complexity Documentation

Release 0.1

Audrey Roy

July 18, 2013

CONTENTS

Contents:

CHAPTER

ONE

COMPLEXITY

A refreshingly simple static site generator, for those who like to work in HTML.

Of course, @pydanny (https://twitter.com/pydanny) came up with the name for this.

1.1 Documentation

The full documentation is at http://complexity.rtfd.org.

1.2 Quickstart

Using Complexity is easy! Try it out:

```
$ pip install complexity
$ git clone git@github.com:audreyr/complexity-example.git my_project
$ cd my_project
$ complexity
```

Open a web browser to http://127.0.0.1:9090 to see your newly generated Complexity static site.

1.3 Features

- Takes simple HTML templates as input.
- Template inheritance, filters, etc. (Brought to you by Jinja2.)
- Data from .json files turns into template context data.

1.4 Best Used With

Complexity is designed to be used with these packages:

- *Simplicity*: Converts ReStructuredText into JSON, which Complexity can use as input.
- *A Lot of Effort*: Deploys a static website (e.g. the output of Complexity) to Amazon S3.
- *Cookiecutter*: Creates projects from project templates.

Sure, they could have all been built into Complexity, but decoupling them seemed like a nice thing to do.

1.5 Dependencies

- Jinja2

CHAPTER

TWO

USAGE

2.1 Setup

Create this directory structure for your site:

```
my_project/
    input/
    output/
```

In *input/*, create your templates.

Put static files into the *css/*, *js/*, and *img/* directories of *output/*. (Creating additional directories like *ico/* is fine.)

2.2 To Generate HTML and Serve It Locally

Run the *complexity* command:

```
$ complexity
```

Open a web browser to http://127.0.0.1:9090. You should see your newly generated site!

2.3 To Upload Your Site to Amazon S3

TODO.

CHAPTER

THREE

CONTRIBUTING

Contributions are welcome!

3.1 Submitting Feedback

The best way to send feedback is to file an issue at https://github.com/audreyr/complexity/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.

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 :)

3.2 Getting Started

Here's how to set up *design* for local development.

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

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

3.3 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+ and 3.3+. Check https://travis-ci.org/audreyr/complexity/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a particular test:

```
$ python -m unittest tests.test_complexity.TestComplexity.test_make_sure_path_exists
```

To run a subset of tests:

```
$ python -m unittest tests.test_complexity
```

CHAPTER

FOUR

CREDITS

4.1 Development Lead

- Audrey Roy <audreyr@gmail.com>

4.2 Contributors

- Daniel Greenfeld (@pydanny) <3

CHAPTER

FIVE

HISTORY

5.1 0.3 (2013-07-18)

- Graceful shutdown/restart of dev server.
- Required input and output dir arguments.
- Optional port argument.
- Improved server start/stop messages.
- Major internal refactor.

5.2 0.2.1 (2013-07-15)

- Fixes to setup.py.

5.3 0.2.0 (2013-07-15)

- Data from .json files now gets read as template context data.
- Tested (and passing!) on Python 2.6, 2.7, 3.3, PyPy.

5.4 0.1.1 (2013-07-10)

- First release on PyPI.

CHAPTER

SIX

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*