# Contributing to React Native Svg

Thank you for helping out with react-native-svg!

We'd like to make contributions as pleasant as possible, so here's a small guide of how we see it. Happy to hear your feedback about anything, so please let us know.

### Modifying react-native-svg

1. Fork this repository

2. Clone your fork

3. Make a branch for your feature or bug fix (i.e. `git checkout -b added-getfoobar`)

4. Work your magic

5. Execute `yarn link` when done.

### Testing your changes

You can test your changes on any React Native application you have set up locally.

You can also use the testing application available at https://github.com/msand/react-native-svg-e2e/

or: https://github.com/magicismight/react-native-svg-example

Just `cd` to your application and type `yarn link react-native-svg` to make your app use your local modified package instead of the one from npmjs.com (this is what point 5 was about in the previous section).

If you made changes in the native code, don't forget to run `react-native link` before building your React Native application so that latest changes are taken into account.

## Tests

We use `typescript` for type checks, `eslint` with `prettier` for linting/formatting, `jest/detox` for tests (unit and e2e). All tests are run on travis-ci for all opened pull requests, but you should use them locally when making changes.

- `yarn test`: Run all tests, except for e2e (see note below).

- `yarn lint`: Run `eslint` check.

- `yarn tsc`: Run `typescript` check.

- `yarn flow`: Run `flow` type check.

- `yarn jest`: Run `jest` type check.

Currently e2e tests exist here: https://github.com/msand/react-native-svg-e2e/

## Sending a pull request

When you're sending a pull request:

- Communication is a key. If you want fix/add something, please open new/find existing issue, so we can discuss it.

- We prefer small pull requests focused on one change, as those are easier to test/check.

- Please make sure that all tests are passing on your local machine.

- Follow the template when opening a PR.

## Commits and versioning

All PRs are merged into the `develop` branch, following [conventional commit message](https://www.conventionalcommits.org/en/v1.0.0-beta.3). Combined with [semantic versioning](https://semver.org/), this allows us to have a frequent releases of the library.

\_Note\_: We don't force this convention on Pull Requests from contributors, but it's a clean way to see what type of changes are made, so feel free to follow it.

Most notably prefixes you'll see:

- \*\*fix\*\*: Bug fixes, triggers \_patch\_ release

- \*\*feat\*\*: New feature implemented, triggers \_minor\_

- \*\*chore\*\*: Changes that are not affecting end user (CI config changes, scripts, ["grunt work"](https://stackoverflow.com/a/26944812/3510245))

- \*\*docs\*\*: Documentation changes.

- \*\*perf\*\*: A code change that improves performance.

- \*\*refactor\*\*: A code change that neither fixes a bug nor adds a feature.

- \*\*test\*\*: Adding missing tests or correcting existing tests.

## Release process

We use [Semantic Release](http://semantic-release.org) to release new versions of the library when changes are merged into the `master` branch, which we plan to keep stable. Bug fixes take priority in the release order. The master branch should always contain the latest released code.

## Reporting issues

You can report issues on our [bug tracker](https://github.com/react-native-community/react-native-svg/issues). Please search for existing issues and follow the issue template when opening an one. Except no need to add any notes to the changelog as semtice released handles that automatically based on the commit messages.

## License

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