# Contributing to Sharetribe

First of all, thank you so much! We greatly appreciate all contributions. However, before you spend time working on Sharetribe, please read over our guidelines:

## Reporting issues

Use the [ISSUE\_TEMPLATE.md](https://raw.githubusercontent.com/sharetribe/sharetribe/master/ISSUE\_TEMPLATE.md) to report issues. Here's an example issue: [raw](https://raw.githubusercontent.com/sharetribe/sharetribe/master/ISSUE\_EXAMPLE.md) / [formatted](https://github.com/sharetribe/sharetribe/blob/master/ISSUE\_EXAMPLE.md)

If you'd like to know what happens after you've reported an Issue, see [How to handle Github Issues](docs/how-to-handle-github-issues.md)

## Bug fixes ?

1. Fix the issue in a branch of your own fork of [Sharetribe](https://github.com/sharetribe/sharetribe).

1. Rebase your branch to include the latest changes from `master`. We do not accept pull requests with merge commits.

1. Open a pull request from your fork to `master`.

1. Make sure all the [tests pass](https://github.com/sharetribe/sharetribe#running-tests).

1. Update [CHANGELOG.md](CHANGELOG.md)

## Refactoring only Pull Requests ?

[As the smart people at Discourse say it:](https://meta.discourse.org/t/discourse-development-contribution-guidelines/3823)

> It is often tempting to submit PRs that improve Code Climate score or amend internal logic to make it more readable.

> Though we strive to have well factored code that is easy to reason about we can not afford risking regressions on a production product without immediate tangible gain.

> If you wish to improve an area of the code, fix a bug in that area and also improve the code.

Of course, there might be exceptions to the rule, e.g. removing some unused code which is clearly used nowhere anymore.

## New features are welcome ONLY IF they have been greenlighted by the Sharetribe team ??

By default, we don't accept pull requests that introduce new features if it hasn’t been greenlighted by the Sharetribe team in advance.

Join the [Sharetribe Community Forum](https://www.sharetribe.com/community/) and tell us what you are about to do. The team might be able to offer some suggestions on how to proceed. Features that don’t fit with Sharetribe’s vision and roadmap will not be accepted. Talking with the team is the best way to find out whether your feature is in line with Sharetribe’s plans.

From a technical point of view, the Sharetribe Ruby on Rails app is not always built using the traditional "Rails way". In addition to the MVC layers, additional API and Store layers are used to define self-contained boundaries inside the application. All new features have to follow this structure.

After you've received a go-ahead from the team:

1. Start coding in a branch of your own fork of Sharetribe.

1. Open a pull request from your fork to master as early as possible. Pull requests are a [great way to start a conversation around a feature](https://github.com/blog/1124-how-we-use-pull-requests-to-build-github). If the pull request is work in progress and not yet ready to be reviewed, add a \[WIP\] prefix to the title.

1. Make tests for your new feature. It’s the best way to guarantee that other developers don't accidentally break your feature.

1. Rebase your branch to include the latest changes from `master`. We do not accept pull requests with merge commits.

1. Make sure all the [tests pass](https://github.com/sharetribe/sharetribe#running-tests).

1. Update [CHANGELOG.md](CHANGELOG.md)

1. Remove the \[WIP\] prefix and ping the core team to review the pull request.