# Contributing Guide

Hi! I’m really excited that you are interested in contributing to Rax. Before submitting your contribution though, please make sure to take a moment and read through the following guidelines.

- [Issue Reporting Guidelines](#issue-reporting-guidelines)

- [Pull Request Guidelines](#pull-request-guidelines)

- [Git Commit Specific](./GIT\_COMMIT\_SPECIFIC.md)

## Development Workflow

After cloning rax, run `npm install` to fetch its dependencies.

Run `npm run setup` link and bootstrap project before development.

Then, you can run several commands:

\* `npm run lint` checks the code style.

\* `npm test` runs the complete test suite.

\* `npm test -- --watch` runs an interactive test watcher.

\* `npm test <pattern>` runs tests with matching filenames.

\* `npm run build` creates `lib` and `dist` folder with all the packages.

## Issue Reporting Guidelines

- The issue list of this repo is \*\*exclusively\*\* for bug reports and feature requests. Non-conforming issues will be closed immediately.

- For simple beginner questions, you can get quick answers from

- For more complicated questions, you can use Google or StackOverflow. Make sure to provide enough information when asking your questions - this makes it easier for others to help you!

- Try to search for your issue, it may have already been answered or even fixed in the development branch.

- Check if the issue is reproducible with the latest stable version of Rax. If you are using a pre-release, please indicate the specific version you are using.

- It is \*\*required\*\* that you clearly describe the steps necessary to reproduce the issue you are running into. Issues with no clear repro steps will not be triaged. If an issue labeled "need repro" receives no further input from the issue author for more than 5 days, it will be closed.

- For bugs that involves build setups, you can create a reproduction repository with steps in the README.

- If your issue is resolved but still open, don’t hesitate to close it. In case you found a solution by yourself, it could be helpful to explain how you fixed it.

## Pull Request Guidelines

- Only code that's ready for release should be committed to the master branch. All development should be done in dedicated branches.

- Checkout a \*\*new\*\* topic branch from master branch, and merge back against master branch.

- Work in the `src` folder and \*\*DO NOT\*\* checkin `dist` in the commits.

- Make sure `npm test` passes.

- If adding new feature:

- Add accompanying test case.

- Provide convincing reason to add this feature. Ideally you should open a suggestion issue first and have it greenlighted before working on it.

- If fixing a bug:

- If you are resolving a special issue, add `(fix #xxxx[,#xxx])` (#xxxx is the issue id) in your PR title for a better release log, e.g. `update entities encoding/decoding (fix #3899)`.

- Provide detailed description of the bug in the PR. Live demo preferred.

- Add appropriate test coverage if applicable.

## Git Commit Specific

- Your commits message must follow our [git commit specific](./GIT\_COMMIT\_SPECIFIC.md).

- We will check your commit message, if it does not conform to the specification, the commit will be automatically refused, make sure you have read the specification above.

- You could use `git cz` with a CLI interface to replace `git commit` command, it will help you to build a proper commit-message, see [commitizen](https://github.com/commitizen/cz-cli).

- It's OK to have multiple small commits as you work on your branch - we will let GitHub automatically squash it before merging.