Setting up the development environment for React Native Android development using Android Studio.

# System requirements

1. Windows XP, Windows 7 (32/64 bit) or higher
2. Minimum 4 GB RAM and higher
3. 10 GB available space on the hard disk
4. At least one Internet Browser e.g., Chrome, Firefox, Microsoft Edge, etc.
5. Node.js
6. Active internet connection minimum speed 512kbps and above.
7. At least one installed code Editor to test and debug your code e.g.

# Installation instructions

Setting up the React Native environment will require us to use either Expo Go or React Native CLI. This guide will focus on React Native CLI. React Native CLI requires Xcode or Android Studio to begin. Android Studio will be the focus of these instructions. This guide will be for a Windows development OS with an Android Target OS.

The first step to setting up the environment will be to install dependencies. You will need to install Node, the React Native command line interface, a JDK, and Android Studio. It is recommended to install Node via [Chocolatey](https://community.chocolatey.org/), which is a well-known Windows package manager. While downloading Node, it is recommended to use the LTS version. React Native also requires a Java SE Development Kit (JDK), which can also be installed using Chocolatey as well.

To download JDK11 and the LTS version of Node run the command prompt as Administrator and then run the following command: “choco install -y nodejs-lts microsoft-openjdk11”

**Note:** To find additional installation options explore [Node’s Downloads page.](https://nodejs.org/en/download/)

**Note:** If you happen to be using the latest version of JDK, you must change the Gradle version of the project for the project to recognize the JDK. To do this, go to {project root folder}\android\gradle\wrapper\gradle-wrapper.properties and change the distributionUrl value to upgrade the Gradle version. Check out the latest Gradle releases [here](https://gradle.org/releases/).

Next, there are a few things to configure for Android development. To begin, [install Android Studio](https://developer.android.com/studio/index.html). While installing make sure the Android Studio and Android Virtual Device options on the Choose Components page are checked and press next.

Once installed, you’ll want to open Android Studio. While Android Studio installs the latest Android SDK by default, building a React Native app with native code requires the Android 13 (Tiramisu) SDK. To install this SDK, use the SDK Manager in Android Studio. To navigate to the SDK Manager Open Android Studio, click on “More Action” and select “SDK Manager”.

Select the “SDK Platforms” tab within the SDK Manager and check the box next to “Show Package Details” in the bottom right corner. Search for the “Android (Tiramisu)” entry and check the boxes for the following items:

* Android SDK Platform 33
* Intel x86 Atom\_64 System Image OR Google APIs Intel x86 Atom System Image

Once the two items have been selected navigate to the “SDK Tools” tab within the SDK Manager and check the box next to “Show Package Details”. Search for the “Android SDK Build-Tools entry and make sure that 33.0.0 is selected. Finally, use the “Apply” button to download and install the Android SDK and the related build tools.

Next, configure the ANDROID\_HOME environment variables for some tools used by React Native. To do so, open the Windows Control Panel. Once open, click on User Accounts, and then click on User Accounts again. On this page, click on “Change my environment variables”. A new window will pop up. To create a new ANDROID\_HOME user variable you must click on “New…” and name it “ANDROID\_HOME” and provide the path that leads to your Android SDK (e.g., C:\Users\glecl\AppData\Local\Android\Sdk).

**Note:** You can find the actual location of the SDK in the Android Studio “Settings” dialog under “Languages & Frameworks” → “Android SDK”

Once the ANDROID\_HOME environment variable has been created “platform-tools” must be added to the Path variable. Select the Path variable and click Edit. Click New and add the path to the platform-tools to the list. This default path will be the path to your Android SDK with “\platform-tools” concatenated to the end of it. (e.g., C:\Users\glecl\AppData\Local\Android\Sdk\platform-tools). Carefully close the Windows Control Panel, saving your changes as you do.

The next step is to create a new application. To do so, we must make sure that any previously installed global “react-native-cli” packages are removed. To do so we can run the following command in the terminal we wish to create our new application:

“npm uninstall -g react-native-cli @react-native-community/cli”

Once all previously installed global “react-native-cli” packages are removed we can create a new React Native application named “SuperCoolProject” by running the command:

“npx react-native@latest init AwesomeProject”

**Note:** “AwesomeProject” can be replaced with the desired name of the application.

Now that the application has been created, we need an Android device to run your React Native Android app. A physical Android device can be used, but this guide will focus on using an Android Virtual Device that allows us to emulate an Android device on your computer.

To set up a virtual device use Android Studio to open “./AwesomeProject/android”. This may take a while. Once opened, see the list of available Android Virtual Devices by opening the “AVD Manager” from within Android Studio. Choose to create a new virtual device, then pick any phone from the list and click “Next”, then select the Tiramisu API Level 33 image. Click on “Next” then “Finish” to create your AVD. You should be able to launch your AVD by clicking the green triangle button next to the AVD’s information.

The next step of the process is starting Metro, the JavaScript bundler that ships with React native. To start metro run “npm start” in your React Native project folder. Once ran, the command will prompt you to return one of the following values:

* r - reload the app
* d - open the developer menu
* i - run on iOS.
* a - run on Android

type “a” into the terminal and your app should load on your AVD. This may take a while.

# Resources

|  |  |
| --- | --- |
| [1] | React Native, "Setting up the development environment," React Native, 29 Aug 2023. [Online]. Available: https://reactnative.dev/docs/environment-setup?guide=native. [Accessed 10 11 2023]. |