# Contributing

Contributions are \*\*welcome\*\* and will be fully \*\*credited\*\*.

We accept contributions via pull requests via

[Github](https://github.com/cviebrock/eloquent-sluggable).

1. Fork the project.

2. Create your bugfix/feature branch and write your (well-commented) code.

3. Create unit tests for your code:

- Run `composer install --dev` in the root directory to install required testing packages.

- Add your test classes/methods to the `/tests/` directory.

- Run `vendor/bin/phpunit` and make sure everything passes (new and old tests).

3. Commit your changes (and your tests) and push to your branch.

4. Create a new pull request against this package's `master` branch.

## Pull Requests

- \*\*Use the [PSR-2 Coding Standard](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md).\*\*

The easiest way to apply the conventions is to install [PHP Code Sniffer](http://pear.php.net/package/PHP\_CodeSniffer).

- \*\*Add tests!\*\* Your pull request won't be accepted if it doesn't have tests.

- \*\*Document any change in behaviour.\*\* Make sure the `README.md` and any other relevant

documentation are kept up-to-date.

- \*\*Consider our release cycle.\*\* We try to follow [SemVer v2.0.0](http://semver.org/).

Randomly breaking public APIs is not an option.

- \*\*Create feature branches.\*\* Don't ask us to pull from your master branch.

- \*\*One pull request per feature.\*\* If you want to do more than one thing, send multiple pull requests.

- \*\*Send coherent history.\*\* - Make sure each individual commit in your pull request is meaningful.

If you had to make multiple intermediate commits while developing, please

[squash them](http://www.git-scm.com/book/en/v2/Git-Tools-Rewriting-History#Changing-Multiple-Commit-Messages)

before submitting.

- Don't worry about updating `CHANGELOG.md` or `.semver`. The package administrator

will handle updating those when new releases are created.

\*\*Thank you!\*\*