## **Installing dependencies**

You will need Node, the React Native command line interface, a JDK, and Android Studio.

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

### **Node, JDK**

We recommend installing Node via [Chocolatey](https://chocolatey.org/install), a popular package manager for Windows.

It is recommended to use an LTS version of Node. If you want to be able to switch between different 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/jdk/17/), which can be installed using Chocolatey as well.

Open an Administrator Command Prompt (right click Command Prompt and select "Run as Administrator"), then run the following command:

powershell

choco install -y nodejs-lts microsoft-openjdk17

If you have already installed Node on your system, make sure it is Node 18 or newer. If you already have a JDK on your system, we recommend JDK17. You may encounter problems using higher JDK versions.

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 latest releases of Gradle](https://gradle.org/releases/).

### **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](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
* If you are not already using Hyper-V: Performance (Intel ® HAXM) ([See here for AMD or Hyper-V](https://android-developers.googleblog.com/2018/07/android-emulator-amd-processor-hyper-v.html))

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

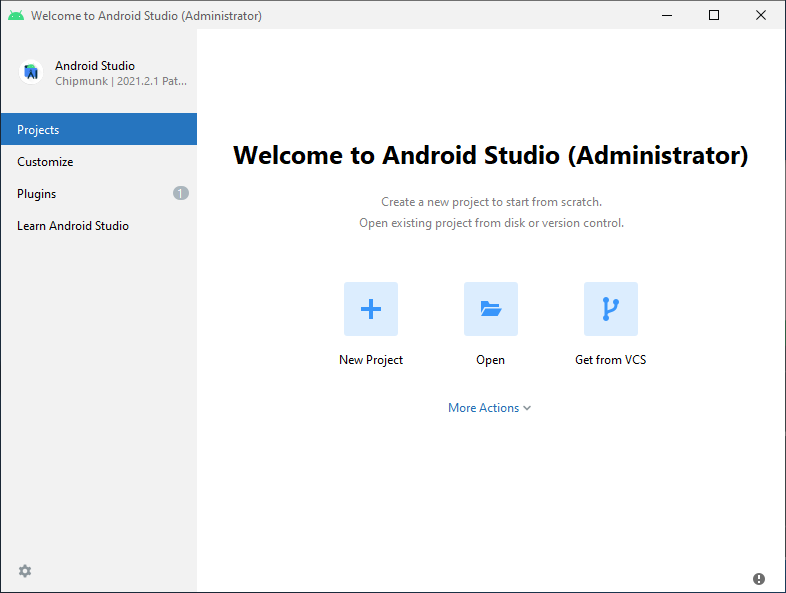
If the checkboxes are grayed out, you will have a chance to install these components later on.

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 15 (VanillaIceCream) SDK in particular. Additional Android SDKs can be installed through the SDK Manager in Android Studio.

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



The SDK Manager can also be found within the Android Studio "Settings" dialog, under Languages & Frameworks → 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 15 (VanillaIceCream) entry, then make sure the following items are checked:

* Android SDK Platform 35
* Intel x86 Atom\_64 System Image or Google APIs Intel x86 Atom System Image

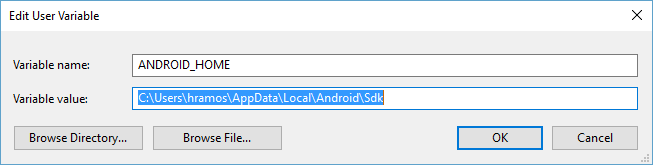
Next, select the "SDK Tools" tab and check the box next to "Show Package Details" here as well. Look for and expand the Android SDK Build-Tools entry, then make sure that 35.0.0 is selected.

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:

powershell

%LOCALAPPDATA%\Android\Sdk

You can find the actual location of the SDK in the Android Studio "Settings" dialog, under Languages & Frameworks → 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:

powershell

%LOCALAPPDATA%\Android\Sdk\platform-tools

## **Preparing the Android device**

You will need an Android device to run your React Native Android app. This can be either a physical Android device, or more commonly, you can use an Android Virtual Device which allows you to emulate an Android device on your computer.

Either way, you will need to prepare the device to run Android apps for development.

### **Using a physical device**

If you have a physical Android device, you can use it for development in place of an AVD by plugging it in to your computer using a USB cable and following the instructions [here](https://reactnative.dev/docs/running-on-device).

### **Using a virtual device**

If you use Android Studio to open ./AwesomeProject/android, you can see the list of available Android Virtual Devices (AVDs) by opening the "AVD Manager" from within Android Studio. Look for an icon that looks like this:

Android Studio AVD Manager

If you have recently installed Android Studio, you will likely need to [create a new AVD](https://developer.android.com/studio/run/managing-avds.html). Select "Create Virtual Device...", then pick any Phone from the list and click "Next", then select the VanillaIceCream API Level 35 image.

If you don't have HAXM installed, click on "Install HAXM" or follow [these instructions](https://github.com/intel/haxm/wiki/Installation-Instructions-on-Windows) to set it up, then go back to the AVD Manager.

Click "Next" then "Finish" to create your AVD. At this point you should be able to click on the green triangle button next to your AVD to launch it.

### **That's it!**

Congratulations! You successfully set up your development environment.



## **Now what?**

* If you want to add this new React Native code to an existing application, check out the [Integration guide](https://reactnative.dev/docs/integration-with-existing-apps).
* If you're curious to learn more about React Native, check out the [Introduction to React Native](https://reactnative.dev/docs/getting-started).

Is this page useful?

[Edit page for next release](https://github.com/facebook/react-native-website/edit/main/docs/set-up-your-environment.md)[Edit page for current release](https://github.com/facebook/react-native-website/edit/main/website/versioned_docs/version-0.80/set-up-your-environment.md)

*Last updated on* ***Jun 17, 2025***

[Previous](https://reactnative.dev/docs/environment-setup)

[Get Started with React Native](https://reactnative.dev/docs/environment-setup)