
myscipkg2 Documentation

Release unknown

Ryan Dwyer

January 15, 2015

1	Features	3
1.1	Contents:	3
1.2	Feedback	5

A package for science using numpy, matplotlib, readthedocs, etc.

- TODO

1.1 Contents:

1.1.1 Installation

At the command line via pip:

```
$ pip install myscipkg2
```

Or install the latest GitHub version using:

```
$ git clone https://github.com/{ cookiecutter.github_username }/myscipkg2.git
$ cd myscipkg2
$ python setup.py install
```

1.1.2 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:

Types of Contributions

Report Bugs

Report bugs at <https://github.com/ryanpdwyer/myscipkg2/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” is open to whoever wants to implement it.

Implement Features

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

Write Documentation

myscipkg2 could always use more documentation, whether as part of the official myscipkg2 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/ryanpdwyer/myscipkg2/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 *myscipkg2* for local development.

1. Fork the *myscipkg2* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/myscipkg2.git
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. When you’re done making changes, check that your changes pass tests by running:

```
$ python setup.py test
```

5. 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
```

6. 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 2.7. Check <https://travis-ci.org/ryanpdwyer/myscipkg2> under pull requests for active pull requests to check that all configurations pass.

Tips

To run a subset of tests:

```
$ py.test test/test_myscipkg2.py
```

1.1.3 Credits

Development Lead

- Ryan Dwyer <ryanpdwyer@gmail.com>

Contributors

None yet. Why not be the first?

1.2 Feedback

If you have any suggestions or questions about **myscipkg2** feel free to email me at ryanpdwyer@gmail.com.

If you encounter any errors or problems with **myscipkg2**, please let me know! Open an Issue at the GitHub <http://github.com/ryanpdwyer/myscipkg2> main repository.