# Contributing to React Native CLI

## Development Process

All work on React Native CLI happens directly on GitHub. Contributors send pull requests which go through review process.

> \*\*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/)).

1. Run `yarn` or `npm install` to install all required dependencies.

1. Run `yarn watch` to automatically build the changed files.

1. Now you are ready to do the changes.

## Repository

Repository is splitted into two packages:

- `cli` - Historically, it was included in `react-native` package. Contains all the commands code.

- `global-cli` - Historically, it was a `react-native-cli` package and the only reason this package existed was to initialize an empty project.

## Testing your changes

> Please make sure the version of React Native matches the one present in devDependencies of the CLI. Otherwise, you may get unexpected errors.

Because of a modular design of the CLI, we recommend developing using symbolic links to its packages. This way you can use it seamlessly in the tested project, as you'd use the locally installed CLI. Here's what you need to run in the terminal:

```sh

cd /path/to/cloned/cli/

yarn link-packages

```

And then:

```sh

cd /my/new/react-native/project/

yarn link "@react-native-community/cli-platform-ios" "@react-native-community/cli-platform-android" "@react-native-community/cli" "@react-native-community/cli-server-api" "@react-native-community/cli-types" "@react-native-community/cli-tools" "@react-native-community/cli-debugger-ui"

npx react-native start --watchFolders /path/to/cloned/cli/

npx react-native run-android

```

\_Note: you must use the `--watchFolders` flag with the `start` command when testing the CLI with `yarn link` like this. Otherwise Metro can't find the symlinked folder and this may result in errors such as `ReferenceError: SHA-1 for file ... is not computed`.\_

Once you're done with testing and you'd like to get back to regular setup, run `yarn unlink` instead of `yarn link` from above command. Then `yarn install --force`.

## Testing `init` command

You can test your changes by installing local npm proxy - `verdaccio`, and publishing custom versions of `@react-native-community/cli` and `react-native`.

- Install `verdaccio`

```sh

yarn global add verdaccio

```

- Run verdaccio

```sh

verdaccio

```

- Set npm registry to `verdaccio` proxy server

```sh

npm set registry http://localhost:4873/

```

- Clone `react-native` and `@react-native-community/cli`

- Release new version of `@react-native-community/cli` to local npm proxy. If you have any issues, head over to [verdaccio](https://github.com/verdaccio/verdaccio) and check out the docs.

```

cd /path/to/cli/packages/cli && npm publish

```

- Install new version of `@react-native-community/cli` in `react-native` and publish new version of it.

```sh

# RN\_CLI\_VERSION is the version of localy released cli

cd /path/to/react-native && yarn add @react-native-community/cli@${RN\_CLI\_VERSION} && npm publish

```

- You are ready to go

```sh

# RN\_VERSION is the version of localy released react-native

react-native init --version ${RN\_VERSION}

```

- Cleanup

```sh

npm config set registry https://registry.npmjs.org/

```

## Running `start` command

In order for linked dependencies to work correctly when running `start` locally, set `--watchFolders` with a path to the root folder of the CLI project:

```

node path/to/cli/packages/cli/build/bin.js start --watchFolders path/to/cli

```

## Running CLI with React Native from the source

First make sure you have RN repo checked out and CLI repo checked out and built. Then you can start a new RN project with local version of CLI and RN without publishing or proxy:

1. Check out `react-native` repo. Then update template in local `react-native/template/package.json`, replacing dependency version of `react-native` with the absolute path of the react native repo, for example: "react-native": "file:///Users/username/react-native" (you can find the absolute path using `pwd` command)

1. Go back up and create a new RN project: `node ./cli/packages/cli/build/index.js init --template=file:///path/to/local/react-native RNTestProject`

1. To work with android, update gradle config in the newly created project following the second part of [Christoph's instructions](https://gist.github.com/cpojer/38a91f90614f35769e88410e3a387b48)

1. Run start (as described above) and compile your app eg `node ../cli/packages/cli/build/index.js run-android` (make sure you definitely have NDK r17c installed before building android)

## Typechecking, linting and testing

Currently we use TypeScript for typechecking, `eslint` with `prettier` for linting and formatting the code and `jest` for testing.

- `yarn lint`: run `eslint` and `prettier`

- `yarn test`: run unit tests

## Commit message convention

We prefix our commit messages with one of the following to signify the kind of change:

- \*\*build\*\*: Changes that affect the build system or external dependencies

- \*\*ci\*\*, \*\*chore\*\*: Changes to our CI configuration files and scripts

- \*\*docs\*\*: Documentation only changes

- \*\*feat\*\*: A new feature

- \*\*fix\*\*: A bug fix

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

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

- \*\*style\*\*: Changes that do not affect the meaning of the code

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

## 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.

## Reporting issues

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

## License

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