

# Get started

This section will help you walk through the minimum required setup to run the project.

## Download Link

Carer React Native App Template

Carer App Template – is a React Native Mobile app that demo UI for Android and iPhone. Manage drugs, doctor, book appointment ,search... [codecanyon.net](https://codecanyon.net)

## System Requirements

### NodeJS

Make sure you have a recent version (8.11 or later) of Node installed globally.

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

1. `brew install node`
2. `brew install watchman`

If you have already installed Node on your system, make sure it is Node 12.x or newer.

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

## Yarn (recommend)

Please follow yarn official install guide.

<https://yarnpkg.com/en/docs/install#mac-stable>

## React Native Developer Tools

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

Run the following command in a Terminal:

```
npm install -g react-native-cli
```

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

Or with yarn :

```
yarn global add react-native-cli
```

Note: (Ubuntu) make sure you add ~/.yarn/bin to your PATH

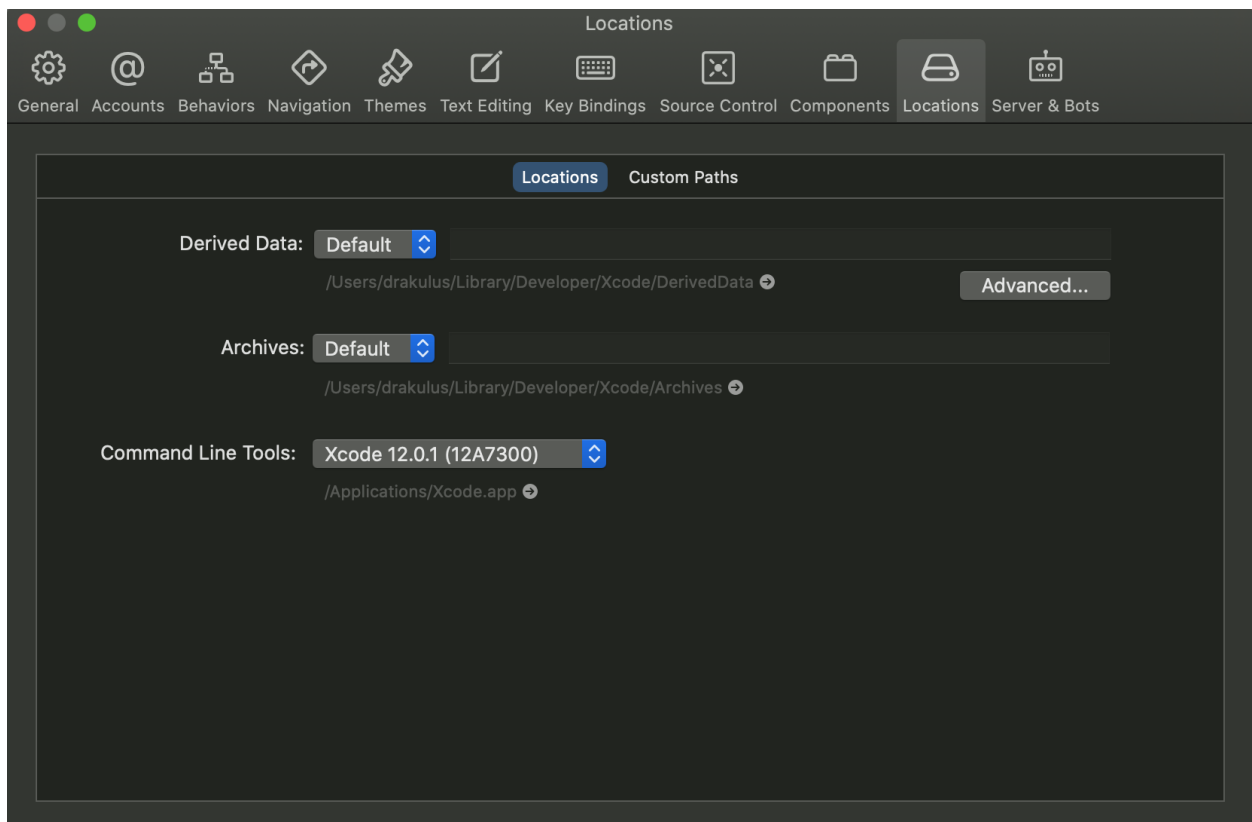
# Xcode & CocoaPods

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 12.0 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.



## Installing an iOS Simulator in Xcode

To install a simulator, open Xcode > Preferences...and select the Components tab. Select a simulator with the corresponding version of iOS you wish to use.

## CocoaPods

CocoaPods is built with Ruby and it will be installable with the default Ruby available on macOS. You can use a Ruby Version manager, however we recommend that you use the standard Ruby available on macOS unless you know what you're doing.

Using the default Ruby install will require you to use sudo when installing gems (This is only an issue for the duration of the gem installation, though.)

```
sudo gem install cocoapods
```

For more information, please visit

[CocoaPods Getting Started guide](#)

# Java Development Kit

We recommend installing JDK using Homebrew. Run the following commands in a Terminal after installing Homebrew:

```
brew cask installadoptopenjdk/openjdk/adoptopenjdk14
```

If you have already installed JDK on your system, make sure it is JDK 14 or newer.

## Android development environment

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) (See here for AMD)
- Android Virtual Device

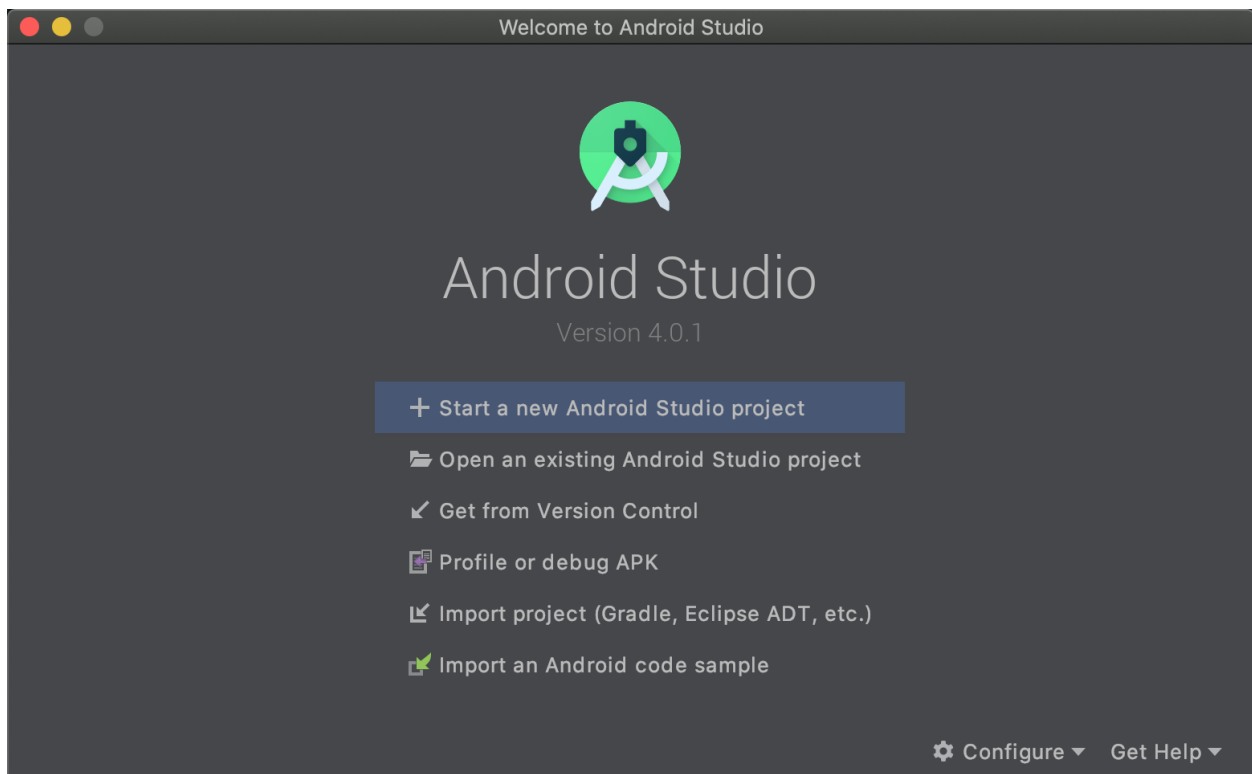
Then, click "Next" to install all of these components. If the checkboxes are grayed out, you will have a chance to install these components later on. Once setup has finalized and you're presented with the Welcome screen, proceed to the next step.

## 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 9 (Pie) 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 9 (Pie) entry, then make sure the following items are checked:

- Android SDK Platform 28
  - 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 29.0.2 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 `$HOME/.bash_profile` or `$HOME/.bashrc` config file:

```
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
```

`.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`

# Expo CLI

Assuming that you have Node 10+ installed, you can use npm to install the Expo CLI command line utility:

```
npm install -g expo-cli
```

Or with yarn

```
yarn global add expo-cli
```

## Running UI with Expo

Install node modules with command:

```
yarn  
yarn start
```

Or

```
npm install  
npm run start
```

## Running on a simulator

Run `react-native run-ios` or `react-native run-android` inside your React Native project folder.

```
cd /project_path/expo_carer  
react-native run-ios  
react-native run-android
```



# 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.

<https://facebook.github.io/react-native/docs/running-on-device>

## Our products:

Evez React Native: <https://1.envato.market/emE1r>

Finany – Cashflow Manager App: <https://1.envato.market/d1v17>

Imba React Native: <https://1.envato.market/M3KWM>

Healer React Native: <https://1.envato.market/v6BDd>

Healer Flutter: <https://1.envato.market/b1xk9>

Finey React Rative: <https://1.envato.market/dJQyM>

Finey Flutter: <https://1.envato.market/RdB09>

Kid Draw React Native: <https://1.envato.market/Z3nQk>