## Project License: Eclipse Public License v1.0

- You will only Submit Contributions where You have authored 100% of the content.

- You will only Submit Contributions to which You have the necessary rights. This means that if You are employed You have received the necessary permissions from Your employer to make the Contributions.

- Whatever content You Contribute will be provided under the Project License(s).

## Coding Conventions

### Formatting

See [CODING\_STYLE.txt](CODING\_STYLE.txt) for how we format our code.

## How to submit a pull request

We love pull requests. Here is a quick guide:

1. You need to have Maven and a JDK (at least version 1.5) installed.

2. [Fork the repo](https://help.github.com/articles/fork-a-repo).

3. [Create a new branch](https://help.github.com/articles/creating-and-deleting-branches-within-your-repository/) from `main`.

4. Ensure that you have a clean state by running `./mvnw verify`.

5. Add your change together with a test (tests are not needed for refactorings and documentation changes).

6. Format your code: Import the JUnit project in Eclipse and use its formatter or apply the rules in the `CODING\_STYLE` file manually. Only format the code you've changed; reformatting unrelated code makes it harder for us to review your changes.

7. Run `./mvnw verify` again and ensure all tests are passing.

8. Push to your fork and [submit a pull request](https://help.github.com/articles/creating-a-pull-request/).

Now you are waiting on us. We review your pull request and at least leave some comments.

Note that if you are thinking of providing a fix for one of the bugs or feature requests, it's usually

a good idea to add a comment to the bug to make sure that there's agreement on how we should proceed.

## Limitations

The JUnit team is not accepting changes to the code under the following paths:

\* `src/main/java/junit`

\* `test/java/junit/tests/framework`

\* `test/java/junit/tests/extensions`

The reasoning is that the JUnit team feels that our users should focus on using either the JUnit4 or JUnit5 APIs.

The team is also reluctant to accept changes that only update code from one code style to another.

Generally the code in JUnit was approved by at least one person, so two people agreed that the style was reasonable.

To find other places where you can have an impact, please see the [Issues tagged "up-for-grabs"](https://github.com/junit-team/junit4/issues?q=is%3Aissue+is%3Aopen+label%3Aup-for-grabs).