# Contributing

First of all, \*\*thank you\*\* for contributing, \*\*you are awesome\*\*!

Everybody should be able to help. Here's how you can do it:

1. [Fork it](https://github.com/jolicode/JoliNotif/fork\_select)

2. improve it

3. submit a [pull request](https://help.github.com/articles/creating-a-pull-request)

Here's some tips to make you the best contributor ever:

\* [Rules](#rules)

\* [Green tests](#green-tests)

\* [Standard code](#standard-code)

\* [Keeping your fork up-to-date](#keeping-your-fork-up-to-date)

## Rules

Here are a few rules to follow in order to ease code reviews, and discussions

before maintainers accept and merge your work.

\* You MUST follow the [PSR-1](http://www.php-fig.org/psr/1/) and

[PSR-2](http://www.php-fig.org/psr/2/) (see [Rules](#rules)).

\* You MUST run the test suite (see [Green tests](#green-tests)).

\* You MUST write (or update) unit tests.

\* You SHOULD write documentation.

Please, write [commit messages that make

sense](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html),

and [rebase your branch](http://git-scm.com/book/en/Git-Branching-Rebasing)

before submitting your Pull Request (see also how to [keep your

fork up-to-date](#keeping-your-fork-up-to-date)).

One may ask you to [squash your

commits](http://gitready.com/advanced/2009/02/10/squashing-commits-with-rebase.html)

too. This is used to "clean" your Pull Request before merging it (we don't want

commits such as `fix tests`, `fix 2`, `fix 3`, etc.).

Also, while creating your Pull Request on GitHub, you MUST write a description

which gives the context and/or explains why you are creating it.

Your work will then be reviewed as soon as possible (suggestions about some

changes, improvements or alternatives may be given).

## Green tests

Run the tests using the following script:

```shell

make test

```

## Standard code

Use [PHP CS fixer](http://cs.sensiolabs.org/) to make your code compliant with

JoliNotif's coding standards:

```shell

make cs

```

## Keeping your fork up-to-date

To keep your fork up-to-date, you should track the upstream (original) one

using the following command:

```shell

git remote add upstream https://github.com/jolicode/JoliNotif.git

```

Then get the upstream changes:

```shell

git checkout master

git pull --rebase origin master

git pull --rebase upstream master

git checkout <your-branch>

git rebase master

```

Finally, publish your changes:

```shell

git push -f origin <your-branch>

```

Your pull request will be automatically updated.

Thank you!