How to setup Android SDK without Android Studio.

Easily setup Android SDK using Android CLI for React Native, Flutter etc.

Nitish Sharma · Follow

Published in ProAndroidDev · 5 min read · Mar 22, 2021

Photo by Glenn Carstens-Peters on Unsplash

The foundation of android mobile development using any library is the “Android SDK”. Android SDK is the prerequisite for building android apps, be it via native Kotlin, or other popular libraries like React Native and Flutter.

Now I’ve been building apps using React Native for about 4 years now, and didn’t have any need for a full-fledged Android Studio IDE other than to install SDK(s) and emulator(s). Also I’ll be honest, it’s a big IDE, till last month I was using a early 2015 macbook air with 128G of storage, so you can guess yourself how precious the space was to me.

Also, I like using command line as much as I can, because for me, it’s easier than the GUI (debatable, I know, but we all have our preferences).

So I looked for a way to install Android SDK and other stuff, without installing the “Android Studio”, and I found it. Fortunately, Google has provided us with Android Command-Line Tools. So in this article I would like to show you how you can set it up.

Prerequisites

For this guide I assume you’ve already installed the Java JDK of your choice. I’d suggest installing openjdk8, as it is prime choice for Android development, You can install it via commands below.

For Mac -

```
brew install --cask adoptopenjdk/openjdk/adoptopenjdk8
```

For Linux -

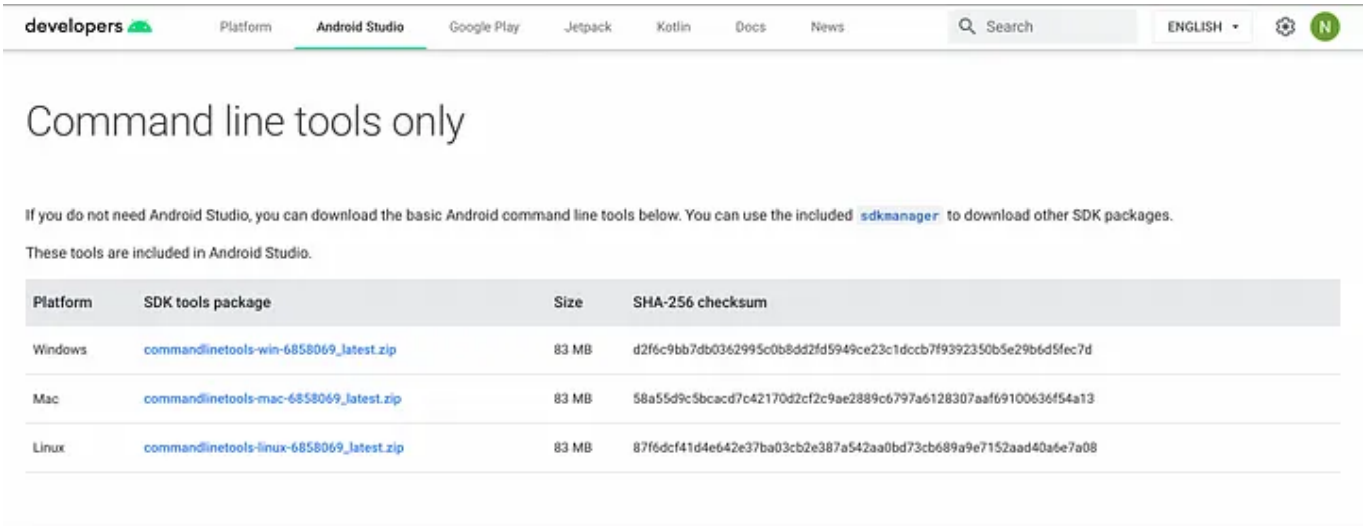
You may download and install OpenJDK from AdoptOpenJDK or your system packager.

Moving on, follow the Steps below to setup Android tools and install Android SDK.

Step 1 — Download the Command Line Tools

Yes, Download, instead of directly installing them, I know this is a drag but just bear with me.

Click on this link to visit the download page, then to the Command Line Tools Only section, and Download the zip file according to you operating sytem (preferably Linux OR Mac, If you are using windows, switch your OS).



Here is the section you need to visit and click on the tools next to your operating system

Now after you've downloaded the zip file, move it to your home location i.e. `~/ .`

Now there's another way to do this in one step, you can just copy the link to the zip file, then open a terminal window and:

```
~ $ wget <your zip link here>
```

Step 2 — Setting up the Android Tools (CLI)

Now that you've dowloaded the tools zip and moved it to home folder of your system, we can go on ahead on setting them up, so that the CLI is available to you.

First we need to create a directory to store the android sdk and other stuff, so open a terminal window and follow the steps:

```
~ $ mkdir android
~ $ cd android
```

Then we need to move and unzip the tools in android directory we just created:

```
~/android $ mv ~/commandlinetools-mac-6858069_latest.zip ./
~/android $ unzip commandlinetools-mac-6858069_latest.zip
~/android $ rm commandlinetools-mac-6858069_latest.zip
```

You can change the file name according to yours. At the time this article was written this was the latest zip avaialable (for mac).

Now, here's the tricky part which even confused me the first time I setup android tools.

The above created android directory will act as our $ANDROID_HOME so other libraries can access is from the environment variables we're going to add ahead.

After unzipping the content, you will get a directory named **cmdline-tools**, now follow the next steps carefully.

In the android directory:

```
$ cd cmdline-tools
$ mkdir tools
$ mv -i * tools
```

Last command will probably give you a warning, but you don't need the worry about that.

After running the commands above the new directory structure should look like something like this:

```
android
└── cmdline-tools
    └── tools
        ├── NOTICE.txt
        ├── bin
        ├── lib
        └── source.properties
```

Step 3 — Adding tools to $PATH.

If you don't have any experience with the environment variable $PATH, this guide will probably give you a start. $PATH is used to tell the terminal where the binaries are to be located, that are defined by user.

The file in which you have to append the PATH of the tools is in your home directory ~ . for a bash terminal it's **.bash_profile** where as for the newer zsh terminals it's **.zshrc**.

Now before we can add tools to path we have to add $ANDROID_HOME to the path, to do that just open the **.zshrc** or **.bash_profile** in you preffered

terminal file editor (nano or vim) and add the following code at the end of the file:

```
export ANDROID_HOME=$HOME/android
export PATH=$ANDROID_HOME/cmdline-tools/tools/bin/:$PATH
export PATH=$ANDROID_HOME/emulator/:$PATH
export PATH=$ANDROID_HOME/platform-tools/:$PATH
```

After adding the code, save the file, close the terminal window and open a new terminal window (I prefer this way as it makes reloading easier, without extra commands).

After you've opened a new terminal window just type the following command and hit return/enter.

```
$ sdkmanager
```

If you see the following progressbar, your tools have setup successfully:

```
[==                                    ] 6% Fetch remote
repository...
```

If not you can go through the guide and check if you've followed the steps carefully. If the probrem persists, feel free to drop in a comment.

Step 4 — Installing the Android SDK

If you've reached this step, congratulations, the journey ahead is clean and simple.

Use the following command to list all the available sdks, platform-tools, build tools, emulator, ndks and what-not.

```
$ sdkmanager --list
```

To install the package you want, just copy the package name and install:

```
$ sdkmanager --install "package_name"
```

The basic packages you should install are, `platform-tools` , `platforms;android-29` , `build-tools;29.0.2` , `emulator` .

```
$ sdkmanager --install "platform-tools" "platforms;android-29"
"build-tools;29.0.2" "emulator"
```

This will install all the basic necessary tools you'll require to start up your android development.

Conclusion

If you've reached here, Congratulations, again. You've successfully setup your Android Development Environment, without Android Studio. You can use the library you'd like to work with i.e React Native or Flutter etc.

Although, the tools above should suffice for building your basic debug app, but, if any other tools are to be installed, the libraries handle the installation automatically for the most part, if not you can follow **Step 4**.

Thanks for Reading.

CHEERS!

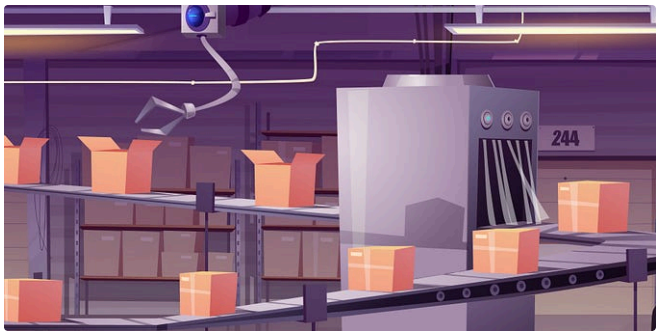
Android | Command Line | Mobile App Development | React Native | Flutter

Written by Nitish Sharma

18 Followers · Writer for ProAndroidDev

Web and Mobile Developer (React.Js, React Native, Expo, Electron, Firebase, Amplify).
India 🇮🇳 https://www.linkedin.com/in/nitishxyz

More from Nitish Sharma and ProAndroidDev



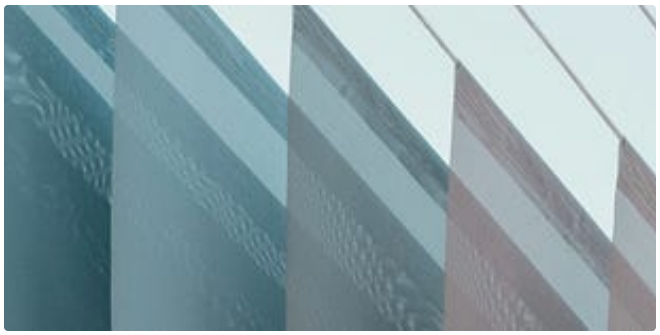
Tom Colvin in ProAndroidDev

Seven recipes to understand flows and asynchrony in Kotlin

In the clean world of Kotlin coroutines, we can have many tasks running at different times....

8 min read · Mar 14, 2024

650 3



Nick Rout in ProAndroidDev

Opinion: Jetpack Compose needs a Design System layer

Material Design should be a choice, not the default.

23 min read · Mar 15, 2024

740

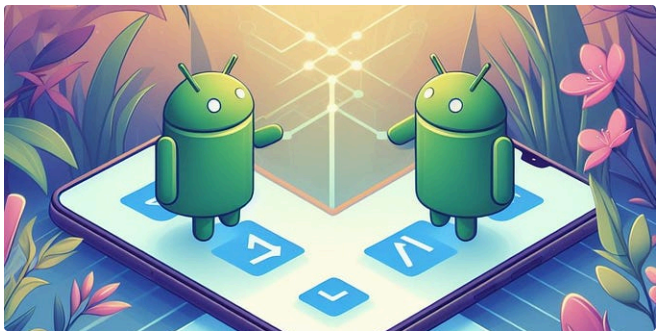
Tom Colvin in ProAndroidDev

Seven demos to understand coroutines: scope, context and...

Understanding coroutines—really understanding them, not just learning...

6 min read · Feb 18, 2024

530 5



Muhammad Mohsin Shafqat in ProAndroidDev

Choosing Between Sealed Class and Sealed Interface in Kotlin for...

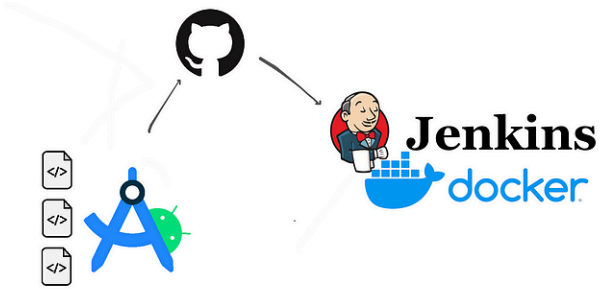
2 min read · Feb 22, 2024

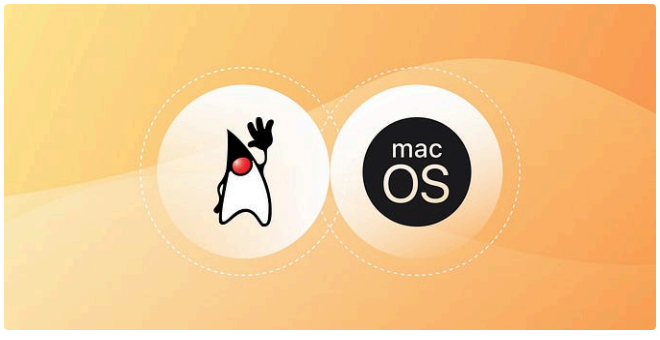
322 7

See all from Nitish Sharma

See all from ProAndroidDev

Recommended from Medium





Jorge Luis Castro Medina

How to build a CI/CD Pipeline for Android with Jenkins and Docker...

Jenkins in Docker: A Solution for Android CI/CD

8 min read · Oct 12, 2023

242

Snyk

Installing and managing Java on macOS

5 min read · Oct 12, 2023

13

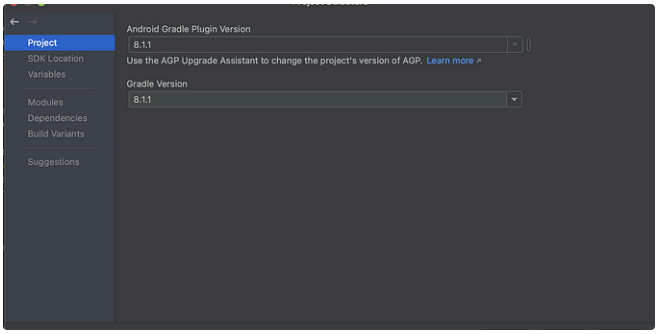
Lists

Stories to Help You Grow as a Software Developer

19 stories · 924 saves

Staff Picks

605 stories · 854 saves



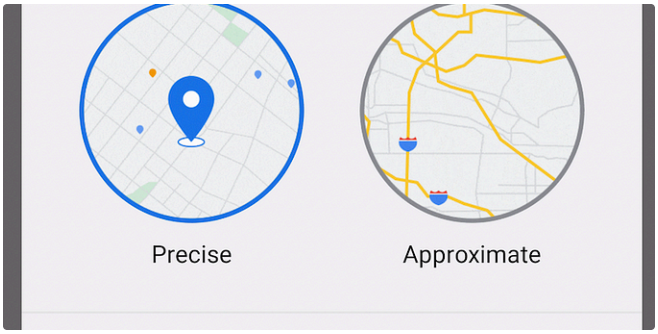
ramji sri

Gradle Plugin Version vs Gradle Version

When starting the development of a new Android application today, you may encount...

2 min read · Nov 14, 2023

1



Mun Bonecci

How to get your location in Jetpack Compose

In this short tutorial, we will see how to obtain your location and request the necessary...

6 min read · Feb 2, 2024

52 1

bectorhimanshu

How to Set JAVA_HOME environment variable on MacOS X...

Having the JAVA_HOME environment variable properly configured on your Mac running OS...

3 min read · Oct 13, 2023

bhavana vadodariya

How to reverse engineer Android applications on Linux system—2...

Diving deep into the sea of reverse engineering resources, it's hard to ignore the...

3 min read · Oct 18, 2023

16

See more recommendations