Getting Started

This guide will help you install and build your first React Native app

Installing dependencies

You will need Node, Watchman, the React Native command line interface, and Xcode.

While you can use any editor of your choice to develop your app, you will need to install Xcode in order to set up the necessary tooling to build your React Native app for iOS.

Node, Watchman

We recommend installing Node and Watchman using Homebrew. Run the following commands in a Terminal after installing Homebrew:

If you have already installed Node on your system, make sure it is Node 8.3 or newer.

Watchman is a tool by Facebook for watching changes in the filesystem. It is highly recommended you install it for better performance.

The React Native CLI

Node comes with npm, which lets you install the React Native command line interface.

Run the following command in a Terminal:

If you get an error like `Cannot find module 'npmlog'`, try installing npm directly: `curl -0 -L https://npmjs.org/install.sh | sudo sh`.

App dependencies installation

Change directory to your project directory and run:

```
npm install

react-native link
```

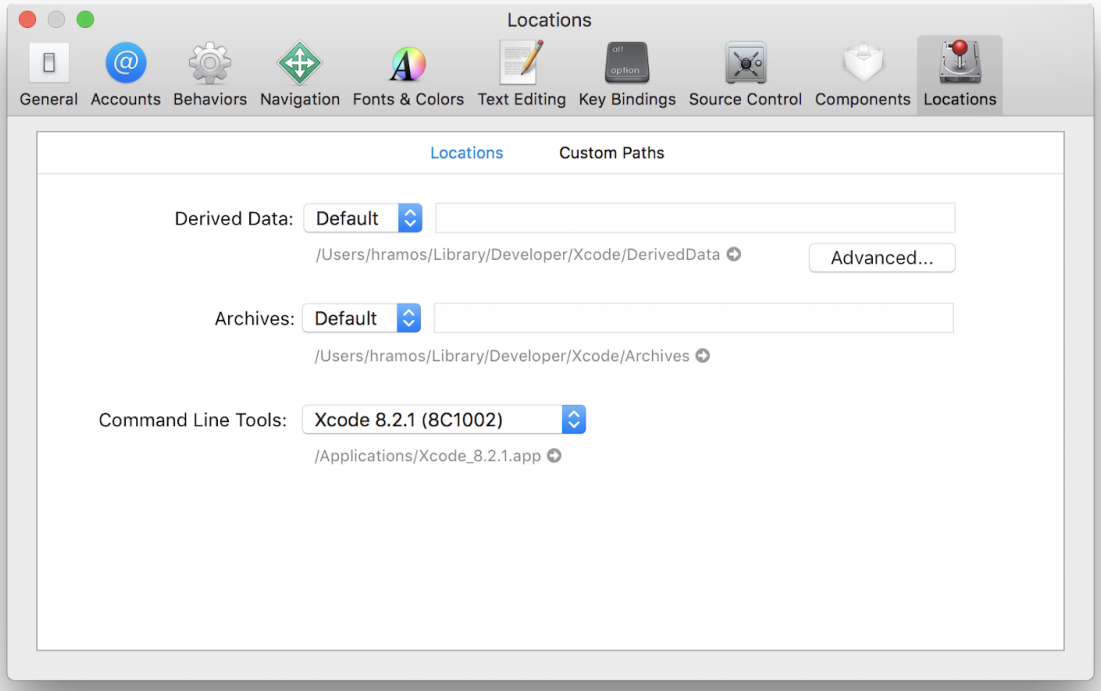
Xcode

The easiest way to install Xcode is via the Mac App Store. Installing Xcode will also install the iOS Simulator and all the necessary tools to build your iOS app.

If you have already installed Xcode on your system, make sure it is version 9.4 or newer.

Command Line Tools

You will also need to install the Xcode Command Line Tools. Open Xcode, then choose "Preferences..." from the Xcode menu. Go to the Locations panel and install the tools by selecting the most recent version in the Command Line Tools dropdown.



Android development environment

Java Development Kit

React Native requires a recent version of the Java SE Development Kit (JDK). Download and install Oracle JDK 8 if needed. You can also use OpenJDK 8 as an alternative.

Setting up your development environment can be somewhat tedious if you're new to Android development. If you're already familiar with Android development, there are a few things you may need to configure. In either case, please make sure to carefully follow the next few steps.

1. Install Android Studio

Download and install Android Studio. Choose a "Custom" setup when prompted to select an installation type. Make sure the boxes next to all of the following are checked:

- `Android SDK`
- `Android SDK Platform`
- `Performance (Intel ® HAXM)`
- `Android Virtual Device`

Then, click "Next" to install all of these components.

If the checkboxes are grayed out, you will have a chance to install these components later on.

Once setup has finalized and you're presented with the Welcome screen, proceed to the next step.

2. Install the Android SDK

Android Studio installs the latest Android SDK by default. Building a React Native app with native code, however, requires the `Android 8.1 (Oreo)` SDK in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

The SDK Manager can be accessed from the "Welcome to Android Studio" screen. Click on "Configure", then select "SDK Manager".

The SDK Manager can also be found within the Android Studio "Preferences" dialog, under Appearance & Behavior → System Settings → Android SDK.

Select the "SDK Platforms" tab from within the SDK Manager, then check the box next to "Show Package Details" in the bottom right corner. Look for and expand the `Android 8.1 (Oreo)` entry, then make sure the following items are checked:

- `Android SDK Platform 27`
- `Intel x86 Atom_64 System Image` or `Google APIs Intel x86 Atom System Image`

Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build-Tools" entry, then make sure that `27.0.3` is selected.

Finally, click "Apply" to download and install the Android SDK and related build tools.

3. Configure the ANDROID_HOME environment variable

The React Native tools require some environment variables to be set up in order to build apps with native code.

Add the following lines to your `$HOME/.bash_profile` config file:

`.bash_profile` is specific to `bash`. If you're using another shell, you will need to edit the appropriate shell-specific config file.

Type `source $HOME/.bash_profile` to load the config into your current shell. Verify that ANDROID_HOME has been added to your path by running `echo $PATH`.

Please make sure you use the correct Android SDK path. You can find the actual location of the SDK in the Android Studio "Preferences" dialog, under Appearance & Behavior → System Settings → Android SDK.

Running your React Native application

Run `react-native run-ios` inside your React Native project folder:

You should see your new app running in the iOS Simulator shortly.

`react-native run-ios` is just one way to run your app. You can also run it directly from within Xcode.

If you can't get this to work, see the Troubleshooting page.

Running on a device

The above command will automatically run your app on the iOS Simulator by default. If you want to run the app on an actual physical iOS device, please follow the instructions here.

Modifying your app

Now that you have successfully run the app, let's modify it.

- Open `App.js` in your text editor of choice and edit some lines.
- Hit `⌘R` in your iOS Simulator to reload the app and see your changes!

That's it!

Congratulations! You've successfully run and modified your React Native app.