# Contributing

## Raising Issues

Feel free to raise an issue if you have a question, an enhancement, or a bug report.

Use the issue search functionality to search all \*\*\_opened and closed\_\*\* issues before raising a new issue. If you raise a duplicate issue, you're just creating noise for everyone watching this repo.

Before raising a bug, ensure you are using the latest version of our packages. We release every week, so there's a good chance your issue might have already been fixed.

Finally, when raising a new issue, please fill out the issue template - \*\*\_please don't skip sections\_\*\*.

Please provide \*\*\_as much information as possible\_\*\*. This project is maintained by volunteers, so the more the more information you provide, the less likely we will have to waste everyone's time in asking you for more information.

If you have a particularly complex issue - consider creating a small, self-contained reproduction repo. This will help you in figuring out the exact problem, and will help us in reproducing and diagnosing the bug.

\*\*\_Help us to help you\_\*\*

## Commenting

Feel free to comment on any open issue if you have more information that you feel like you can provide. If you don't have more information, instead use the "reaction" feature on the root comment for the issue. We use reactions to help gauge which issues are important to the community, so these are the best way to show us an issue is important.

Please refrain from leaving useless comments on issues. Comments like "+1", or "when's this getting fixed", or "any progress on this" just serve as spam, and annoy every single person subscribed to the issue. Generally we will just delete those comments, so save everyone time and think twice.

Please refrain from commenting on old, closed issues and PRs. Your issue is rarely related enough to a closed issue to warrant "necroing" a dead thread - raising a new issue means you can fill in the template, and make it easier for us to help you. Often times if you comment on a closed issue, we will just ask you to open a new issue, so please save everyone's time, and \*\*\_help us to help you\_\*\*.

Please refrain from commenting on `master` commits. Commit comments are not searchable, meaning that nobody else can discover your comments. Raise an issue and reference the commit instead so that everyone can see your comment, and you can fill out the template.

---

## Pull Requests

Anyone is free to help us build and maintain this project. If you see an issue that needs working on because it's important to you, comment on the issue to help make sure that nobody else works on it at the same time, and then start working.

Developing in this repo is easy:

- First fork the repo, and then clone it locally.

- Create a new branch.

- In the root of the project, run `yarn install`.

- This will install the dependencies, link the packages and do a build.

- Make the required changes.

### Validating Your Changes

We have a sophisticated CI process setup which gets run on every PR. You must pass all of the checks for us to consider merging your PR. Here is a list of checks that are done automatically, that you can also perform locally before pushing.

- Ensure your code is properly formatted.

- You can run `yarn format` in any package or in the root.

- Alternatively, you can run prettier on save.

- Ensure there are no typechecking errors.

- You can run `yarn typecheck` in any package or in the root.

- Ensure your changes are adequately tested.

- You can run `yarn test` in any package.

- We aim for around `90%` branch coverage for every PR.

- Coverage reports should automatically be generated locally, and the `codecov` bot should also comment on your PR with the percentage, as well as links to the line-by-line coverage of each file touched by your PR.

- Ensure you have no lint errors.

- You can run `yarn lint` in any package or in the root.

- You can run `yarn lint:markdown` in the root.

- If you have made changes to any markdown documentation, ensure there are no spelling errors

- You can run `yarn check:spelling` in the root.

- Or if you are using vscode, you can use [`Code Spell Checker`](https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker) plugin.

- If you have made changes within the `eslint-plugin` package, ensure the configs and documentation are valid.

- You can run `yarn check:configs` and `yarn check:docs` in the root, or in the `eslint-plugin` folder.

### Raising a PR

Once your changes are ready, you can raise a PR. The title of your PR should match the following format:

```text

<tag>(<package>): <short description>

```

Where `<tag>` is one of:

- `feat` - for any new functionality additions

- `fix` - for any bug fixes that don't add new functionality

- `test` - if you only change tests, and not shipped code

- `docs` - if you only change documentation, and not shipped code

- `chore` - anything else

And `<package>` is the name of the package you have made changes within (`eslint-plugin`, `parser`, `typescript-estree`, etc). If you make significant changes across multiple packages, you can omit this (i.e. `feat: foo bar`).

And `<short description>` is a succinct title for the PR.

Within the body of your PR, make sure you reference the issue that you have worked on, as well as pointing out anything of note you wish us to look at during our review.

Make sure you use the "Fixes #xxx" format to reference issues, so that GitHub automatically closes the issues when we merge the PR. Also note that if you are fixing multiple issues at once, you can only reference one issue per line, and must put one "Fixes #xxx" per issue number.

In terms of your commit history - we do not care about the number, or style of commits in your history, because we squash merge every PR into master. Feel free to commit in whatever style you feel comfortable with.

\*\*\_One thing we ask is to please avoid force pushing after you have raised a PR\_\*\*. GitHub is not able to track changes across force pushes, which makes it impossible to efficiently do incremental reviews. This slows us down, and means it will take longer for us to get your PR merged.

### Addressing Feedback and Beyond

With your PR raised, and the CI showing green, your PR will [sit in the queue to be reviewed](https://github.com/typescript-eslint/typescript-eslint/pulls?q=is%3Apr+is%3Aopen+sort%3Acreated-asc+-label%3A%22breaking+change%22+-label%3A%22awaiting+response%22+-label%3A%221+approval%22+-label%3A%22DO+NOT+MERGE%22). We generally review PRs oldest to newest, unless we consider a newer PR higher priority (i.e. if it's a bug fix).

Please note that as this project is maintained by volunteers, it may take a while for us to get around to your PR (sometimes a month or more). Be patient, we'll get to it. Please refrain from commenting asking for a review, or similar bump comments. \*\*\_These just create spam for maintainers, and does not push your PR higher in the queue\_\*\*.

Once we have reviewed your PR, we will provide any feedback that needs addressing. If you feel a requested change is wrong, don't be afraid to discuss with us in the comments. Once the feedback is addressed, and the PR is reviewed, we'll ensure the branch is up to date with master, and merge it for you.