
Git2JSON Documentation

Release 0.2.3

Tavish Armstrong

February 13, 2014

1	Git2JSON	3
1.1	Installation	3
1.2	Usage	3
1.3	Example JSON	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
5	Credits	11
5.1	Development Lead	11
5.2	Contributors	11
6	History	13
6.1	0.1.0 (2013-09-23)	13
6.2	0.1.1 (2013-09-25)	13
6.3	0.2.0 (2013-09-28)	13
7	Indices and tables	15

Contents:

Convert git logs to JSON for easier analysis.

- Free software: BSD license
- Documentation: <http://git2json.rtfid.org>.

1.1 Installation

The easiest way to install `git2json` is through `pip`:

```
pip install git2json
```

If you want the most up-to-date version (at a greater risk of encountering bugs) you can clone this repository and install it manually:

```
git clone https://github.com/tarmstrong/git2json.git
cd git2json
python setup.py install
```

1.2 Usage

```
usage: git2json [-h] [--git-dir GIT_DIR]
```

optional arguments:

```
-h, --help            show this help message and exit
--git-dir GIT_DIR    Path to the .git/ directory of the repository you are
                    targeting
```

The resulting JSON log is printed to standard output.

1.3 Example JSON

The following shows the structure of the JSON emitted by the tool.

```
[{
  "committer": {
    "date": 1379903278,
```

```
    "timezone": "-0400",
    "name": "Tavish Armstrong",
    "email": "tavisharmstrong@gmail.com"
  },
  "parents": ["e307663594031738c932877c8589552d5aafc953"],
  "author": {
    "date": 1379903278,
    "timezone": "-0400",
    "name": "Tavish Armstrong",
    "email": "tavisharmstrong@gmail.com"
  },
  "tree": "bd03127651335e3a51241f507f3bf194d8336485",
  "commit": "d06454c160218b4a782afad2429abdaladd54df0",
  "message": "Allow user to specify git-dir on the command line.",
  "changes": [
    [23, 3, "git2json/__init__.py"]
  ]
},
// ... More commits
]
```

1.3.1 Usage Examples

- [How Long Does It Take To Review an IPython Pull Request?](<http://nbviewer.ipython.org/urls/raw.github.com/tarmstrong/code-analysis/master/IPythonReviewTime.ipynb>) by Tavish Armstrong (git2json author)

If you are using `git2json` in interesting ways, please share and I will add your notebooks/blog posts/code to this list.

Installation

At the command line:

```
$ easy_install git2json
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv git2json  
$ pip install git2json
```

Usage

To use Git2JSON in a project:

```
import git2json
```

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/tarmstrong/git2json/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

Git2JSON could always use more documentation, whether as part of the official Git2JSON 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/tarmstrong/git2json/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 *git2json* for local development.

1. Fork the *git2json* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/git2json.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 git2json
$ cd git2json/
$ 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 git2json tests
$ python setup.py nosetests
```

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 and 2.7 (in the future 3.x and PyPy will be supported). Check https://travis-ci.org/tarmstrong/git2json/pull_requests and make sure that the tests pass for all supported Python versions.

Credits

5.1 Development Lead

- Tavish Armstrong <tavisharmstrong@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2013-09-23)

- First release on PyPI.

6.2 0.1.1 (2013-09-25)

- Fixed broken PyPI package.

6.3 0.2.0 (2013-09-28)

- Parents now an array instead of a single value.

6.4 0.2.1 (2013-10-07)

- Re-implemented parser with regular expressions.
- Character encoding issues fixed.

6.5 0.2.2 (2013-11-09)

- Fix bug #7 (hidden filenames not parsed correctly): <https://github.com/tarmstrong/git2json/issues/7>

6.6 0.2.3 (2013-11-09)

- Python 3 compatibility

Indices and tables

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