Gingerit Documentation

Release 0.5.8

Tim Kleinschmidt

October 18, 2015

Contents

1 Gingerit **3**
- 1.1 Installation: . . . 3
- 1.2 Usage: . . . 3
- 1.3 TODO: . . . 3
- 1.4 Contributing: . . . 3
- 1.5 Thanks . . . 4

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**

7 0.5.5 (2015-01-11) **17**

8 Indices and tables **19**

Contents:

CHAPTER 1

Gingerit

Correcting spelling and grammar mistakes based on the context of complete sentences. Wrapper around the gingersoftware.com API

- Free software: ISC license
- Documentation: https://gingerit.readthedocs.org.

1.1 Installation:

```
pip install gingerit
```

1.2 Usage:

```
from gingerit.gingerit import GingerIt

text = 'The smelt of fliwers bring back memories.'

parser = GingerIt()
parser.parse(text)
```

1.3 TODO:

- More tests
- Commandline Tool

1.4 Contributing:

1. Fork it
2. Create your feature branch (*git checkout -b my-new-feature*)
3. Commit your changes (*git commit -am 'Add some feature'*)
4. Push to the branch (*git push origin my-new-feature*)

5. Create new Pull Request

1.5 Thanks

Thank you for [Ginger Proofreader](http://www.gingersoftware.com/) for such awesome service. Hope they will keep it free :)

Thanks to @subosito for this inspriration https://github.com/subosito/gingerice (Ruby Gem)

CHAPTER 2

Installation

At the command line:

```
$ easy_install gingerit
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv gingerit
$ pip install gingerit
```

CHAPTER 3

Usage

To use Gingerit in a project:

```
import gingerit
```

CHAPTER 4

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/azd325/gingerit/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

Gingerit could always use more documentation, whether as part of the official Gingerit 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/azd325/gingerit/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 *gingerit* for local development.

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

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

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, and 3.4, and for PyPy. Check https://travis-ci.org/azd325/gingerit/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

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

CHAPTER 5

Credits

5.1 Development Lead

- Tim Kleinschmidt <tim.kleinschmidt@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

CHAPTER 7

0.5.5 (2015-01-11)

- Updates requirements and python package

CHAPTER 8

Indices and tables

- genindex
- modindex
- search