# Contributing

Hi there! We're thrilled that you'd like to contribute to this project. Your help is essential for keeping it great.

Please note that this project is released with a [Contributor Code of Conduct][code-of-conduct]. By participating in this project you agree to abide by its terms.

# ? Probot v8 beta

Heads up: we are currently working on version 8. If you submit a pull request, please select the `beta` branch as your base.

If you need a fix in v7, please submit the fix to the `beta` branch first (if applicable). Then create another pull request against the `v7.x` branch.

## Submitting a pull request

1. [Fork][fork] and clone the repository

1. Configure and install the dependencies: `npm install`

1. Make sure the tests pass on your machine: `npm test`, note: these tests also apply the [linter][linter] and run the TypeScript compiler (`tsc`) to check for type errors, so there's no need to run these commands separately.

1. Create a new branch: `git checkout -b my-branch-name`

1. Make your change, add tests, and make sure the tests still pass

1. Push to your fork and [submit a pull request][pr]

1. Pat your self on the back and wait for your pull request to be reviewed and merged.

Here are a few things you can do that will increase the likelihood of your pull request being accepted:

- Follow the [style guide][style] which is using standard. Any linting errors should be shown when running `npm test`

- Write and update tests.

- Keep your change as focused as possible. If there are multiple changes you would like to make that are not dependent upon each other, consider submitting them as separate pull requests.

- Write a [good commit message](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html).

Work in Progress pull requests are also welcome to get feedback early on, or if there is something that blocked you.

To submit a pull request for a previous version of Probot, checkout the maintenance branch before creating a new branch, e.g. `git checkout 7.x && git checkout -b my-branch-name`. When submitting the pull request, set the base branch to the same maintenance branch.

## Merging the Pull Request & releasing a new version

Releases are automated using [semantic-release](https://github.com/semantic-release/semantic-release).

The following three commit message conventions determine which version is released:

1. `fix: ...` or `fix(scope name): ...` prefix in subject: bumps fix version, e.g. `1.2.3` ? `1.2.4`

2. `feat: ...` or `feat(scope name): ...` prefix in subject: bumps feature version, e.g. `1.2.3` ? `1.3.0`

3. `BREAKING CHANGE: ` in body: bumps breaking version, e.g. `1.2.3` ? `2.0.0`

Only one version number is bumped at a time, the highest version change trumps the others. Besides publishing a new version to npm, semantic-release also creates a git tag and release on GitHub, generates changelogs from the commit messages and puts them into the release notes.

New versions are released from the following branches

1. `master`: new versions are published to npm using the `@latest` dist-tag

1. `next`: new versions are published to npm using the `@next` dist-tag. When merging the `next` branch into `master`, the `@latest` dist-tag will be bumped automatically.

1. `beta`: a new `@beta` version will be released. For example, if the current version is `7.5.0`, then the first version will be published as `8.0.0-beta` using the `@beta` dist-tag. The next version will be `8.0.0-beta.1`, etc. When merged into master, the `8.0.0` version will be released automatically and all release notes will be combined into the `8.0.0` release. See also [Publishing pre-releases](https://github.com/semantic-release/semantic-release/blob/beta/docs/recipes/pre-releases.md).

1. `<version>.x`: For example `7.x`. New bug fix & feature releases are published the given version. No breaking changes are allowed. See also [Publishing maintenance releases](https://github.com/semantic-release/semantic-release/blob/beta/docs/recipes/maintenance-releases.md).

When in doubt, ping [@gr2m](https://github.com/gr2m).

## Just starting out? Looking for how to help?

Use [this search][good-first-issue-search] to find Probot apps that have issues marked with the `good-first-issue` label.

## Resources

- [How to Contribute to Open Source](https://opensource.guide/how-to-contribute/)

- [Using Pull Requests](https://help.github.com/articles/about-pull-requests/)

- [GitHub Help](https://help.github.com)

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

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

[style]: https://standardjs.com/

[code-of-conduct]: CODE\_OF\_CONDUCT.md

[good-first-issue-search]: https://github.com/search?utf8=%E2%9C%93&q=topic%3Aprobot+topic%3Aprobot-app+good-first-issues%3A%3E0&type=

[linter]: https://github.com/probot/probot/blob/ts-readme/tslint.json