# Contributing to react-boilerplate

Love react-boilerplate and want to help? Thanks so much, there's something to do for everybody!

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, they should reciprocate that respect in addressing your issue or assessing patches and features.

## Using the issue tracker

The [issue tracker](https://github.com/react-boilerplate/react-boilerplate/issues) is

the preferred channel for [bug reports](#bugs), [features requests](#features)

and [submitting pull requests](#pull-requests).

<a name="bugs"></a>

## Bug reports

A bug is a \_demonstrable problem\_ that is caused by the code in the repository.

Good bug reports are extremely helpful - thank you!

Guidelines for bug reports:

1. \*\*Use the GitHub issue search\*\* &mdash; check if the issue has already been reported.

2. \*\*Check if the issue has been fixed\*\* &mdash; try to reproduce it using the latest `master` or development branch in the repository.

3. \*\*Isolate the problem\*\* &mdash; ideally create a [reduced test case](https://css-tricks.com/reduced-test-cases/) and a live example.

4. \*\*Use the bug report template\*\* &mdash; please fill in the template which appears when you open a new issue.

A good bug report shouldn't leave others needing to chase you up for more information. Please try to be as detailed as possible in your report. What is your environment? What steps will reproduce the issue? What browser(s) and OS

experience the problem? What would you expect to be the outcome? All these details will help people to fix any potential bugs.

Example:

> ## Description

> A clear and concise description of what the bug is.

>

> Any other information you want to share that is relevant to the issue being

> reported. This might include the lines of code that you have identified as

> causing the bug, and potential solutions (and your opinions on their

> merits).

>

> ## Steps to reproduce

> Steps to reproduce the behavior:

>

> 1. This is the first step

> 2. This is the second step

> 3. Further steps, etc.

>

> (Add link to a demo on https://jsfiddle.net or similar if possible)

>

> \*\*Expected behavior\*\*

> A clear and concise description of what you expected to happen.

>

> \*\*Screenshots\*\*

> If applicable, add screenshots to help explain your problem.

>

> ## Versions

>

> - React-Boilerplate:

> - Node/NPM:

> - Browser:

<a name="features"></a>

## Feature requests

Feature requests are welcome. But take a moment to find out whether your idea fits with the scope and aims of the project. It's up to \_you\_ to make a strong case to convince the project's developers of the merits of this feature. Please provide as many details and as much context as possible.

There is also a template for feature requests. Please make sure to use it.

<a name="pull-requests"></a>

## Pull requests

Good pull requests - patches, improvements, new features - are a fantastic

help. They should remain focused in scope and avoid containing unrelated

commits.

\*\*Please ask first\*\* before embarking on any significant pull request (e.g.

implementing features, refactoring code, porting to a different language),

otherwise you risk spending a lot of time working on something that the

project's developers might not want to merge into the project.

Please adhere to the coding conventions used throughout a project (indentation,

accurate comments, etc.) and any other requirements (such as test coverage).

Since the `master` branch is what people actually use in production, we have a

`dev` branch that unstable changes get merged into first. Only when we

consider that stable we merge it into the `master` branch and release the

changes for real.

Adhering to the following process is the best way to get your work

included in the project:

1. [Fork](https://help.github.com/articles/fork-a-repo/) the project, clone your fork, and configure the remotes:

```bash

# Clone your fork of the repo into the current directory

git clone https://github.com/<your-username>/react-boilerplate.git

# Navigate to the newly cloned directory

cd react-boilerplate

# Assign the original repo to a remote called "upstream"

git remote add upstream https://github.com/react-boilerplate/react-boilerplate.git

```

2. If you cloned a while ago, get the latest changes from upstream:

```bash

git checkout dev

git pull upstream dev

```

3. Create a new topic branch (off the `dev` branch) to contain your feature, change, or fix:

```bash

git checkout -b <topic-branch-name>

```

4. Commit your changes in logical chunks. Please adhere to these [git commit message guidelines](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html) or your code is unlikely be merged into the main project. Use Git's [interactive rebase](https://help.github.com/articles/about-git-rebase/) feature to tidy up your commits before making them public.

5. Locally merge (or rebase) the upstream dev branch into your topic branch:

```bash

git pull [--rebase] upstream dev

```

6. Push your topic branch up to your fork:

```bash

git push origin <topic-branch-name>

```

7. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/)

with a clear title and description.

\*\*IMPORTANT\*\*: By submitting a patch, you agree to allow the project

owners to license your work under the terms of the [MIT License](https://github.com/react-boilerplate/react-boilerplate/blob/master/LICENSE.md).

# Collaborating guidelines

You can find the list of all contributors in [README.md](./README.md).

There are few basic rules to ensure high quality of the boilerplate:

- Before merging, a PR requires at least two approvals from the collaborators unless it's an architectural change, a large feature, etc. If it is, then at least 50% of the core team have to agree to merge it, with every team member having a full veto right. (i.e. every single one can block any PR)

- A PR should remain open for at least two days before merging (does not apply for trivial contributions like fixing a typo). This way everyone has enough time to look into it.

You are always welcome to discuss and propose improvements to this guideline.

# Add yourself as a contributor

This project follows the [All Contributors specification](https://allcontributors.org/). To add yourself to the table of contributors in the README file, please use the [bot](https://allcontributors.org/docs/en/bot/overview) or the [CLI](https://allcontributors.org/docs/en/cli/overview) as part of your PR.

If you've already added yourself to the list and are making a new type of contribution, you can run it again and select the new contribution type.