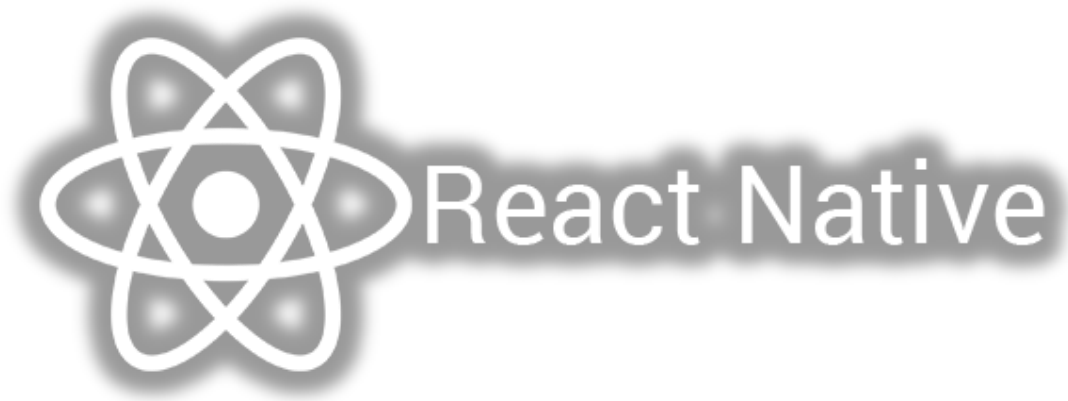


Mobile Development








# React Native

**#1** Getting Started



- ***React Native*** lets you build mobile apps (iOS and/or Android) using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components.
- More info: [\*facebook.github.io/react-native\*](https://facebook.github.io/react-native)

# Top JS Mobile Frameworks 2017

1		<b>React Native</b> A framework for building native apps with React.	+15.6k ★
2		<b>Weex</b> A framework for building Mobile cross-platform UI.	+6.5k ★
3		<b>Ionic</b> Build amazing native and progressive web apps ...	+5.5k ★
4		<b>Quasar</b> Quasar Framework	+3.7k ★
5		<b>NativeScript</b> NativeScript is an open source framework for buil...	+2.9k ★

<https://risingstars.js.org/2017/en/>



# Advantages

- Design simple declarative views for each state in an application.
- Encapsulated components.
- Dynamics properties & state.
- Light & faster virtual DOM.
- Independent of the rest of application.
- Can render on client or server.

# Initial Setup:

- ❖ **Java Dev Kit**
- ❖ **Android Studio**
- ❖ **Android SDK**
- ❖ **ANDROID\_HOME**
- ❖ **JAVA\_HOME**
- ❖ **Platform-tools**



# #1 Java Dev Kit

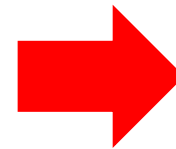
- Download then install jdk8u161:  
[www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

## Java SE 8u161/ 8u162

Java SE 8u161 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u162 is a patch-set update, including all of 8u161 plus additional bug fixes (described in the release notes).

[Learn more](#) ➔

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Licensing Information User Manual](#)
  - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme Files](#)
  - [JDK ReadMe](#)



### JDK

DOWNLOAD ⬇

### Server JRE

DOWNLOAD ⬇

### JRE

DOWNLOAD ⬇

- Download then install Android Studio:  
[developer.android.com/studio/index.html](https://developer.android.com/studio/index.html).
- Choose a "Custom" setup and make sure the following boxes are checked:

***Android SDK***

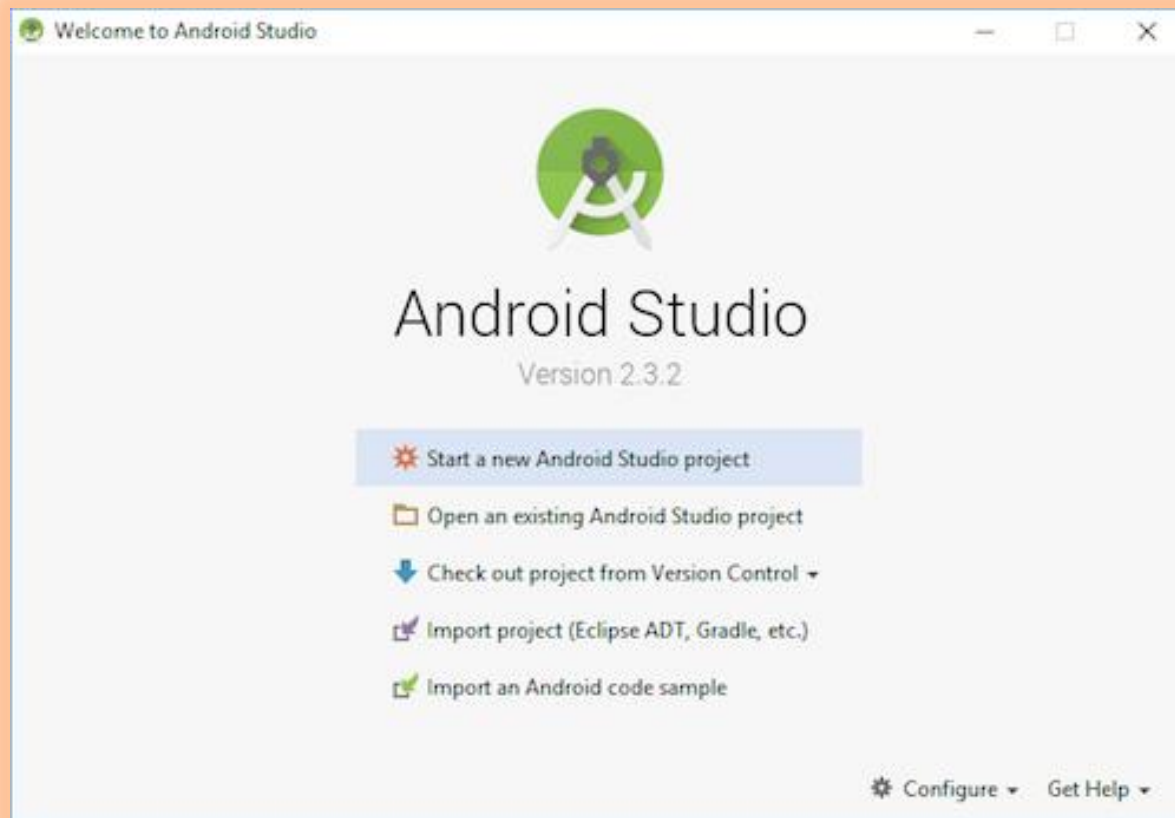
***Android SDK Platform***

***Performance (Intel® HAXM)***

***Android Virtual Device***

# #3 Android SDK

- Access SDK Manager from the **"Welcome to Android Studio"** screen. Click on **Configure**, then select **SDK Manager**.





## #3 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 6.0 (Marshmallow)** entry, then make sure the following items are all checked:

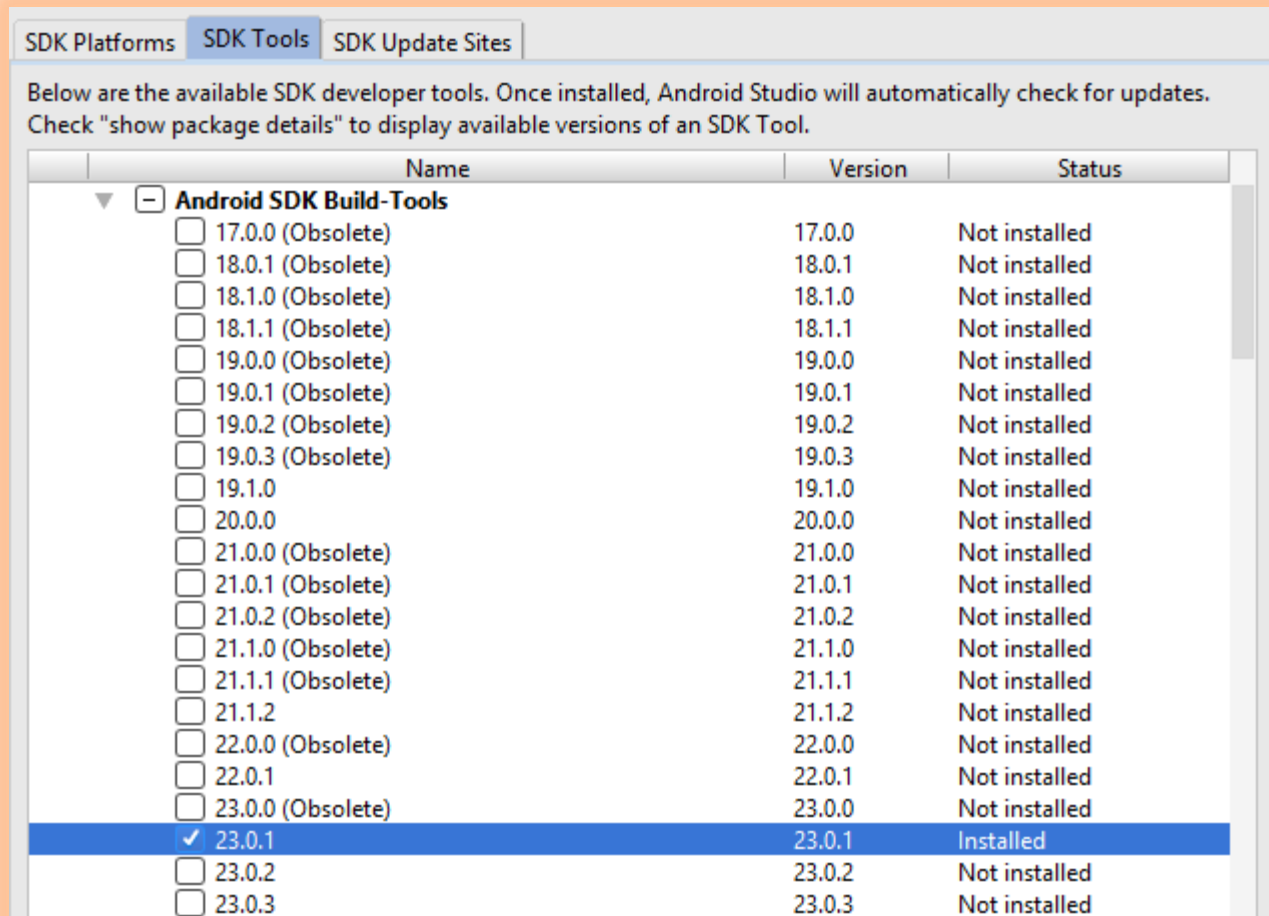
SDK Platforms | SDK Tools | SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Google APIs Intel x86 Atom System Image	24	11	Not installed
<input type="checkbox"/>	Google APIs Intel x86 Atom_64 System Image	24	11	Not installed
<input type="checkbox"/>	Google Play Intel x86 Atom System Image	24	12	Not installed
▼ <input checked="" type="checkbox"/>	<b>Android 6.0 (Marshmallow)</b>			
	<input checked="" type="checkbox"/> Google APIs	23	1	Not installed
	<input checked="" type="checkbox"/> Android SDK Platform 23	23	3	Not installed
	<input type="checkbox"/> Sources for Android 23	23	1	Not installed
	<input type="checkbox"/> Android TV ARM EABI v7a System Image	23	3	Not installed
	<input type="checkbox"/> Android TV Intel x86 Atom System Image	23	9	Not installed
	<input type="checkbox"/> Android Wear ARM EABI v7a System Image	23	6	Not installed
	<input type="checkbox"/> Android Wear Intel x86 Atom System Image	23	6	Not installed
	<input type="checkbox"/> ARM EABI v7a System Image	23	6	Not installed
	<input type="checkbox"/> Intel x86 Atom System Image	23	9	Not installed
	<input checked="" type="checkbox"/> Intel x86 Atom_64 System Image	23	9	Not installed
	<input type="checkbox"/> Google APIs ARM EABI v7a System Image	23	20	Not installed
	<input type="checkbox"/> Google APIs Intel x86 Atom System Image	23	20	Not installed
	<input checked="" type="checkbox"/> Google APIs Intel x86 Atom_64 System Image	23	20	Not installed
▼ <input type="checkbox"/>	<b>Android 5.1 (Lollipop)</b>			
	<input type="checkbox"/> Google APIs	22	1	Not installed
	<input type="checkbox"/> Android SDK Platform 22	22	2	Not installed
	<input type="checkbox"/> Google APIs ARM EABI v7a System Image	22	1	Not installed

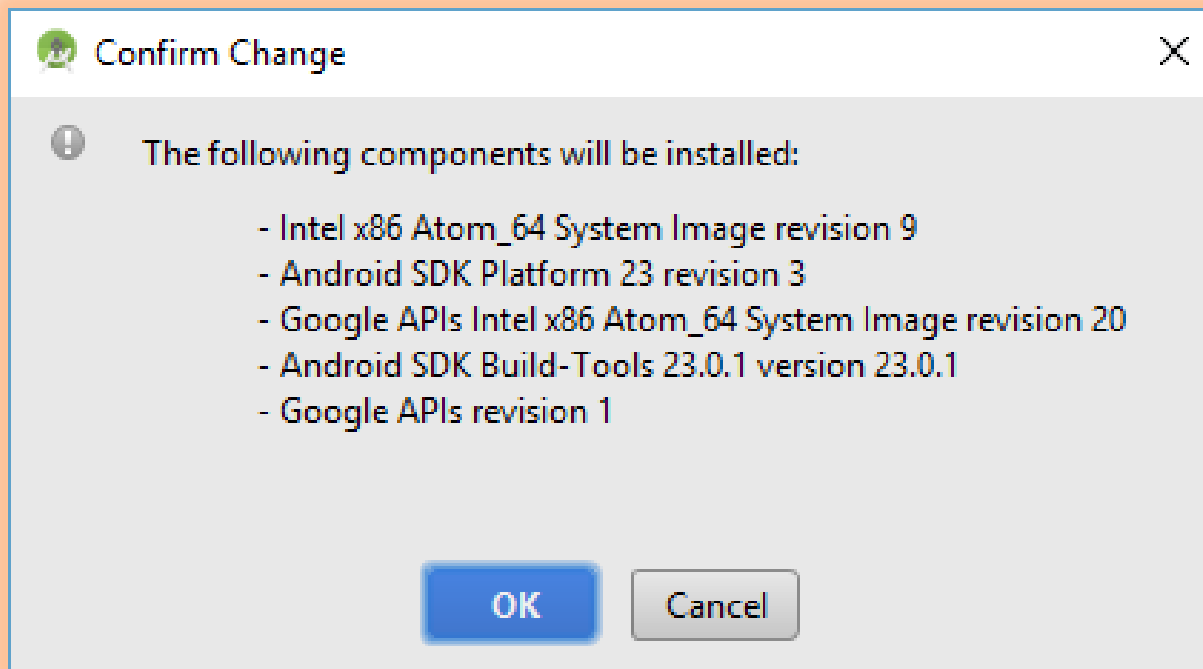
# #3 Android SDK

- 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 **23.0.1** is selected.



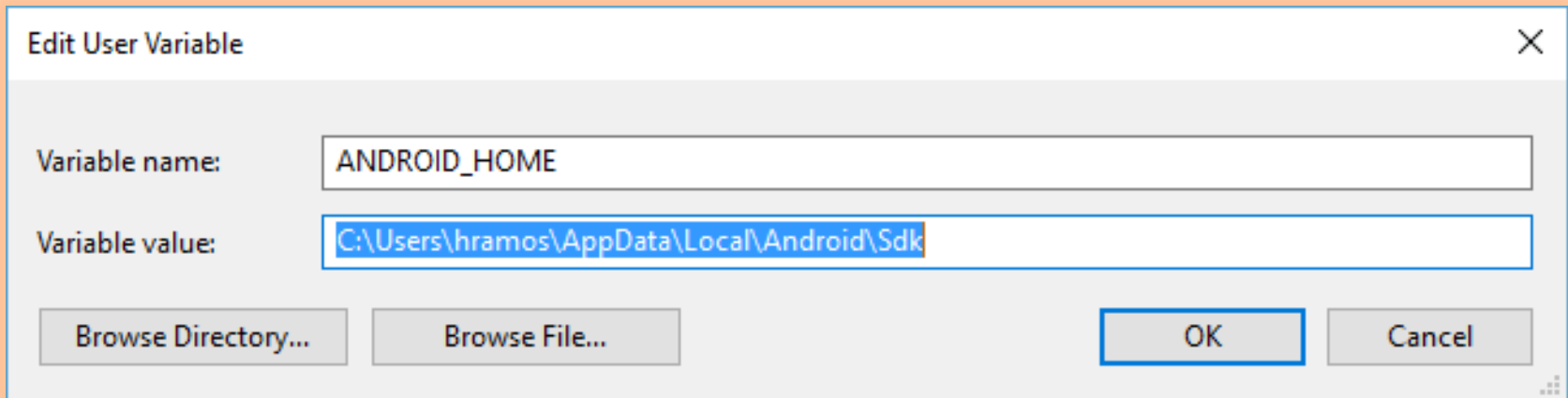
# #3 Android SDK

- Finally, click ***Apply*** to download and install the Android SDK and related build tools.



## #4 ANDROID\_HOME

- Right click **This PC (My Computer) > Properties > Advanced system**. Next, click on **Environment Variables** on the bottom right. Under **User variables** click on **New...** to create a new **ANDROID\_HOME** user variable that points to the path to your Android SDK:



Edit User Variable

Variable name: ANDROID\_HOME

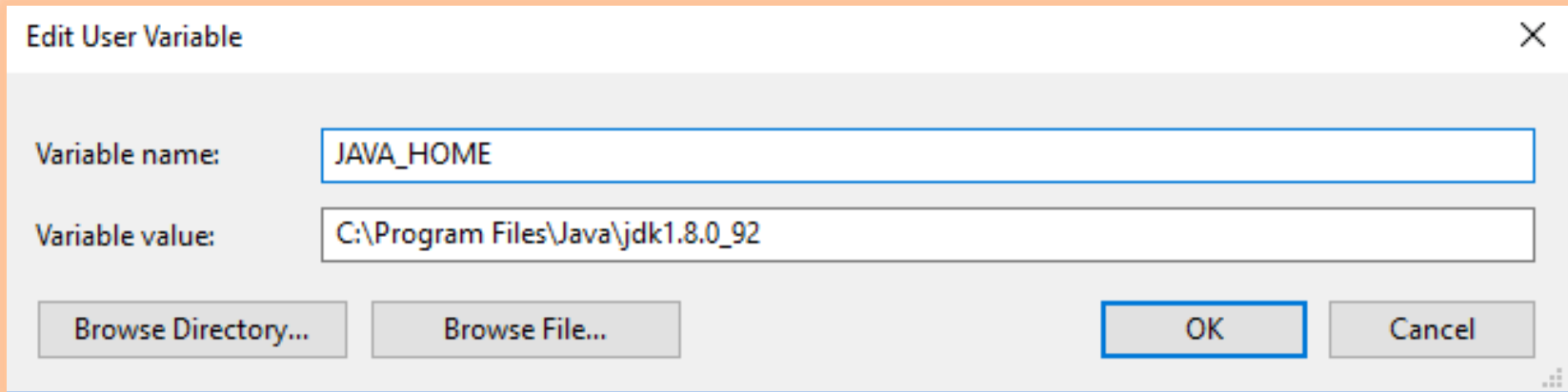
Variable value: C:\Users\hramos\AppData\Local\Android\Sdk

Browse Directory... Browse File... OK Cancel

**ANDROID\_HOME → SDK**

## #5 JAVA\_HOME

- Right click **This PC (My Computer) > Properties > Advanced system**. Next, click on **Environment Variables** on the bottom right. Under **User variables** click on **New...** to create a new **JAVA\_HOME** user variable that points to the path to your Android JDK:



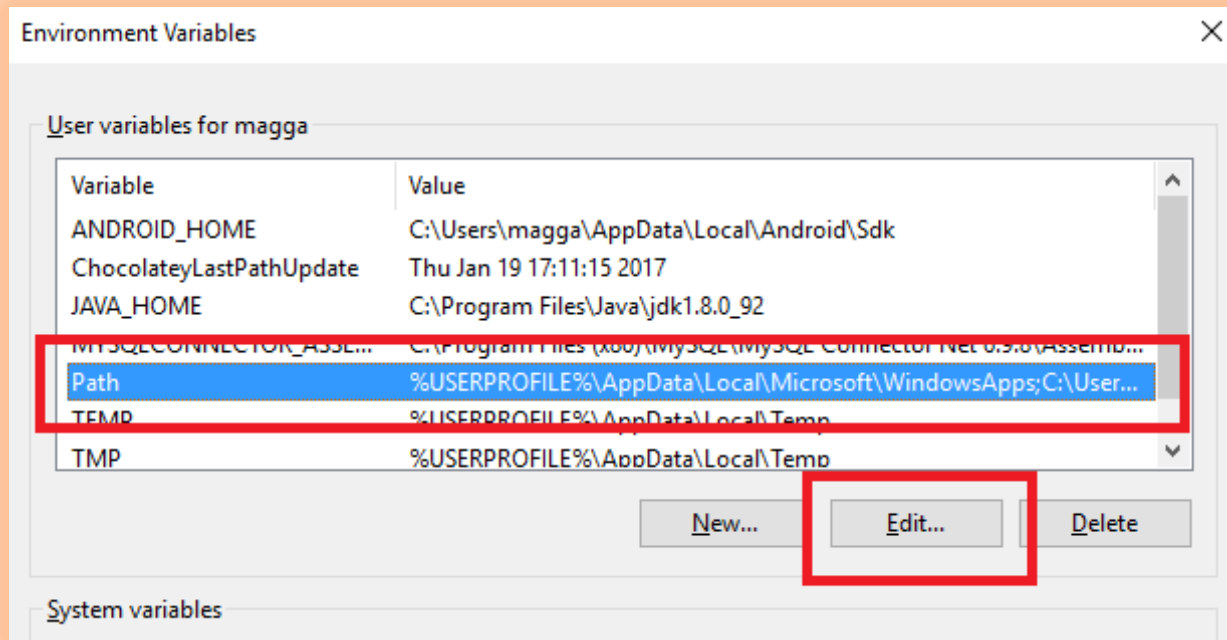
Edit User Variable

Variable name:

Variable value:

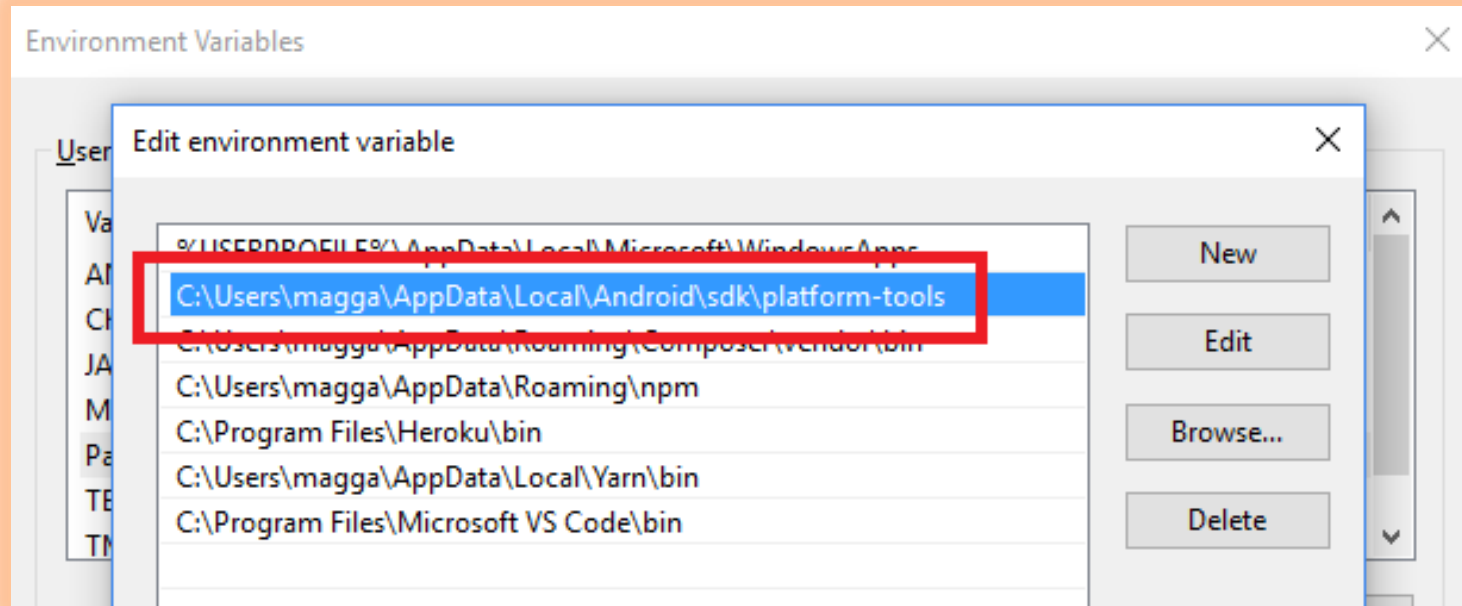
**JAVA\_HOME → JDK**

- Right click **This PC (My Computer) > Properties > Advanced system**. Next, click on **Environment Variables** on the bottom right. Under **User variables** click **Path** then **Edit...**



## #6 platform\_tools

- Click New & add the path to Android SDK's platform tools:  
`C:\Users\name\AppData\Local\Android\Sdk\platform-tools`
- Click **OK** to finish the setup.



# **React Native Setup:**

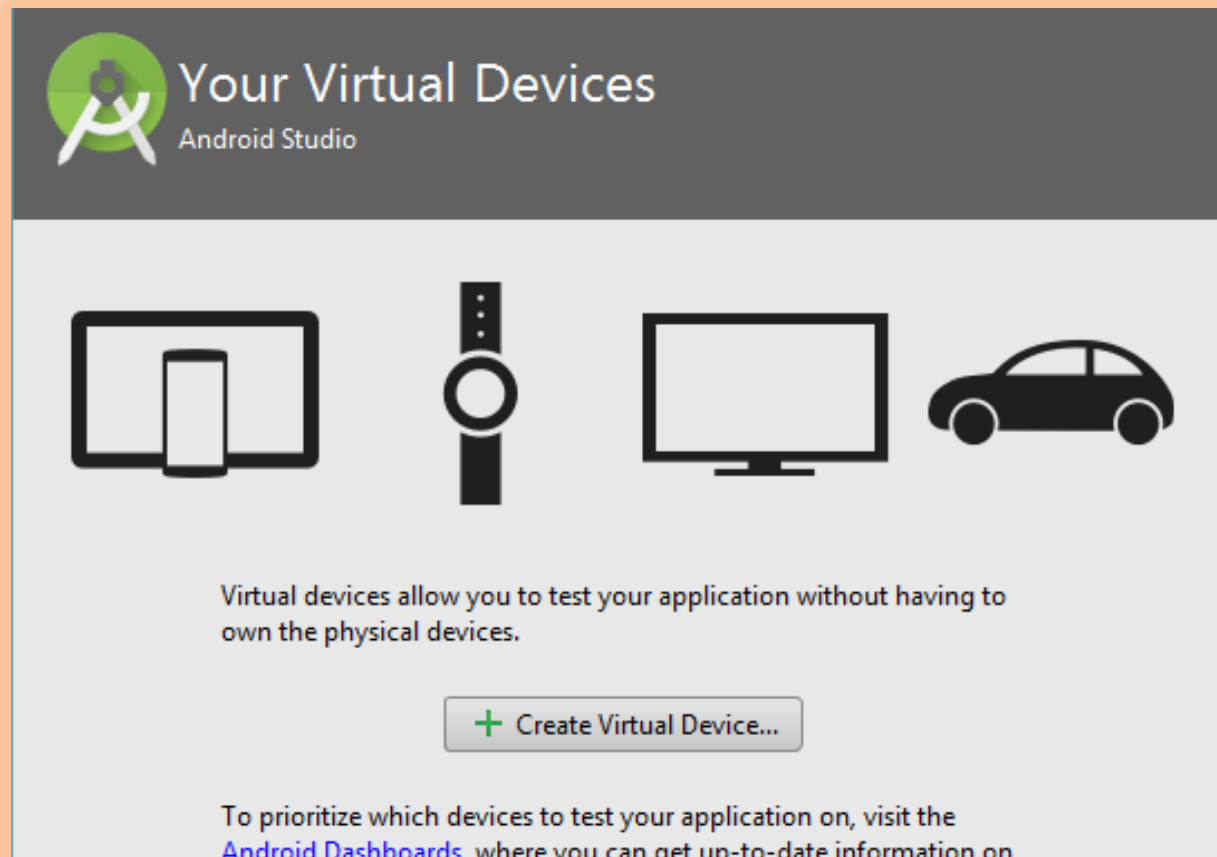
- ❖ **Set Android AVD**
- ❖ **React Native CLI**
- ❖ **local.properties**
- ❖ **Running Project**






# #1 Android Virtual Device

- Open Android Studio then ***Start a New Android Studio project.*** Follow the process until done.
- If you have just installed Android Studio, you will likely need to create a new AVD. Select ***Create Virtual Device...***, then pick any Phone from the list and click ***Next.***



# #1 Android Virtual Device

- Select the *x86 Images* tab, then look for the *Marshmallow API Level 23, x86\_64* ABI image with a *Android 6.0 (Google APIs)* target.


 System Image  
Android Studio

Select a system image

Recommendedx86 ImagesOther Images

Release Name	API Level	ABI	Target
<b>Nougat</b>	25	x86	Android 7.1.1 (Google APIs)
<a href="#">Nougat Download</a>	25	x86_64	Android 7.1.1 (Google APIs)
<a href="#">Nougat Download</a>	24	x86	Android 7.0 (Google APIs)
<a href="#">Nougat Download</a>	24	x86_64	Android 7.0 (Google APIs)
<a href="#">Nougat Download</a>	24	x86_64	Android 7.0
<a href="#">Nougat Download</a>	24	x86	Android 7.0
<b>Marshmallow</b>	23	x86_64	Android 6.0 (Google APIs)
<a href="#">Marshmallow Download</a>	23	x86	Android 6.0 (Google APIs)
<b>Marshmallow</b>	23	x86_64	Android 6.0
<a href="#">Marshmallow Download</a>	23	x86	Android 6.0
<a href="#">Lollipop Download</a>	22	x86	Android 5.1 (Google APIs)
<a href="#">Lollipop Download</a>	22	x86_64	Android 5.1 (Google APIs)
<a href="#">Lollipop Download</a>	22	x86_64	Android 5.1
<a href="#">Lollipop Download</a>	22	x86	Android 5.1
<a href="#">Lollipop Download</a>	21	x86_64	Android 5.0 (Google APIs)

Marshmallow



API Level

**23**

Android

**6.0**

Google Inc.

System Image

**x86\_64**

Recommendation





HAXM is not installed.

[Install Haxm](#)

Questions on API level?  
See the [API level distribution chart](#)

# #1 Android Virtual Device

- If you don't have HAXM installed, click on *Install HAXM* or follow these instructions to set it up, then go back to the *AVD Manager*.

Type	Name	Play Store	...	API	...	CPU/ABI	S...	Actions
	Nexus 5X API 23		.	23	.	x86_64	.	  

- 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, then proceed to the next step.



# #2 React Native CLI

- Install React Native CLI first. Open terminal, on project directory type:

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

- Create React Native project:

```
$ react-native init namaProjek  
$ cd namaProjek
```



# #3 local.properties

- Then you have to create a *local.properties* file on your *projekDir/android*! Fill it with your Android SDK path, for instance:

```
sdk.dir=C:\\Users\\lintangwisesa\\AppData  
\\Local\\Android\\Sdk
```

- Make sure your AVD (Android emulator) is running, then run our React Native project:

```
$ react-native run-android
```



- If everything is set up correctly, you should see your new app running in your Android Studio emulator shortly.

- Press **CTRL+M** to show main menu on our emulator.

Reload

Debug JS Remotely

Enable Live Reload

Enable Hot Reloading

Toggle Inspector

Show Perf Monitor

Start/Stop Sampling Profiler

Dev Settings





**Using Genymotion is  
Lighter Than Android  
Studio AVD!**



***GenyMotion*** is a third party program that offers Android virtualization so you can test against different versions of Android, and for debugging.

Download its personal edition for free (with Virtual Box):  
***[www.genymotion.com/fun-zone](http://www.genymotion.com/fun-zone)***

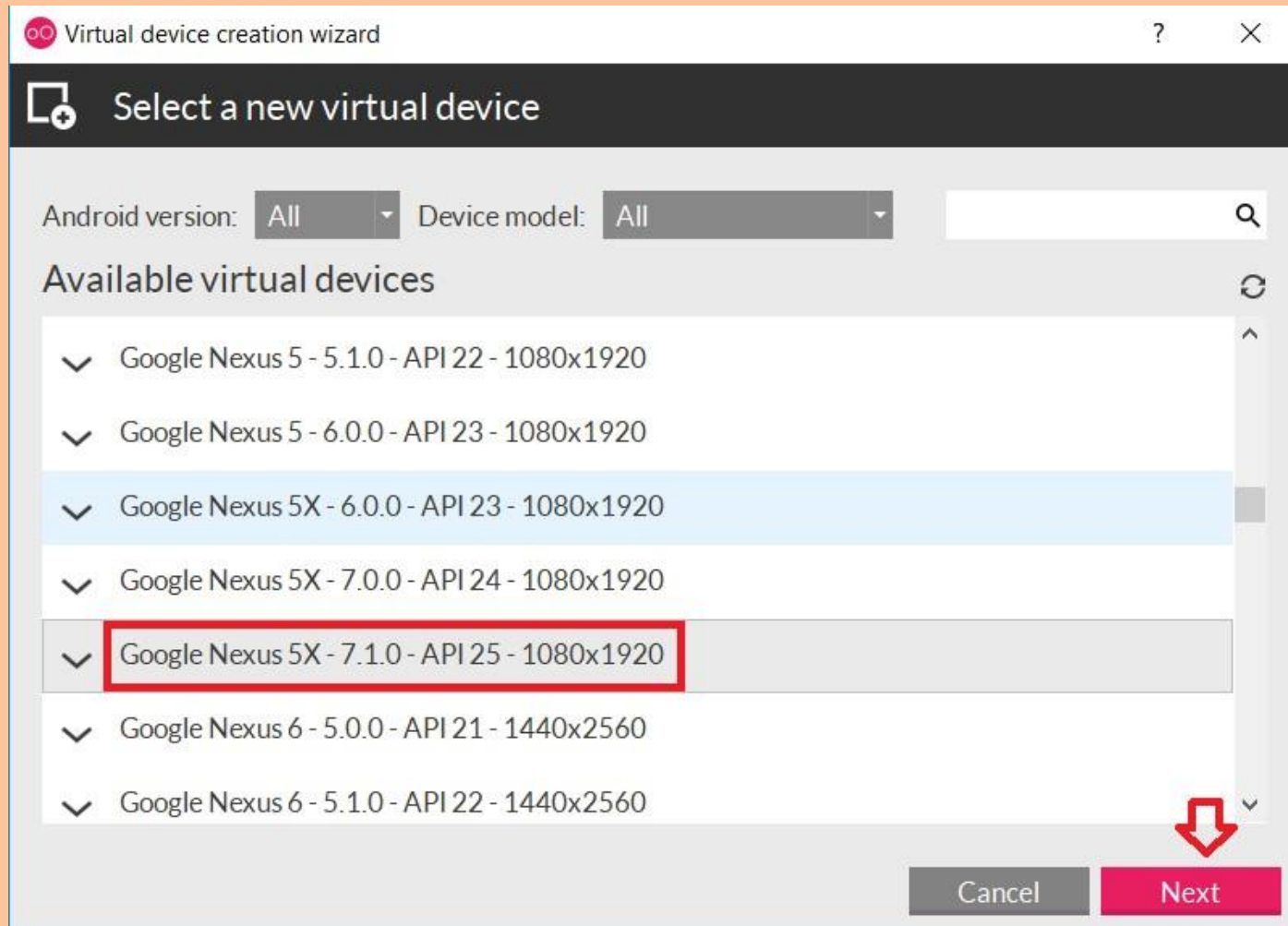
A screenshot of the GenyMotion website's download page. The page has a dark header with the GenyMotion logo and navigation links: Solutions, Pricing, Help, Resources, and Other Products. On the right of the header are links for "Contact Us" and "Trial", and a user profile icon. The main content area is white and features the heading "Download Genymotion 2.11.0". Below this heading, there are two main sections. The left section is titled "Download Genymotion 2.11.0" and contains two download options: "with VirtualBox:" with a "Download for Windows - 152MB" button, and "without VirtualBox:" with a "Download for Windows - 46M" button. A red arrow points to the "with VirtualBox:" button. Below these buttons is a link "How to register my license". The right section is titled "System Requirements" and lists the following: "Microsoft Windows 7, 8/8.1, 10 (32/64 bit)", "64 bit CPU, with VT-x or AMD-V capability, enabled in BIOS settings", "Recent and dedicated GPU", "400 MB disk space", and "2GB RAM". At the bottom of this section are two links: "Checksum Windows (with VirtualBox)" and "Checksum Windows (without VirtualBox)".





# #1 Add Virtual Device

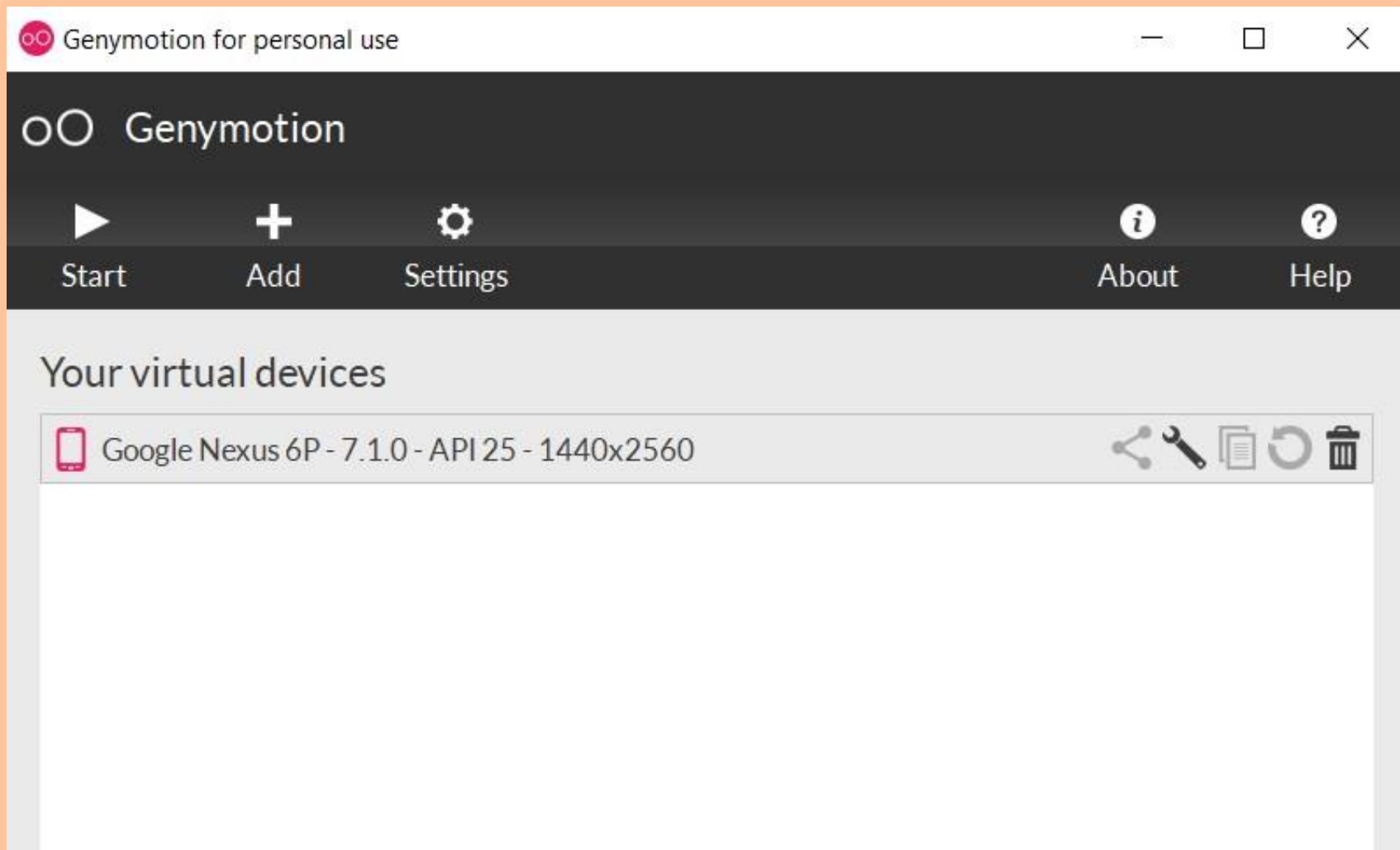
- Once downloaded, open Genymotion and login with your account. Now, you'll be able to add a new Virtual Device (Ctrl+N).





# #1 Add Virtual Device

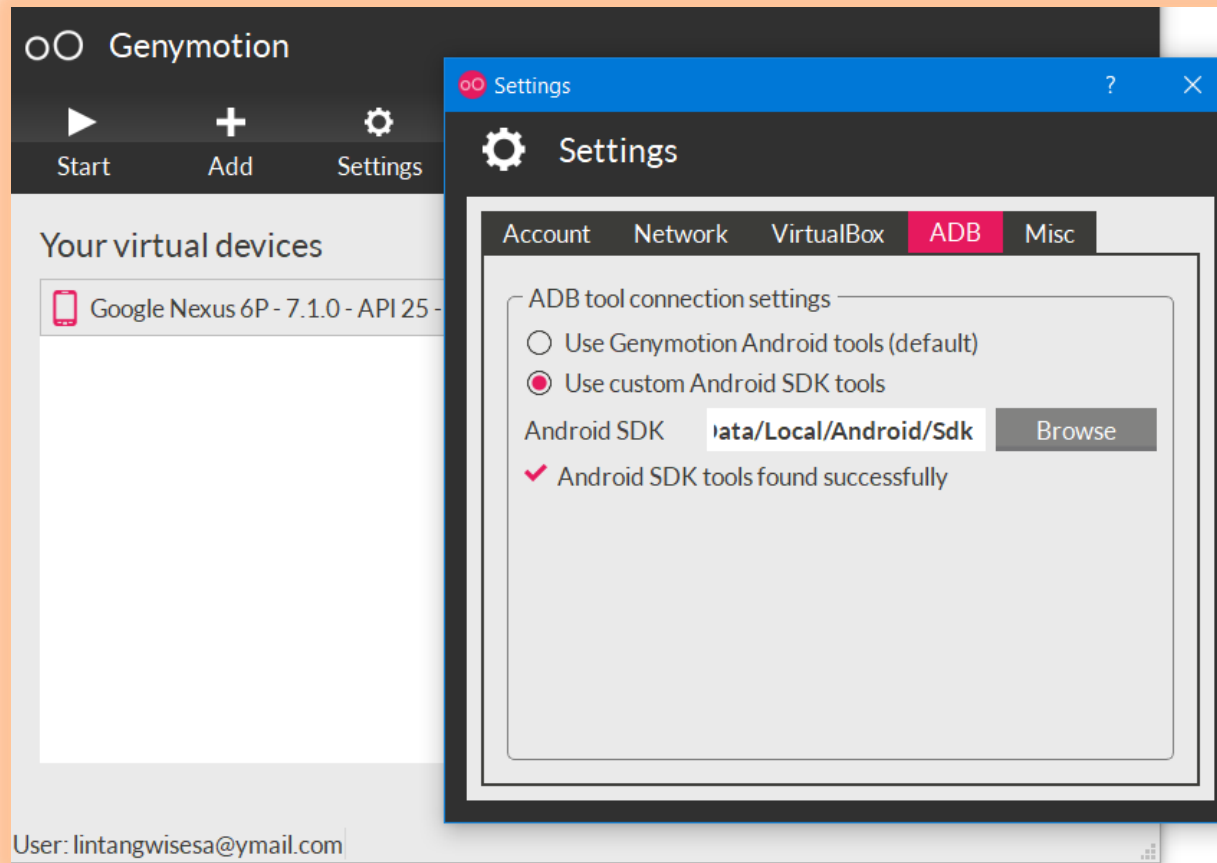
- Once done you'll see your devices listed under Virtual devices:





## #2 ADB to SDK

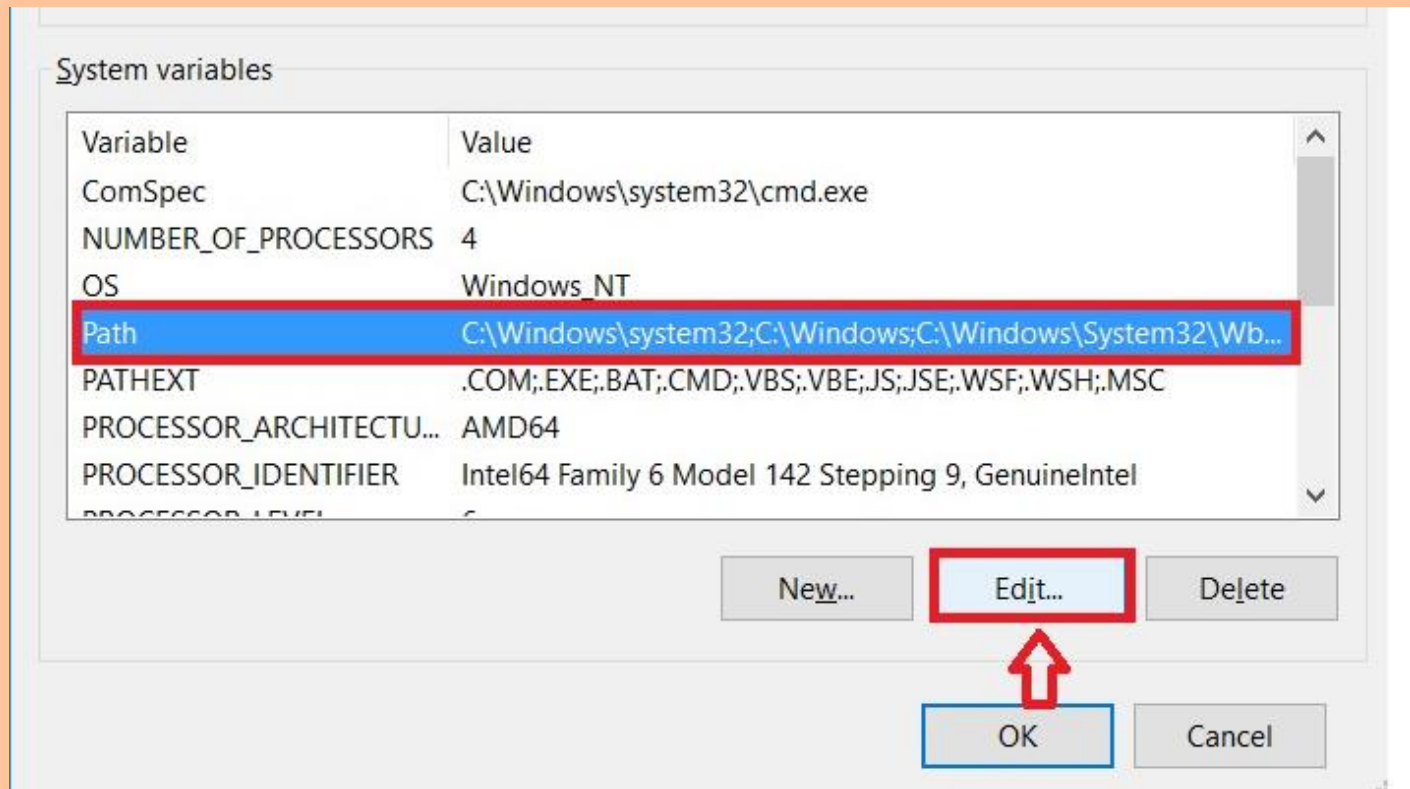
- On Genymotion, click **Settings**, on tab **ADB** choose **Use custom Android SDK tools** then browse your Android SDK's path, e.g:  
`C:\Users\name\AppData\Local\Android\Sdk`





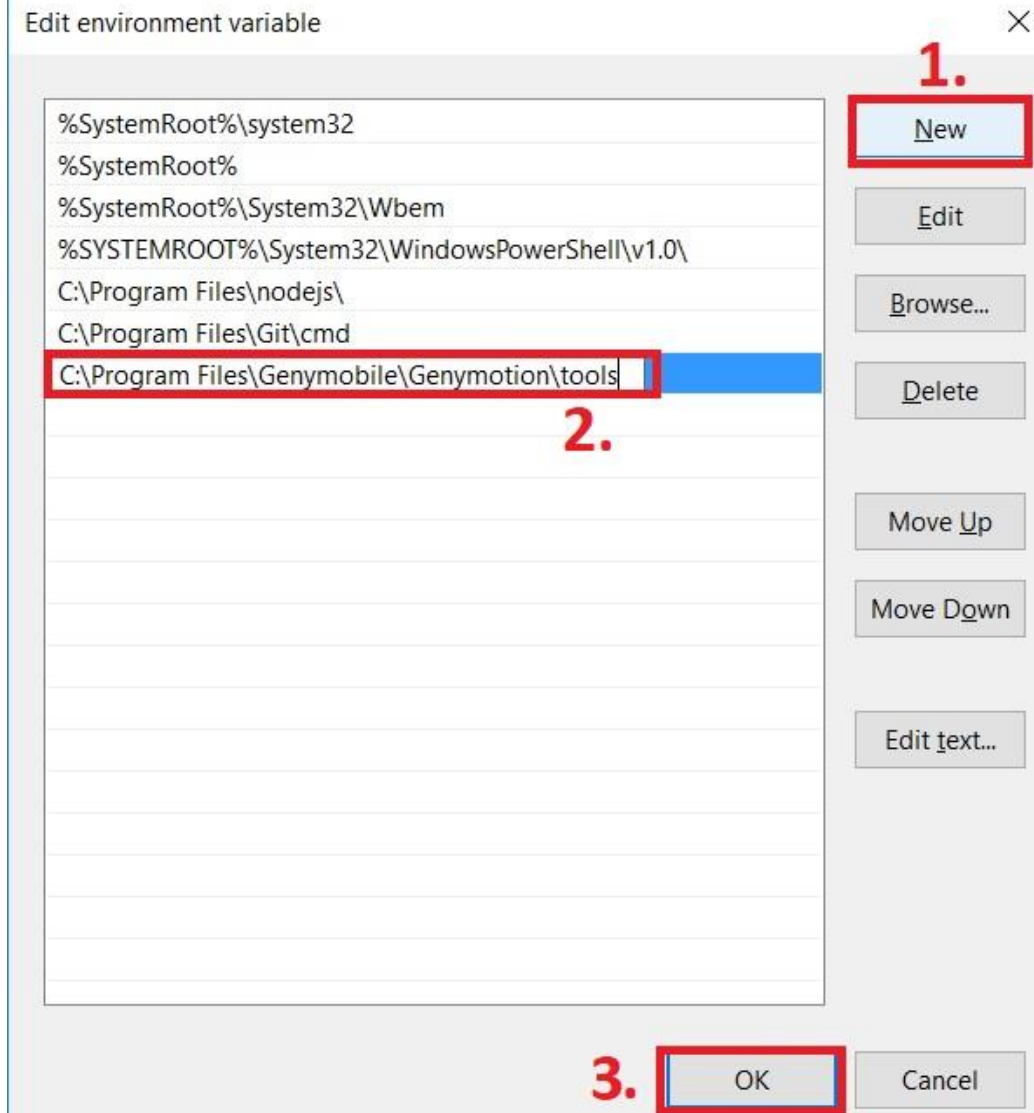
# #3 Environment Variables

- Right click **This PC (My Computer) > Properties > Advanced system**. Next click on **Environment Variables** on the bottom right. Under **System variables** bracket select **Path** and click **Edit**.





# #3 Environment Variables



Next, in the edit environment variable window, click **New**, then add path

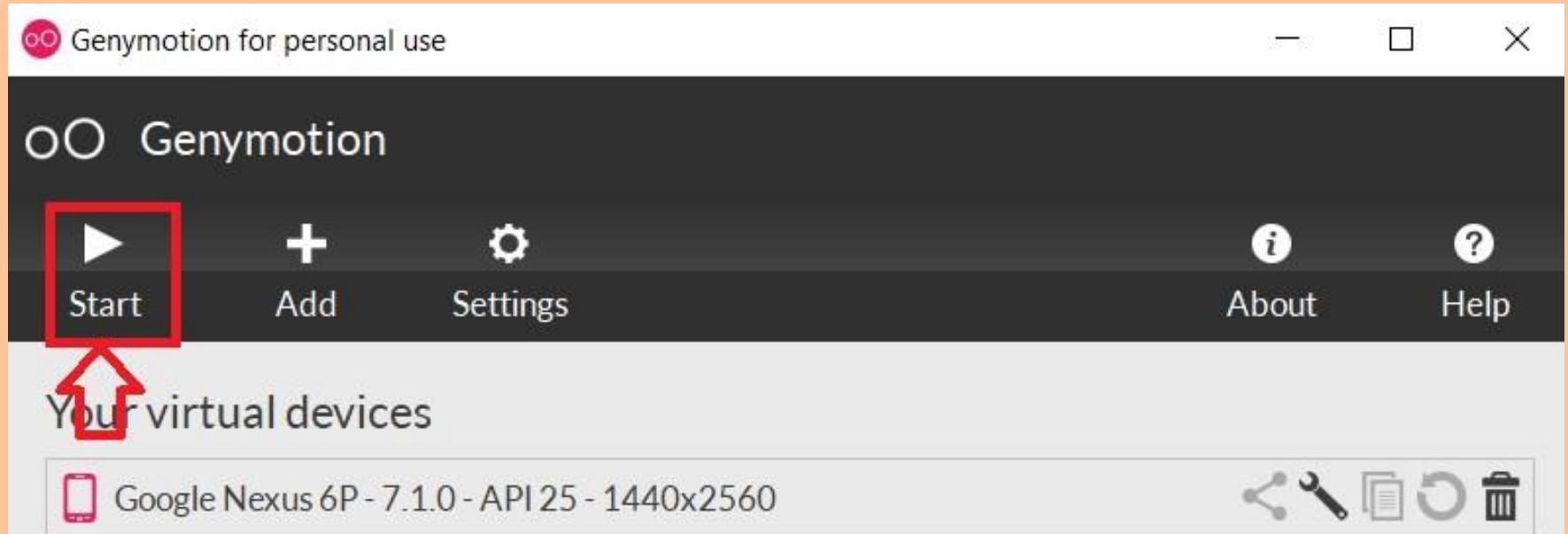
**C:\Program Files\  
Genymobile\  
Genymotion\  
tools**

And finally click **OK** until all windows close.



## #4 Starting Emulator

- Next, restore the previously minimized Genymotion window and click Start.

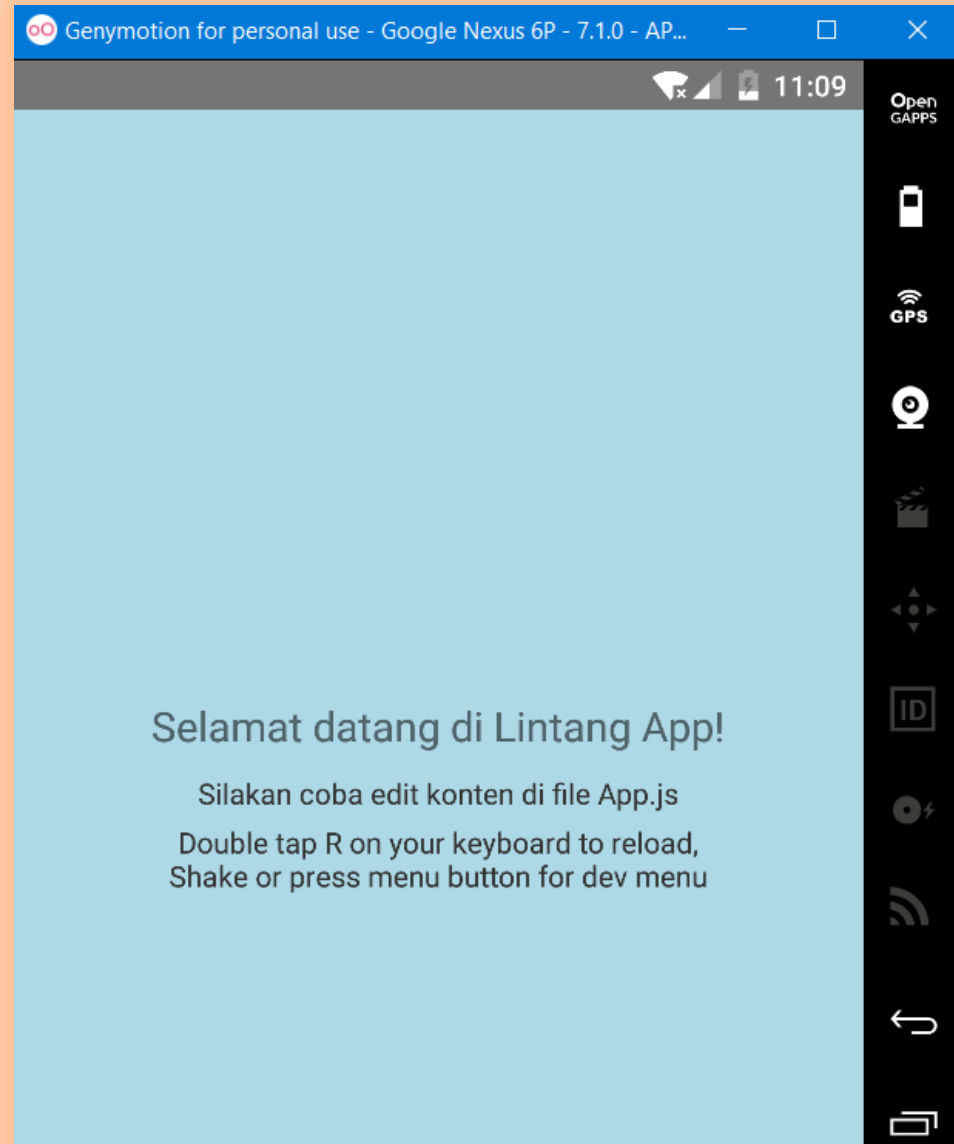


- You should see an Android emulator, then start your React Native project! Don't forget to give it permission, by enabling "Permit drawing over other apps"!



# #4 Starting Emulator

- Press **CTRL+M** to show main menu on our emulator.



Reload

Debug JS Remotely

Enable Live Reload

Enable Hot Reloading

Toggle Inspector

Show Perf Monitor

Start/Stop Sampling Profiler

Dev Settings

Mobile Development



# React Native

**#1** Getting Started