Thank you for your interest in contributing to React Native! From commenting on and triaging issues, to reviewing and sending PRs, [all contributions are welcome](#). In this document, we'll cover the steps to contributing code to React Native.

If you are eager to start contributing code right away, we have a list of [good first issues](#) that contain bugs which have a relatively limited scope. Issues labeled `help_wanted` are good issues to submit a PR for.

Prerequisites

- You have [Node](#) installed at v8.3.0+ and [Yarn](#) at v1.2.0+.
- You have the [React Native CLI](#) installed.
- You have [Watchman](#) installed.
- You are familiar with [Git](#).

> *You may refer to the [Getting Started](#) guide for installation instructions*

In addition to these, there may be additional dependencies for the platform you are building for:

Android:

- You have [Android Studio](#) installed at v3.2.0+.
- You have [Oracle JDK8](#) or [OpenJDK 8](#).

You can use Android Studio's SDK Manager to install the following dependencies.

- Android SDK Platform 28, Android 9.0 (Pie)
- Android SDK Build-Tools 28.0.3
- Android Support Repository 28+
- Android NDK r17c

These packages are available under the SDK Platforms and SDK Tools tabs. You may need to click on "Show Package Details" to see them. You can also use the following links to download the Android NDK r17c directly:

- [macOS (64-bit)](#)
- [Linux (64-bit)](#)
- [Windows (64-bit)](#)
- [Windows (32-bit)](#)

Environment Variables

Configure the `ANDROID_HOME` and `ANDROID_NDK` environment variables to point to your Android SDK and Android NDK install directories, respectively. You'll also need to update your `PATH` environment variable to include several Android utilities.

The following is an example configuration for a macOS machine:

```
# ~/.bash_profile
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/tools/bin
export PATH=$PATH:$ANDROID_HOME/platform-tools
export ANDROID_NDK=$HOME/Library/Android/android-ndk-r17c/
```

iOS:

- You have a Mac running macOS 10.14.3+.
- You have Xcode installed at v10.2.0+.
- You have the Xcode Command Line Tools installed.

> *The Command Line Tools can be installed through the Locations panel in Xcode's Preferences.*

Development Workflow

After cloning React Native, open the directory and run `yarn` to install its dependencies.

Then, you can run several commands:

- `yarn start` starts the Metro packager server.
- `yarn lint` checks the code style.
- `yarn format` automatically formats your code.
- `yarn test` runs the JavaScript test suite.
- `yarn test --watch` runs an interactive JavaScript test watcher.
- `yarn test <pattern>` runs JavaScript tests with matching filenames.
- `yarn flow` runs the [Flow](#) typechecks.
- `yarn flow-check-android` does a full Flow check over `*.android.js` files.
- `yarn flow-check-ios` does a full Flow check over `*.ios.js` files.
- `yarn test-ios` runs the iOS test suite (macOS required).
- `node ./scripts/run-ci-e2e-tests.js --js --android --ios` runs the end-to-tend JavaScript, Android, and iOS tests.

Testing your Changes

Tests help us prevent regressions from being introduced to the codebase. We recommend running `yarn test` or the platform-specific scripts above to make sure you don't introduce any regressions as you work on your change.

The GitHub repository is [continuously tested](#) using Circle and Appveyor, the results of which are available through the Checks functionality on [commits](#) and pull requests. You can learn more about running and writing tests in the [Tests wiki](#).

Coding Style

We use Prettier to format our JavaScript code. This saves you time and energy as you can let Prettier fix up any formatting issues automatically through its editor integrations, or by manually running `npm run prettier`. We also use a linter to catch styling issues that may exist in your code. You can check the status of your code styling by simply running `npm run lint`.

However, there are still some styles that the linter cannot pick up, notably in Java or Objective-C code.

Objective-C:

- Space after `@property` declarations
- Brackets on *every* `if`, on the *same* line
- `- method`, `@interface`, and `@implementation` brackets on the following line
- *Try* to keep it around 80 characters line length (sometimes it's just not possible...)
- `*` operator goes with the variable name (e.g. `NSObject *variableName;`)

Java:

- If a method call spans multiple lines closing bracket is on the same line as the last argument.
- If a method header doesn't fit on one line each argument goes on a separate line.
- 100 character line length

Sending a Pull Request

Code-level contributions to React Native generally come in the form of [a pull request](). The process of proposing a change to React Native can be summarized as follows:

1. Fork the React Native repository and create your branch from `master`.
2. If you've added code that should be tested, add tests.
3. If you've changed APIs, update the documentation.
4. Ensure the test suite passes, either locally or on CI once you opened a pull request.
5. Make sure your code lints (for example via `yarn lint --fix`).
6. Push the changes to your fork.
7. Create a pull request to the React Native repository.
8. Review and address comments on your pull request.
     1. A bot may comment with suggestions. Generally we ask you to resolve these first before a maintainer will review your code.
9. If you haven't already, complete the [Contributor License Agreement ("CLA")]().

If all goes well, your pull request will be merged. If it is not merged, maintainers will do their best to explain their reasoning.

If this is your first time sending a pull request, we have created a [step-by-step guide to help you get started](). For more detailed information on how pull requests are handled, see the [Pull Requests wiki]().

Contributor License Agreement

In order to accept your pull request, we need you to submit a [Contributor License Agreement (CLA)](). You only need to do this once to work on any of Facebook's open source projects. It only takes a minute so you can do it while you wait for your dependencies to install.

License

By contributing to React Native, you agree that your contributions will be licensed under the LICENSE file in the root directory of the React Native source tree.