# Contributing

Pull requests are very welcome, but should be within the scope of the project, and follow the repository's code conventions. Before submitting a pull request, it's always good to file an issue, so we can discuss the details of the PR.

## Reporting a Bug

1. Ensure you've replicated the issue against master. There is a chance the issue may have already been fixed.

2. Search for any similar issues (both opened and closed). There is a chance someone may have reported it already.

3. Provide a demo of the bug isolated in a jsfiddle/jsbin. Sometimes this is not a possibility, in which case provide a detailed description along with any code snippets that would help in triaging the issue. If we cannot reproduce it, we will close it.

4. The best way to demonstrate a bug is to build a failing test. This is not required, however, it will generally speed up the development process.

## Submitting a pull request

1. [Fork][fork] the repository.

1. Ensure that all tests are passing prior to submitting.

1. If you are adding new functionality, or fixing a bug, provide test coverage.

1. Follow syntax guidelines detailed below.

1. Push the changes to your fork and submit a pull request. If this resolves any issues, please mark in the body `resolve #ID` within the body of your pull request. This allows for github to automatically close the related issue once the pull request is merged.

1. Last step, [submit the pull request][pr]!

[pr]: https://github.com/formatjs/formatjs/compare/

[fork]: https://github.com/formatjs/formatjs/fork/

## Releasing a new version

The following the process to release a new version of the `react-intl` package on npm. This repo uses a protected `master` branch so the process involves creating a Pull Request for the version bump:

1. Make sure local `node\_modules` is up to date: `rm -rf node\_modules && npm install`.

2. Create a release branch from `master`: `git checkout -b release`

3. Bump version using `npm version` and choose appropriate `patch`, `minor`, `major` argument.

4. Create a Pull Request for your local `release` branch so Travis CI tests run.

5. If all the tests pass successfully, publish your local `release` branch to npm: `npm publish`.

6. Push the Git tag to the main fork: `git push upstream --tags`.

7. Merge the `release` branch PR into `master` \*\*and make sure to create a merge commit\*\* so the Git tag matches.

8. Create a [release](https://github.com/formatjs/formatjs/releases) post for the new release Git tag.