** **

**Carer TimiStudio**

****

This page will help you install and build your React Native app.

**System Requirements :**

1. **NodeJS** : Make sure you have a recent version ( [12 LTS](https://nodejs.org/en/download/) or or greater installed )
2. **Watchman :** [Watchman](https://facebook.github.io/watchman) is a tool by Facebook for watching changes in the filesystem. It is highly recommended you install it for better performance.
3. **Yarn :** Please follow [yarn official](https://classic.yarnpkg.com/en/docs/install#mac-stable) install guide.

*$ brew install node*

*$ brew install watchman*

**1. MacOS - Android :**

### **Java Development Kit**

### We recommend installing JDK using [Homebrew](http://brew.sh/). Run the following commands in a Terminal after installing Homebrew:

### *$ brew install --cask adoptopenjdk/openjdk/adoptopenjdk8*

If you have already installed JDK on your system, make sure it is JDK 8 or newer.

### **Android development environment**

#### **1. Install Android Studio**

[Download and install Android Studio](https://developer.android.com/studio/index.html). While on Android Studio installation wizard, make sure the boxes next to all of the following items are checked:

* Android SDK
* Android SDK Platform
* Android Virtual Device

#### **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 10 (Q) SDK in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

#### To do that, open Android Studio, click on "Configure" button and select "SDK Manager".

#### **Android Studio Welcome**

#### 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 10 (Q) entry, then make sure the following items are checked:

#### Android SDK Platform 29

#### 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 and check the "Android SDK Command-line Tools (latest)".

#### 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 (if you are using zsh then ~/.zprofile or ~/.zshrc) config file:

*export ANDROID\_HOME=$HOME*

*export PATH=$PATH:$ANDROID\_HOME*

*export PATH=$PATH:$ANDROID\_HOME*

*export PATH=$PATH:$ANDROID\_HOME*

*export PATH=$PATH:$ANDROID\_HOME*

#### *.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* for *bash* or *source $HOME/.zprofile* to load the config into your current shell. Verify that ANDROID\_HOME has been set by running *echo $ANDROID\_HOME* and the appropriate directories have 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**.

**4.Running your React Native application**

### **Step 1: Start Metro**

To start Metro, run npx react-native start inside your React Native project folder:

*$ npx react-native start*

or

*$ yarn react-native start*

*If you use the Yarn package manager, you can use yarn instead of npx when running React Native commands inside an existing project.*

### **Step 2: Start your application**

Let Metro Bundler run in its own terminal. Open a new terminal inside your React Native project folder. Run the following:

*$ npx react-native run-android*

or

*$ yarn react-native run-android*

**2. MacOS - IOS :**

### **Xcode**

The easiest way to install Xcode is via the [Mac App Store](https://itunes.apple.com/us/app/xcode/id497799835?mt=12). Installing Xcode will also install the iOS Simulator and all the necessary tools to build your iOS app.

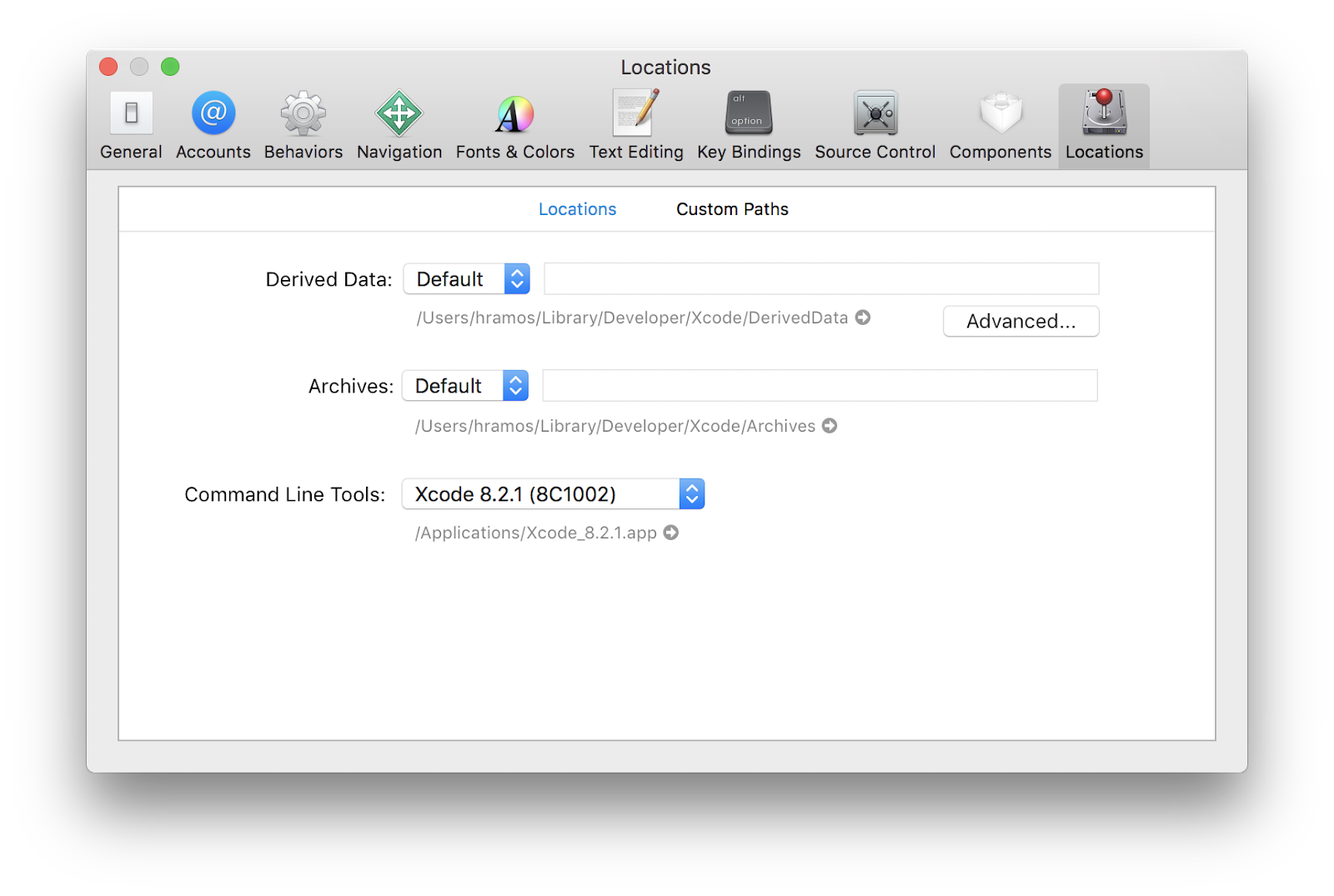
If you have already installed Xcode on your system, make sure it is version 10 or newer.

#### 

#### 

#### **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](https://cocoapods.org/) 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.)

*$ sodo gem install cocoapods*

For more information, please visit [CocoaPods Getting Started guide](https://guides.cocoapods.org/using/getting-started.html).

**Running application IOS**

Open terminal inside project :

1. Node modules with command:

*$ npm*

or

*$ yarn*

2. Install package by CocoaPods with command:

*$ cd ios && pod install*

3. Start Metro with command:

*$ npx react-native start*

or

*$ yarn react-native start*

4. Run :

*$ npx react-native run-ios*

or

*$ yarn react-native run-ios*

**3.Windows- Android :**

**NodeJS** : [Chocolatey](https://chocolatey.org/)

If you want to be able to switch between different Node versions, you might want to install Node via [nvm-windows](https://github.com/coreybutler/nvm-windows), a Node version manager for Windows.

React Native also requires [Java SE Development Kit (JDK)](https://openjdk.java.net/projects/jdk8/), which can be installed using Chocolatey as well.

Open an Administrator Command Prompt:

*$ choco install -y nodejs.install openjdk8*

You can find additional installation options on [Node's Downloads page](https://nodejs.org/en/download/).

If you're using the latest version of Java Development Kit, you'll need to change the Gradle version of your project so it can recognize the JDK. You can do that by going to

*{project root folder}\android\gradle\wrapper\gradle-wrapper.properties*

and changing the distributionUrl value to upgrade the Gradle version. You can check out [here the lastest releases of Gradle](https://gradle.org/releases/).

### **Android development environment**

#### **1. Install Android Studio**

[Download and install Android Studio](https://developer.android.com/studio/index.html). While on Android Studio installation wizard, make sure the boxes next to all of the following items are checked:

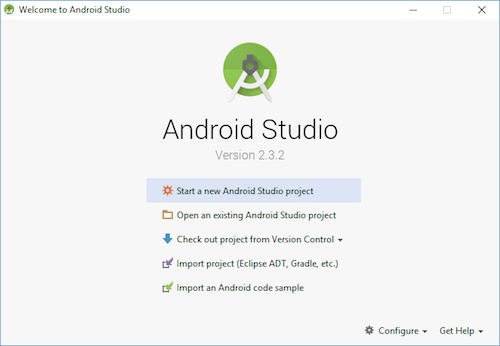
* Android SDK
* Android SDK Platform
* Android Virtual Device

#### 

#### 

#### 

#### **2. Install the Android SDK**



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 10 (Q) entry, then make sure the following items are checked:

* *Android SDK Platform 29*
* *Intel x86 Atom\_64 System Image or Google APIs Intel x86 Atom System Image*

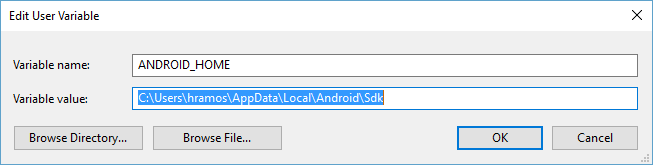
Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the "Android SDK Build-Tools" entry, then make sure that 29.0.2 is selected.

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.

1. Open the **Windows Control Panel**.
2. Click on **User Accounts**, then click **User Accounts** again
3. Click on **Change my environment variables**
4. Click on **New...** to create a new *ANDROID\_HOME* user variable that points to the path to your Android SDK:



The SDK is installed, by default, at the following location:

*%LOCALAPPDATA%\Android\Sdk*

You can find the actual location of the SDK in the Android Studio "Settings" dialog, under **Appearance & Behavior → System Settings → Android SDK***.*

Open a new Command Prompt window to ensure the new environment variable is loaded before proceeding to the next step.

1. Open powershell
2. Copy and paste **Get-ChildItem -Path Env:\** into powershell
3. Verify *ANDROID\_HOME* has been added

#### 

#### 

#### **4. Add platform-tools to Path**

1. Open the **Windows Control Panel**.
2. Click on **User Accounts**, then click **User Accounts** again
3. Click on **Change my environment variables**
4. Select the **Path** variable.
5. Click **Edit**.
6. Click **New** and add the path to platform-tools to the list.

The default location for this folder is:

*%LOCALAPPDATA%\Android\Sdk\platform-tools*

**4.Running your React Native application**

### **Step 1: Start Metro**

To start Metro, run npx react-native start inside your React Native project folder:

*$ npx react-native start*

or

*$ yarn react-native start*

*If you use the Yarn package manager, you can use yarn instead of npx when running React Native commands inside an existing project.*

### **Step 2: Start your application**

Let Metro Bundler run in its own terminal. Open a new terminal inside your React Native project folder. Run the following:

*$ npx react-native run-android*

Or *$ yarn react-native run-android*

Our products:

Runner Flutter Full Application: <https://1.envato.market/qnAOaq>   
Triplan Flutter Template: <https://1.envato.market/doyJdK>

Stacy Roommate Flutter Template: <https://1.envato.market/YgR1Xj>

Fitness Love: <https://1.envato.market/x9kDjx>

Doctor Plus – Patient React Native: <https://1.envato.market/x99yJR>

Doctor Plus – Doctor React Native: <https://1.envato.market/x07Bk>

Carer – Patient React Native App Template: <https://1.envato.market/RzAzy>

Wala – Food React Native: <http://1.envato.market/qbXgy>

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>

Thank you!

# 