
Python Statechart Documentation

Release 0.3.1

Leigh McKenzie

October 16, 2016

1	Python Statechart	3
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.2.0 (2016-08-02)	15
6.2	0.2.1 (2016-08-07)	15
6.3	0.2.2 (2016-08-08)	15
6.4	0.2.3 (2016-08-10)	15
6.5	0.2.4 (2016-08-21)	15
6.6	0.3.0 (2016-10-16)	15
6.7	0.3.1 (2016-10-16)	15
7	Indices and tables	17

Contents:

Python Statechart

Python UML statechart framework

- Free software: ISC license
- Documentation: <https://statechart.readthedocs.org>.

Installation

At the command line:

```
$ easy_install statechart
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv statechart  
$ pip install statechart
```

Usage

To use Python Statechart in a project:

```
import statechart
```

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/leghmck/statechart/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” is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

Python Statechart could always use more documentation, whether as part of the official Python Statechart 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/leghmck/statechart/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 *statechart* for local development.

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

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

```
$ flake8 statechart tests
$ py.test
```

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 3.3, 3.4 and 3.5.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_statechart
```

Credits

5.1 Development Lead

- Leigh McKenzie <maccarav0@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.2.0 (2016-08-02)

- First release on PyPI.

6.2 0.2.1 (2016-08-07)

- Final state bug fixes.

6.3 0.2.2 (2016-08-08)

- Default transition bug fix.

6.4 0.2.3 (2016-08-10)

- Consume event dispatched by child state unless a final state activated.

6.5 0.2.4 (2016-08-21)

- Fix internal transition acting like local transition.

6.6 0.3.0 (2016-10-16)

- Implement display module to generate Plant UML code of a statechart.
- Raise runtime exception if an action is defined on top level statechart.

6.7 0.3.1 (2016-10-16)

- Implement specific statechart deactivate function.

Indices and tables

- `genindex`
- `modindex`
- `search`