# Contributing to React Native Paper

## Code of Conduct

We want this community to be friendly and respectful to each other. Please read [the full text](https://callstack.com/code-of-conduct/?utm\_source=github.com&utm\_medium=referral&utm\_campaign=react-native-paper&utm\_term=code-of-conduct) so that you can understand what actions will and will not be tolerated.

## Our Development Process

The core team works directly on GitHub and all work is public.

### Development workflow

> \*\*Working on your first pull request?\*\* You can learn how from this \*free\* series: [How to Contribute to an Open Source Project on GitHub](https://egghead.io/series/how-to-contribute-to-an-open-source-project-on-github).

1. Fork the repo and create your branch from `master` (a guide on [how to fork a repository](https://help.github.com/articles/fork-a-repo/)).

2. Run `yarn bootstrap` to setup the development environment.

3. Do the changes you want and test them out in the example app before sending a pull request.

### Commit message convention

We follow the [conventional commits specification](https://www.conventionalcommits.org/en) for our commit messages:

\* `fix`: bug fixes, e.g. fix Button color on DarkTheme.

\* `feat`: new features, e.g. add Snackbar component.

\* `refactor`: code refactor, e.g. new folder structure for components.

\* `docs`: changes into documentation, e.g. add usage example for Button.

\* `test`: adding or updating tests, eg unit, snapshot testing.

\* `chore`: tooling changes, e.g. change circleci config.

\* `BREAKING CHANGE`: for changes that break existing usage, e.g. change API of a component.

Our pre-commit hooks verify that your commit message matches this format when committing.

### Linting and tests

We use `typescript` for type checking, `eslint` with `prettier` for linting and formatting the code, and `jest` for testing. Our pre-commit hooks verify that the linter and tests pass when commiting. You can also run the following commands manually:

\* `yarn typescript`: type-check files with `tsc`.

\* `yarn lint`: lint files with `eslint` and `prettier`.

\* `yarn test`: run unit tests with `jest`.

### Sending a pull request

When you're sending a pull request:

\* Prefer small pull requests focused on one change.

\* Verify that `typescript`, `eslint` and all tests are passing.

\* Preview the documentation to make sure it looks good.

\* Follow the pull request template when opening a pull request.

When you're working on a component:

\* Follow the guidelines described in the [official material design docs](https://material.io/guidelines/).

\* Write a brief description of every prop when defining `type Props` to aid with documentation.

\* Provide an example usage for the component (check other components to get a idea).

\* Update the type definitions for Flow and Typescript if you changed an API or added a component.

### Running the example

The example app uses [Expo](https://expo.io/) for the React Native example. You will need to install the Expo app for [Android](https://play.google.com/store/apps/details?id=host.exp.exponent) and [iOS](https://itunes.apple.com/app/apple-store/id982107779) to start developing.

After you're done, you can run `yarn example start` in the project root (or `expo start` in the `example/` folder) and scan the QR code to launch it on your device.

To run the example on web, run `yarn example web` in the project root.

### Working on documentation

The documentation is automatically generated from the [TypeScript](https://www.typescriptlang.org/) annotations in the components. You can add comments above the type annotations to add descriptions. To preview the generated documentation, run `yarn docs start` in the project root.

### Publishing a release

We use [release-it](https://github.com/webpro/release-it) to automate our release. If you have publish access to the NPM package, run the following from the master branch to publish a new release:

```sh

yarn release

```

NOTE: You must have a `GITHUB\_TOKEN` environment variable available. You can create a GitHub access token with the "repo" access [here](https://github.com/settings/tokens).

## Reporting issues

You can report issues on our [bug tracker](https://github.com/callstack/react-native-paper/issues). Please follow the issue template when opening an issue.

## License

By contributing to React Native Paper, you agree that your contributions will be licensed under its \*\*MIT\*\* license.