# Contributing to Ignite CLI

[![js-standard-style](https://img.shields.io/badge/code%20style-standard-brightgreen.svg?style=flat)](http://standardjs.com/) [![Build Status](https://semaphoreci.com/api/v1/ir/ignite/branches/master/shields\_badge.svg)](https://semaphoreci.com/ir/react\_native\_base)

We welcome all contributors to Ignite CLI! This contributing guide will help you get up and running to submit your first pull request.

Before submitting a pull request, you will want to make sure that your branch meets the following requirements:

\_These examples assume you're using yarn rather than npm. If you're not using yarn, replace these commands with the [appropriate npm alternative](https://shift.infinite.red/npm-vs-yarn-cheat-sheet-8755b092e5cc)\_

- Everything works on iOS/Android

- AVA Tests pass in the root folder (`yarn test`)

- Integration tests pass (`yarn run integration`)

- New tests are included for any new functionality

- Code is compliant with StandardJS (`yarn lint`)

- Branch has already been [synced with the upstream repo](https://help.github.com/articles/syncing-a-fork/) and any merge-conflicts have been resolved.

## Requirements

- Node 7.6+

- NPM 4 (ships with Node 7)

## Getting Started

1. Fork and then clone the repo (`git clone git@github.com:<YOURGITHUBUSER>/ignite.git`)

2. CD into the directory (`cd ignite`)

3. Uninstall npm version (`yarn global remove ignite-cli && yarn global remove react-native-ignite`)

4. Pull all package dependencies (`yarn`)

5. Link the local binary (`yarn link`)

Test it out:

```sh

$ ignite --version

<current version here>

$ which ignite

/usr/local/bin/ignite

$ ignite new UberForHeadLice

...

```

Now you're ready to check out a new branch and get hacking on Ignite CLI!

## Source Code

To get familiarized with Ignite CLI's source code, read the [Tour of Ignite CLI's source code](../docs/advanced-guides/tour.md).

## How to Build and Run App

```sh

$ cd ~/your/apps/directory

$ ignite new HackingOnIgnite

```

If you want to play with a new boilerplate, reference it locally:

```sh

$ ignite new HackingOnBoilerplate -b /full/path/to/boilerplate

```

## Testing the App

We use [AVA](https://github.com/avajs/ava) for testing.

To run tests from the ignite-base folder:

```sh

$ yarn test

```

Or to run it in [intelligent watch mode](https://github.com/avajs/ava/blob/master/docs/recipes/watch-mode.md):

```sh

$ yarn test -- --watch

```

You'll also want to run the integration tests before submitting a pull request:

```sh

$ yarn run integration

```

## :no\_entry\_sign: Standard Compliant

[![js-standard-style](https://cdn.rawgit.com/feross/standard/master/badge.svg)](https://github.com/feross/standard)

This project adheres to Standard. Our CI enforces this, so we suggest you enable linting to keep your project compliant during development.

\*\*To Run Lint\*\* from ignite-base:

```sh

$ yarn lint

```

\*\*To Lint on Commit\*\*

1. Install git-hooks => On a Mac `brew install git-hooks` - [Other](https://github.com/icefox/git-hooks/)

2. Setup on Repo => `git hooks --install`

\*\*Bypass Lint\*\*

If you have to bypass lint for a special commit that you will come back and clean (pushing something to a branch etc.) then you can bypass git hooks with adding `--no-verify` to your commit command.

\*\*Understanding Linting Errors\*\*

The linting rules are from JS Standard and React-Standard. [Regular JS errors can be found with descriptions here](http://eslint.org/docs/rules/), while [React errors and descriptions can be found here](https://github.com/yannickcr/eslint-plugin-react).