Specify additional filetypes to watch using the command line argument *–filetype*.

1.2 Installation

```
pip install nose-watcher
```

1.3 Usage

```
nosetests --with-watcher
```

```
nosetests --with-watcher --filetype .txt
```

1.4 Additional Contributors

- Felix Chapman (aelred)

CHAPTER 2

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:

2.1 Types of Contributions

2.1.1 Report Bugs

Report bugs at https://github.com/solarnz/nose-watcher/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.

2.1.2 Fix Bugs

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

2.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

2.1.4 Write Documentation

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

2.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/solarnz/nose-watcher/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 :)

2.2 Get Started!

Ready to contribute? Here's how to set up *nose-watcher* for local development.

1. Fork the *nose-watcher* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/nose-watcher.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 nose-watcher
   $ cd nose-watcher/
   $ pip install -r requirements-testing.txt
   $ 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 nose-watcher tests
   $ python setup.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.

2.3 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, and 3.3, and for 3.4. Check https://travis-ci.org/solarnz/nose-watcher/pull_requests and make sure that the tests pass for all supported Python versions.

2.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_nose-watcher
```

CHAPTER 3

Credits

3.1 Development Lead

- Chris Trotman <chris@trotman.io>

3.2 Contributors

None yet. Why not be the first?

3.3 History

3.4 0.1.2 (2014-06-09)

- Handle KeyboardInterrupts while watching for changes.

3.5 0.1.1 (2014-06-08)

- Improve documentation
- Improve unit tests

3.6 0.1.0 (2014-06-08)

- First release on PyPI.