# Contributing

Thanks for taking the time to checkout this repo and be willing to contribute!

If you have found an issue or would like to request a new feature, simply create a new issue. New issues come with a issue template, so make sure to fill out as much information as possible.

We also welcome pull requests, bug fixes and improved test coverage as contributions!

If this is your first open source contribution, please take a look at [this](https://egghead.io/courses/how-to-contribute-to-an-open-source-project-on-github) guide.

## Reporting Bugs & Feature Requests

If you would like to submit a feature request or report a bug, we encourage you to first look through the [issues](https://github.com/react-native-elements/react-native-elements/issues) and [pull requests](https://github.com/react-native-elements/react-native-elements/pulls) before filing a new issue.

Check out the [example/](https://github.com/react-native-elements/react-native-elements/tree/master/example) folder to see how React Native Elements is used to build the [React Native Elements App](https://expo.io/@monte9/react-native-elements-app) on Expo.

## Submitting a Pull Request

If you wish to submit a pull request for a new feature or issue, you should start by forking this repository first. This should get you setup on your local machine:

### Setup

- Install [Node.js](https://nodejs.org/) if you have not already. (\_We suggest you to use node v6.x.x\_)

- Fork the \*\*react-native-elements\*\* repo [here](https://github.com/react-native-elements/react-native-elements)

- `git clone <your-forked-repo-url> && cd react-native-elements`

- `yarn install` OR `npm install`

- `npm test`

Now create a new branch with a name that loosely describes the issue on which you will be working. Once you think you have addressed the issue in question, submit a pull request to the `next` branch.

### Committing and Pushing Changes

We like to provide informative and useful commit messages when pushing changes to the repo. This helps tremendously when sifting through the commit history to find a particular changeset. A useful guide for creating meaningful commit messages can be found [here.](https://github.com/conventional-changelog-archived-repos/conventional-changelog-angular/blob/ed32559941719a130bb0327f886d6a32a8cbc2ba/convention.md)

### Branching Strategy

There are 3 main branches:

- `master` is the branch with the most recent deployed version

- `next` is the main development branch. For new features and enhancements, base the

changes off this branch.

- `patch` is a branch for working on patch releases. If you have a bug fix

that requires a fast release, use this branch.