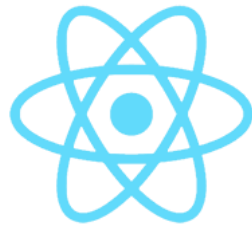


Front-End Development



React Native






#1 Getting Started



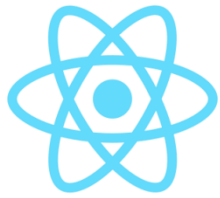
React

- ***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		React Native A framework for building native apps with React.	+15.6k ★
2		Weex A framework for building Mobile cross-platform UI.	+6.5k ★
3		Ionic Build amazing native and progressive web apps ...	+5.5k ★
4		Quasar Quasar Framework	+3.7k ★
5		NativeScript 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:

- ❖ **Android Studio**
- ❖ **Android SDK**
- ❖ **ANDROID_HOME**
- ❖ **JAVA_HOME**
- ❖ **Platform-tools**

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

Android SDK

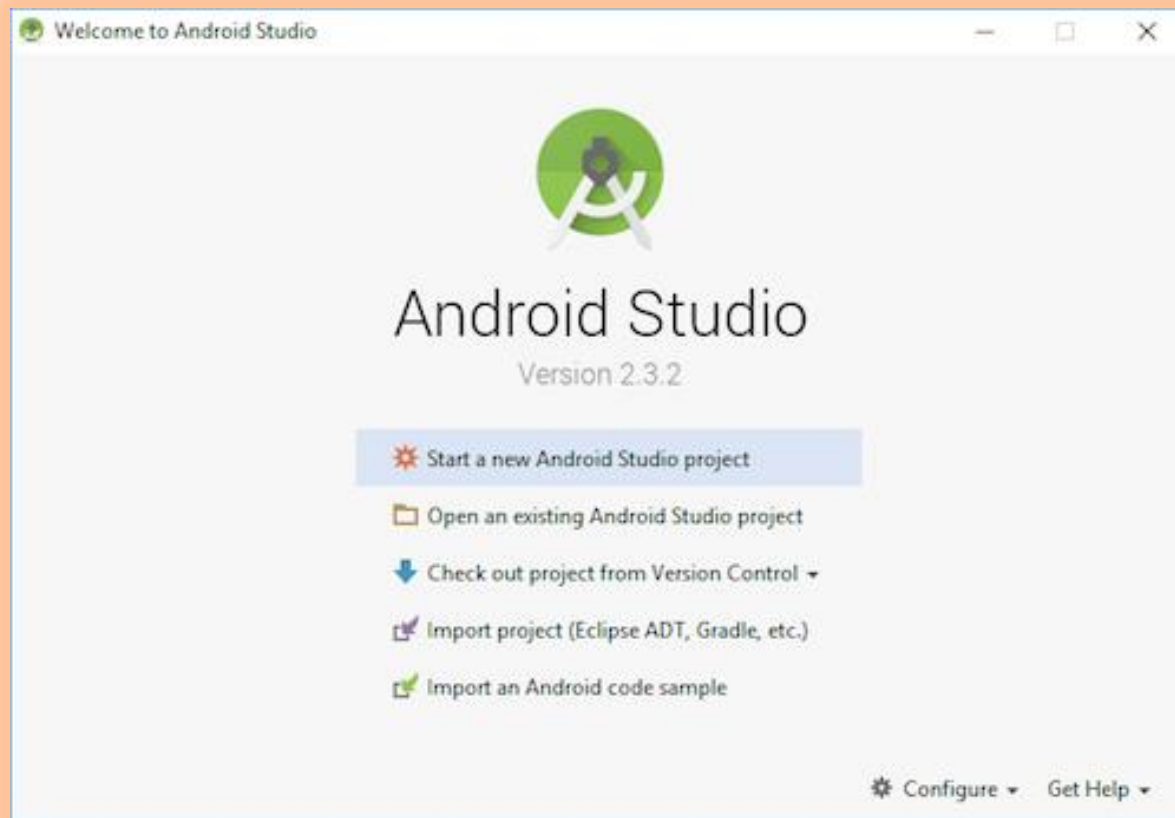
Android SDK Platform

Performance (Intel® HAXM)

Android Virtual Device

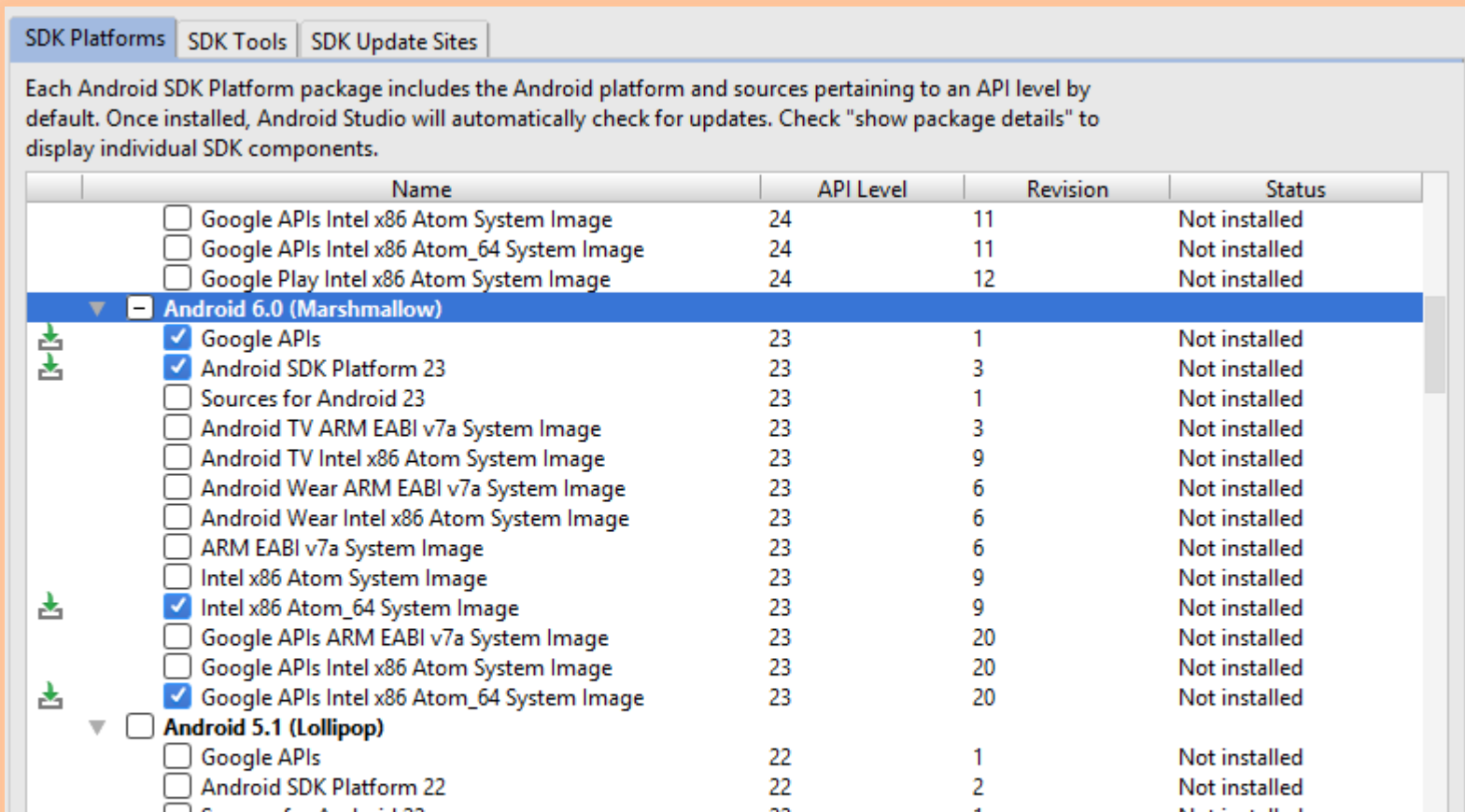
#2 Android SDK

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



#2 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:

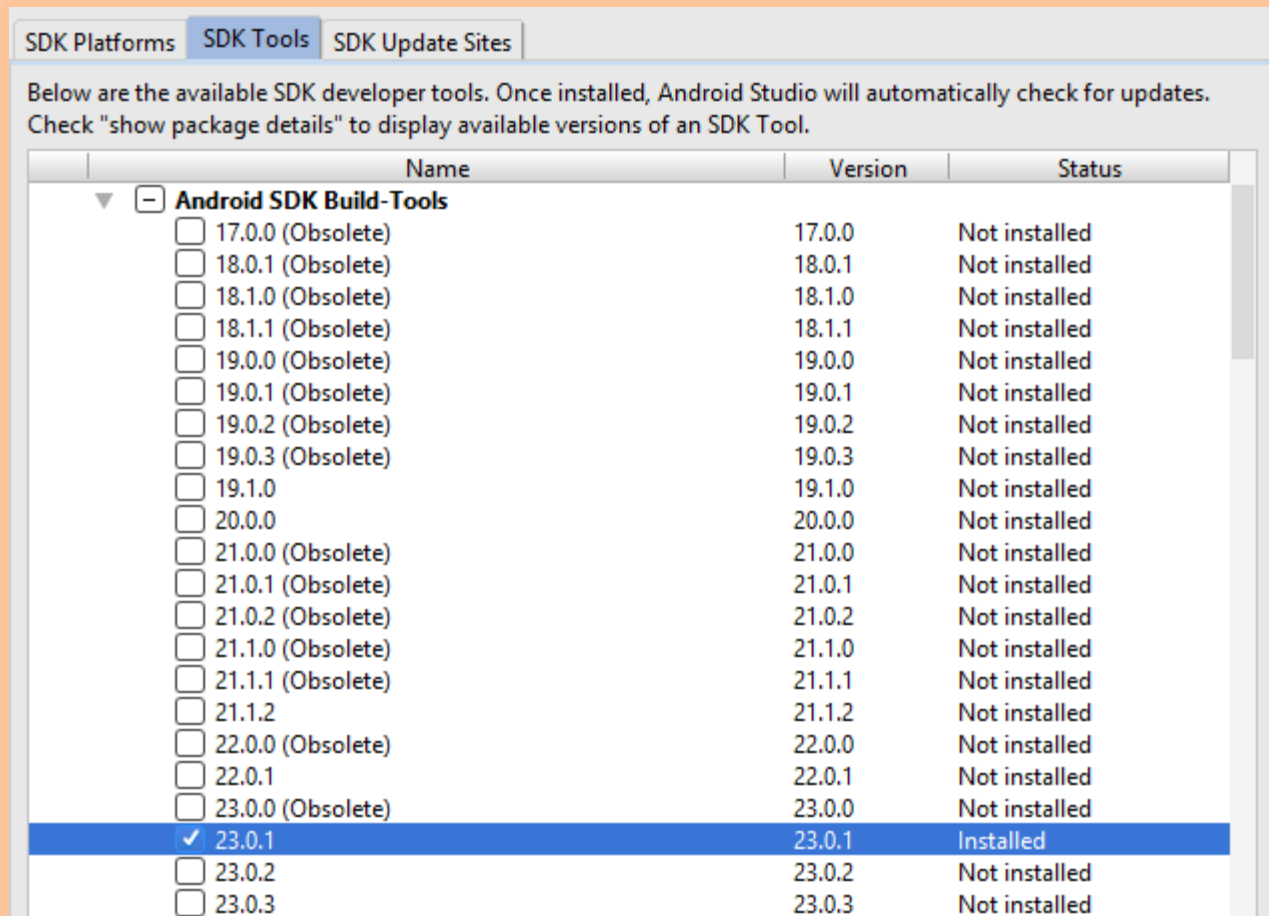


Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	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
▼	Android 6.0 (Marshmallow)			
<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
▼	Android 5.1 (Lollipop)			
<input type="checkbox"/>	Google APIs	22	1	Not installed
<input type="checkbox"/>	Android SDK Platform 22	22	2	Not installed
<input type="checkbox"/>	Sources for Android 22	22	1	Not installed

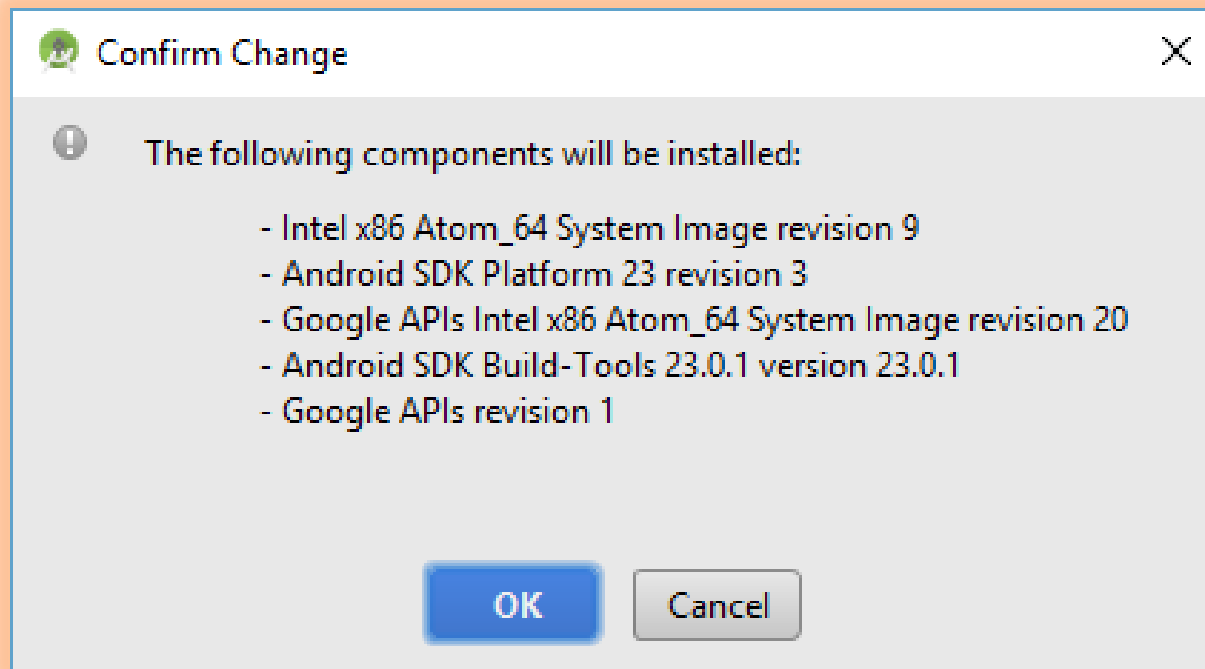
#2 Android SDK

- 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.



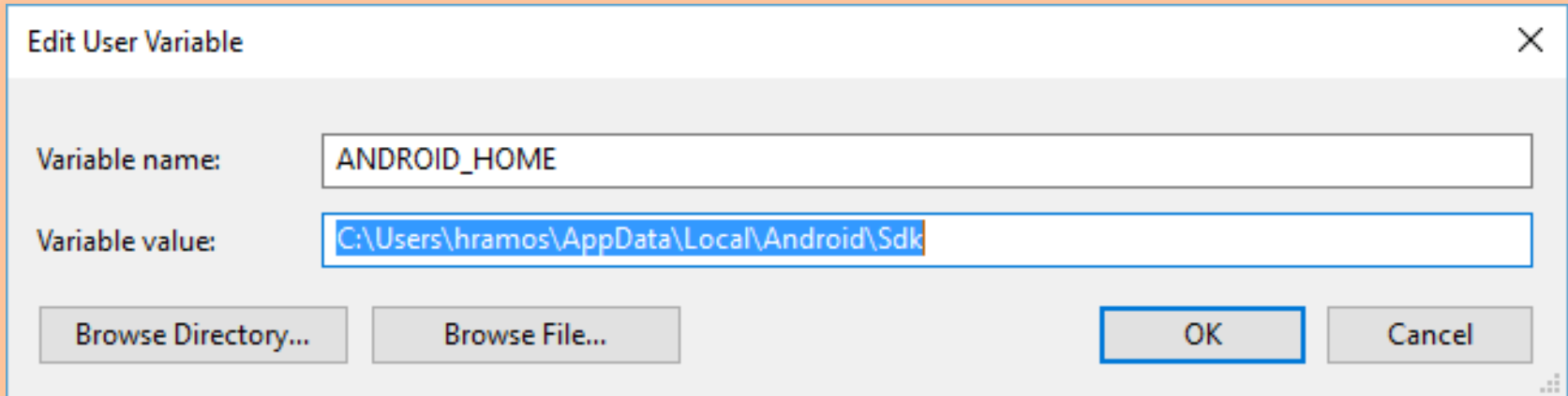
#2 Android SDK

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



#3 ANDROID_HOME

- Open the **System** pane under **System and Security** in the **Control Panel**, then click on **Change settings....** Open the **Advanced** tab and click on **Environment 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

#3 ANDROID_HOME

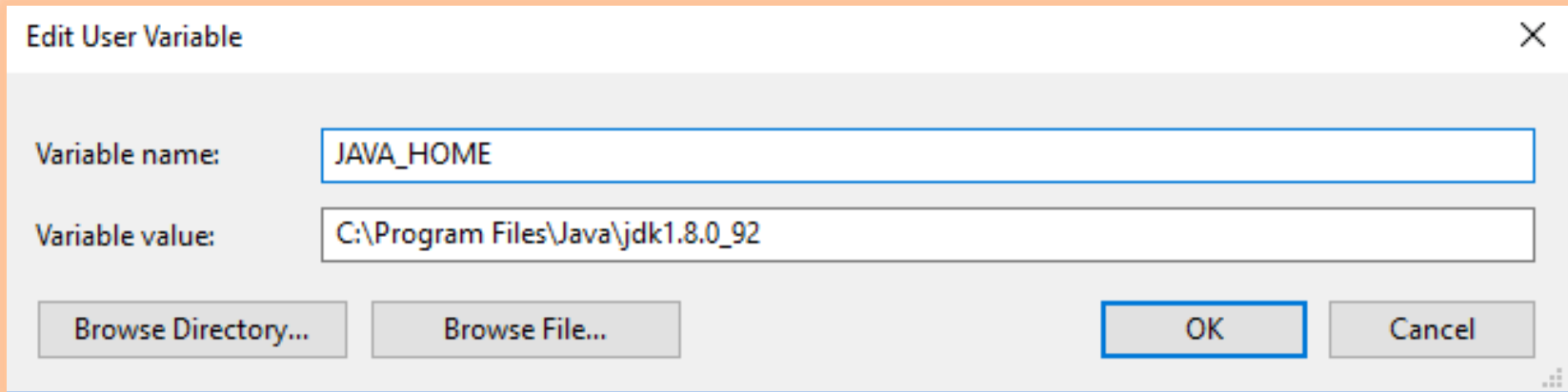
- The SDK is installed, by default, at the following location:

`C:\Users\name\AppData\Local\Android\Sdk`

- You can find the actual location of the SDK in the Android Studio **"Preferences"** 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.

#4 JAVA_HOME

- Open the **System** pane under **System and Security** in the **Control Panel**, then click on **Change settings....** Open the **Advanced** tab and click on **Environment Variables....** Click on **New...** to create a new **JAVA_HOME** user variable that points to the path to your JDK:



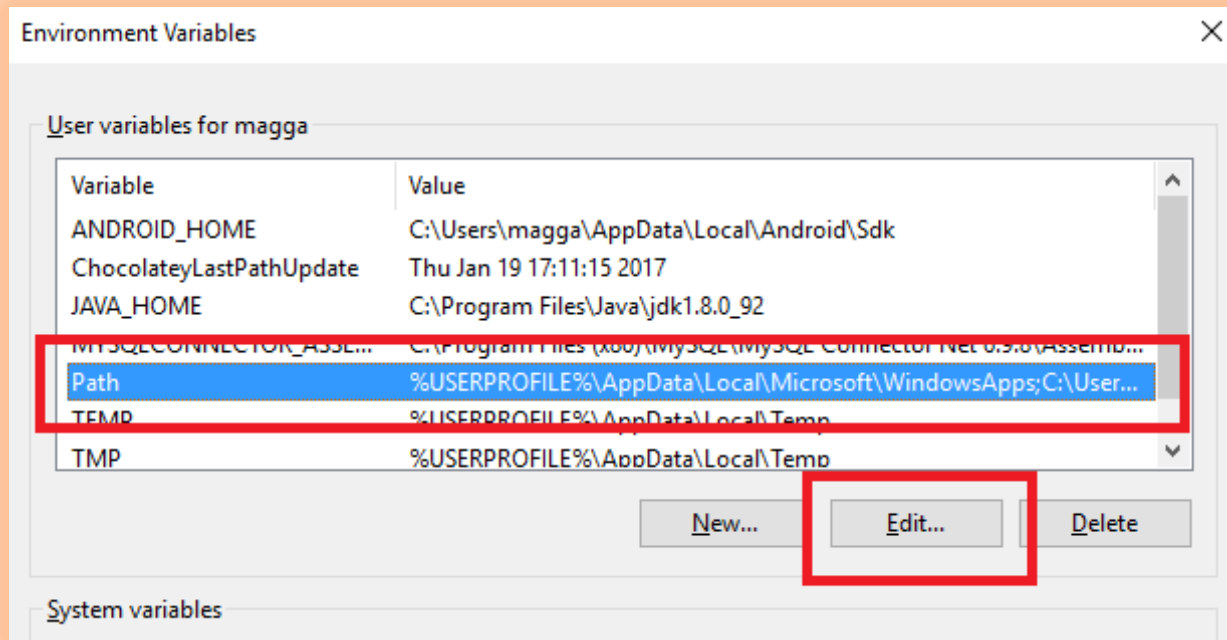
Edit User Variable

Variable name:

Variable value:

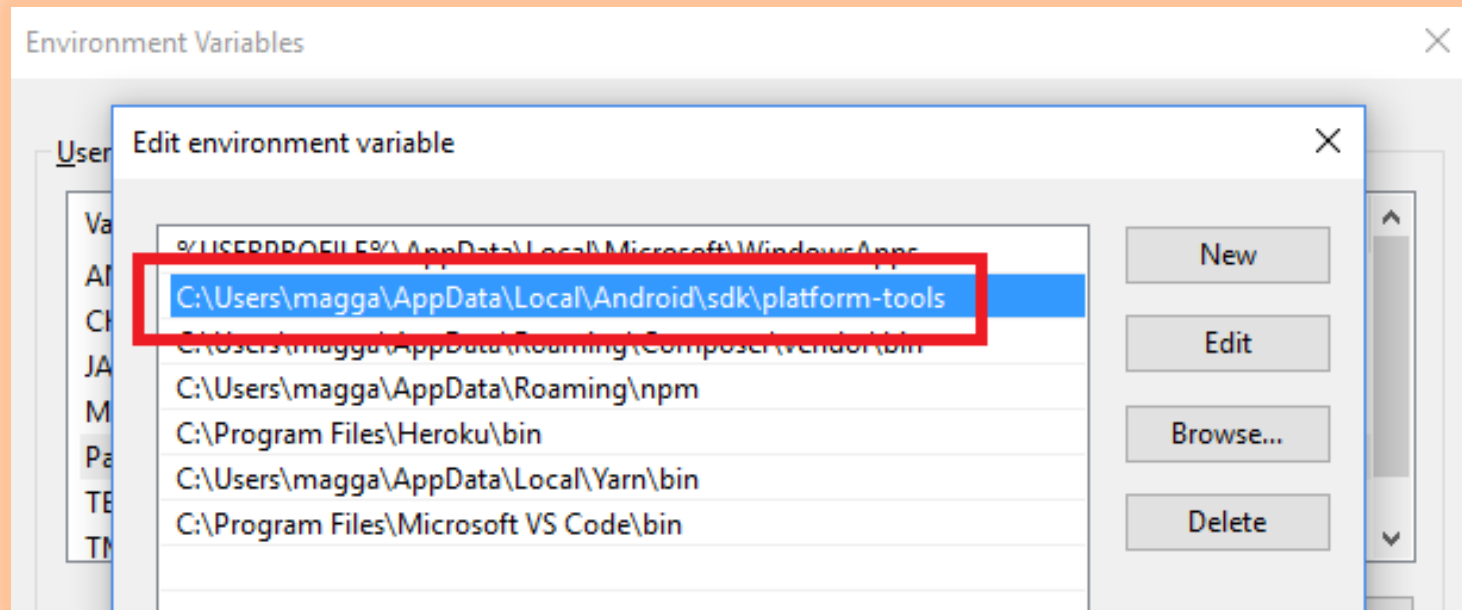
#5 platform_tools

- Open the **System** pane under **System and Security** in the **Control Panel**, then click on **Change settings....** Open the **Advanced** tab and click on **Environment Variables....** Click on **Path** then **Edit**.



#5 platform_tools

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



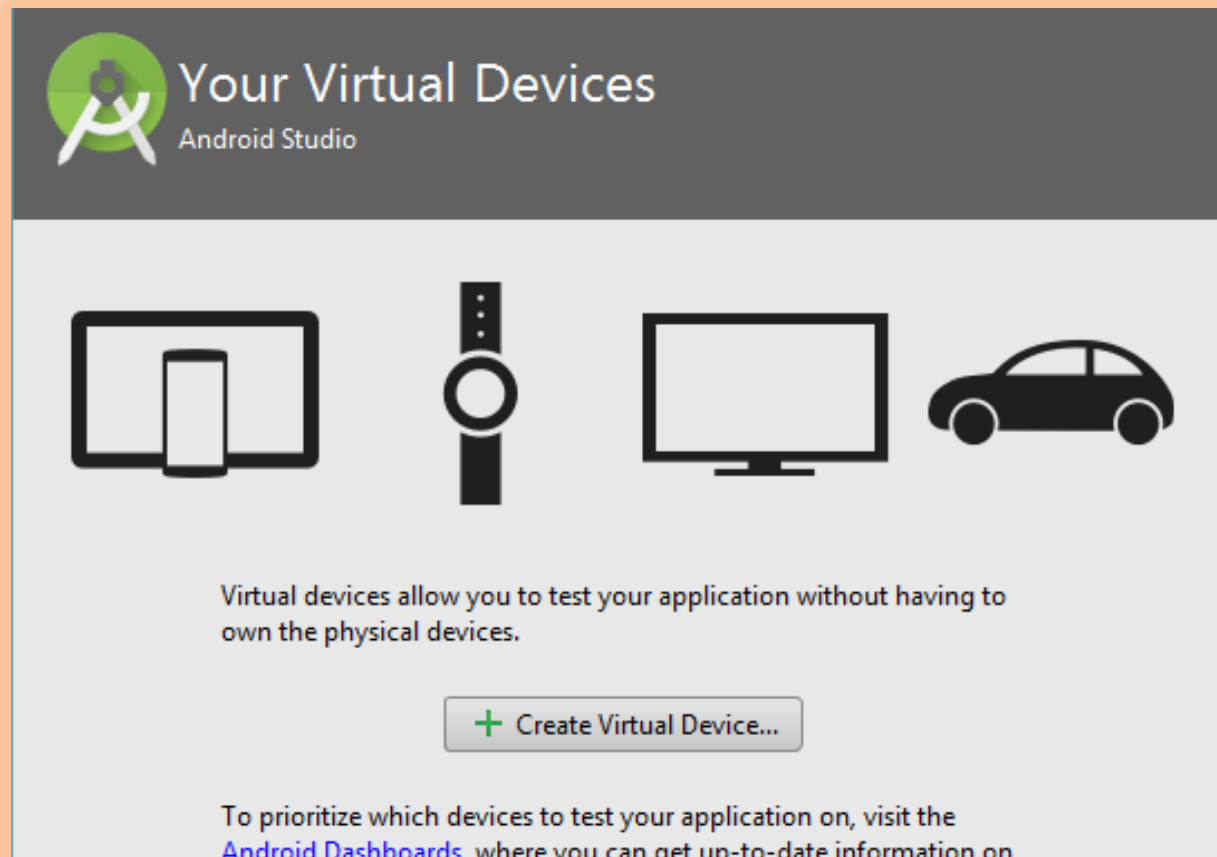
React Native Setup:

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




#1 Android Virtual Device

- 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

Recommended

x86 Images

Other Images

Release Name	API Level	ABI	Target
Nougat	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	25	x86_64	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Nougat Download	24	x86_64	Android 7.0 (Google APIs)
Nougat Download	24	x86_64	Android 7.0
Nougat Download	24	x86	Android 7.0
Marshmallow	23	x86_64	Android 6.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Marshmallow	23	x86_64	Android 6.0
Marshmallow Download	23	x86	Android 6.0
Lollipop Download	22	x86	Android 5.1 (Google APIs)
Lollipop Download	22	x86_64	Android 5.1 (Google APIs)
Lollipop Download	22	x86_64	Android 5.1
Lollipop Download	22	x86	Android 5.1
Lollipop Download	21	x86_64	Android 5.0 (Google APIs)



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 then run it:

```
$ react-native init andro_lin  
$ cd andro_lin  
$ react-native run-android
```

```
// for next (if using Android Studio) just  
$ npm start
```



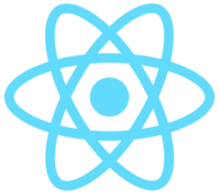
#3 local.properties

- If error when running the project, you have to create a local.properties file on your project_directory/android! Fill it with your Android SDK path, for instance:

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

- Then run again.

```
$ 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

A screenshot of the GenyMotion website. The header includes the GenyMotion logo, navigation links (Solutions, Pricing, Help, Resources, Other Products), and buttons for "Contact Us" and "Trial". The main content area is titled "Download Genymotion 2.11.0". It features two columns. The left column has a red box around the "with VirtualBox:" section, which includes a download icon and a button labeled "Download for Windows - 152MB". Below this is the "without VirtualBox:" section with a button labeled "Download for Windows - 46M". A red arrow points to the "with VirtualBox:" section. The right column is titled "System Requirements" and lists the necessary hardware and software for running GenyMotion. At the bottom of the right column are links for "Checksum Windows (with VirtualBox)" and "Checksum Windows (without VirtualBox)".

GENYMOTION^{oo} by

Solutions ○ Pricing Help ○ Resources ○ Other Products ○

Contact Us Trial

Download Genymotion 2.11.0

with VirtualBox:

Download for Windows - 152MB

without VirtualBox:

Download for Windows - 46M

[How to register my license](#)

System Requirements

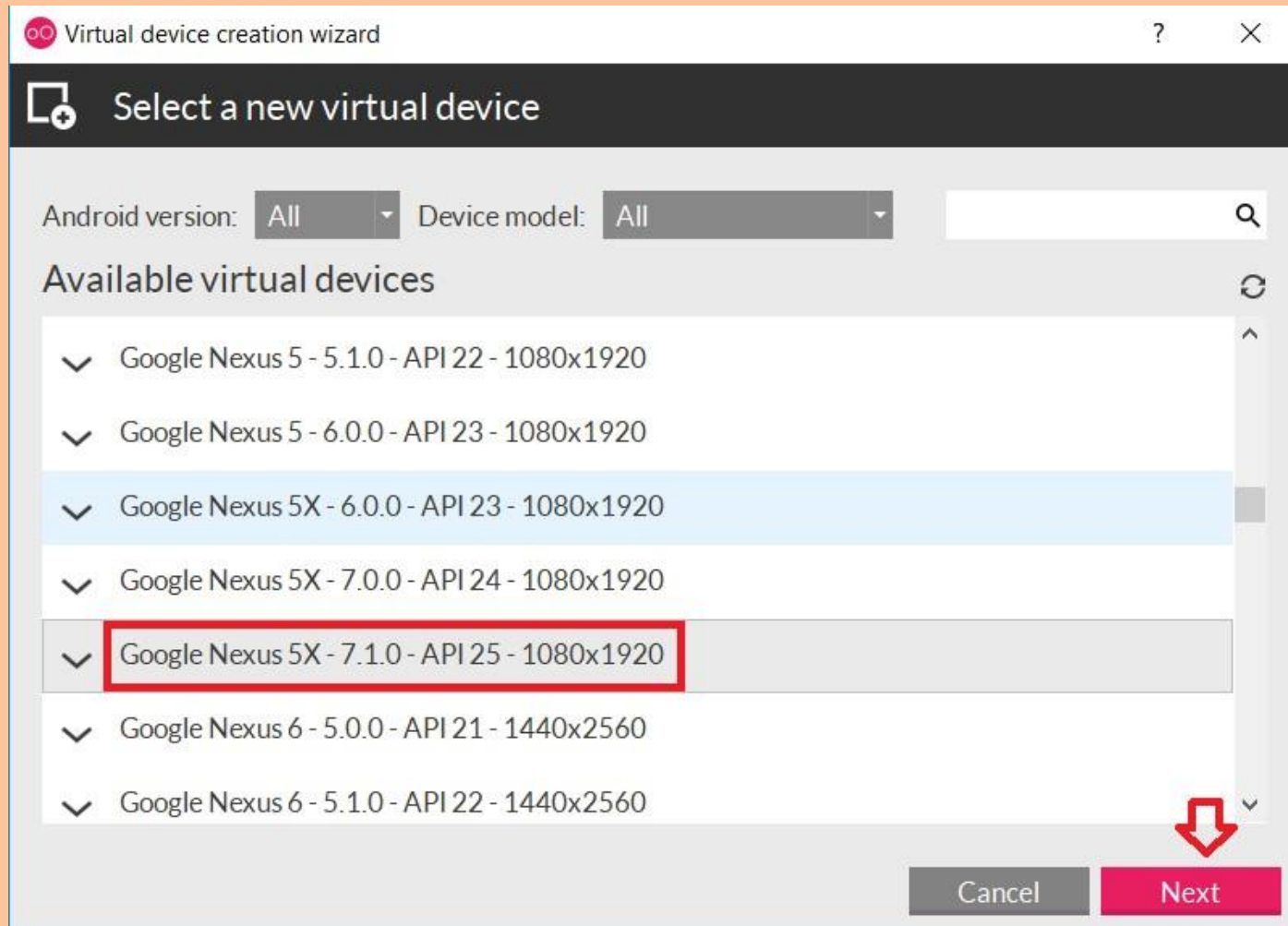
Microsoft Windows 7, 8/8.1, 10 (32/64 bit)
64 bit CPU, with VT-x or AMD-V capability, enabled
in BIOS settings
Recent and dedicated GPU
400 MB disk space
2GB RAM

[Checksum Windows \(with VirtualBox\)](#)
[Checksum Windows \(without VirtualBox\)](#)



#1 Add Virtual Device

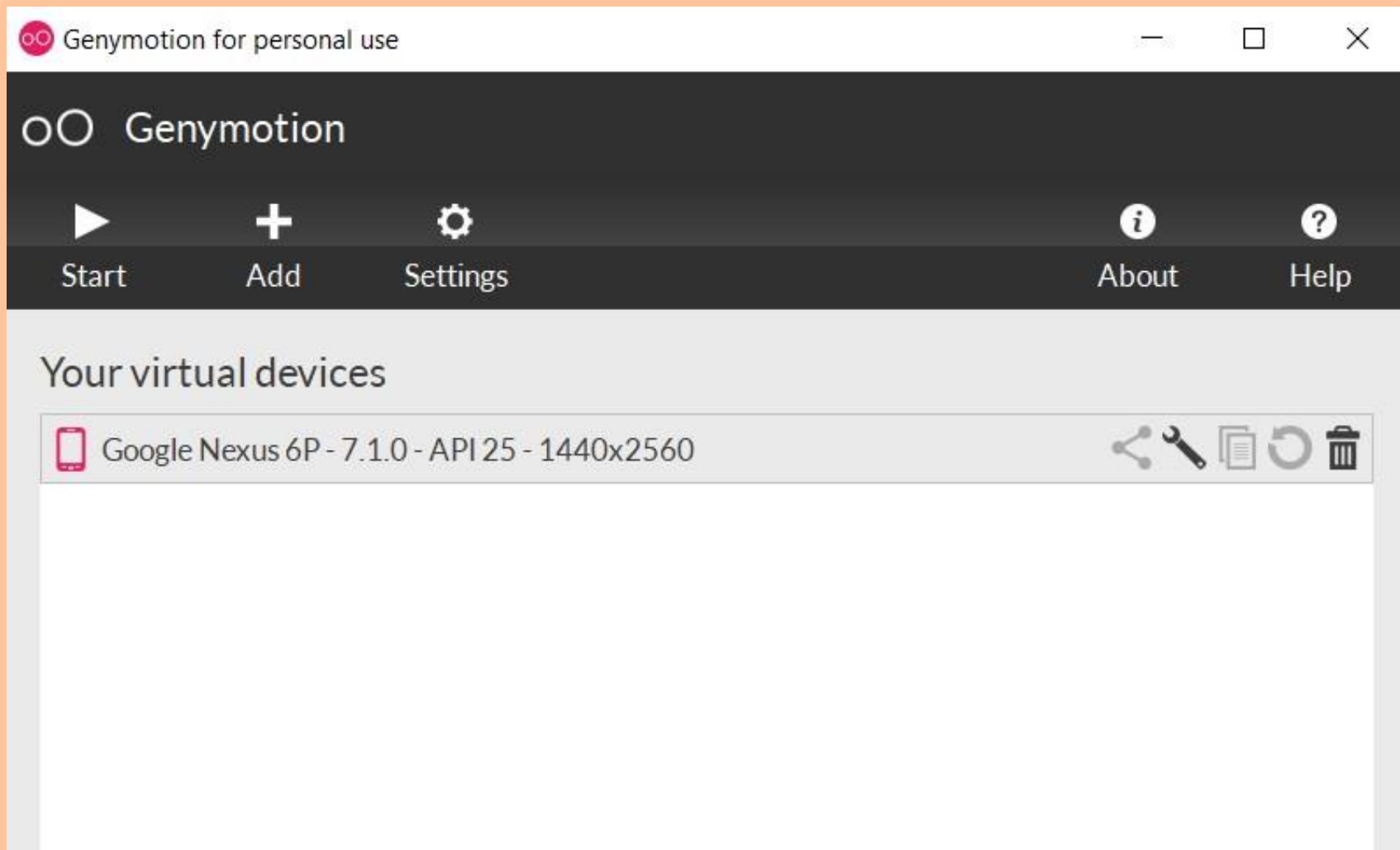
- Once downloaded, open Genymotion and login with your account. Now, you'll be able to add a new Virtual Device (Ctrl+N).





#1 Add Virtual Device

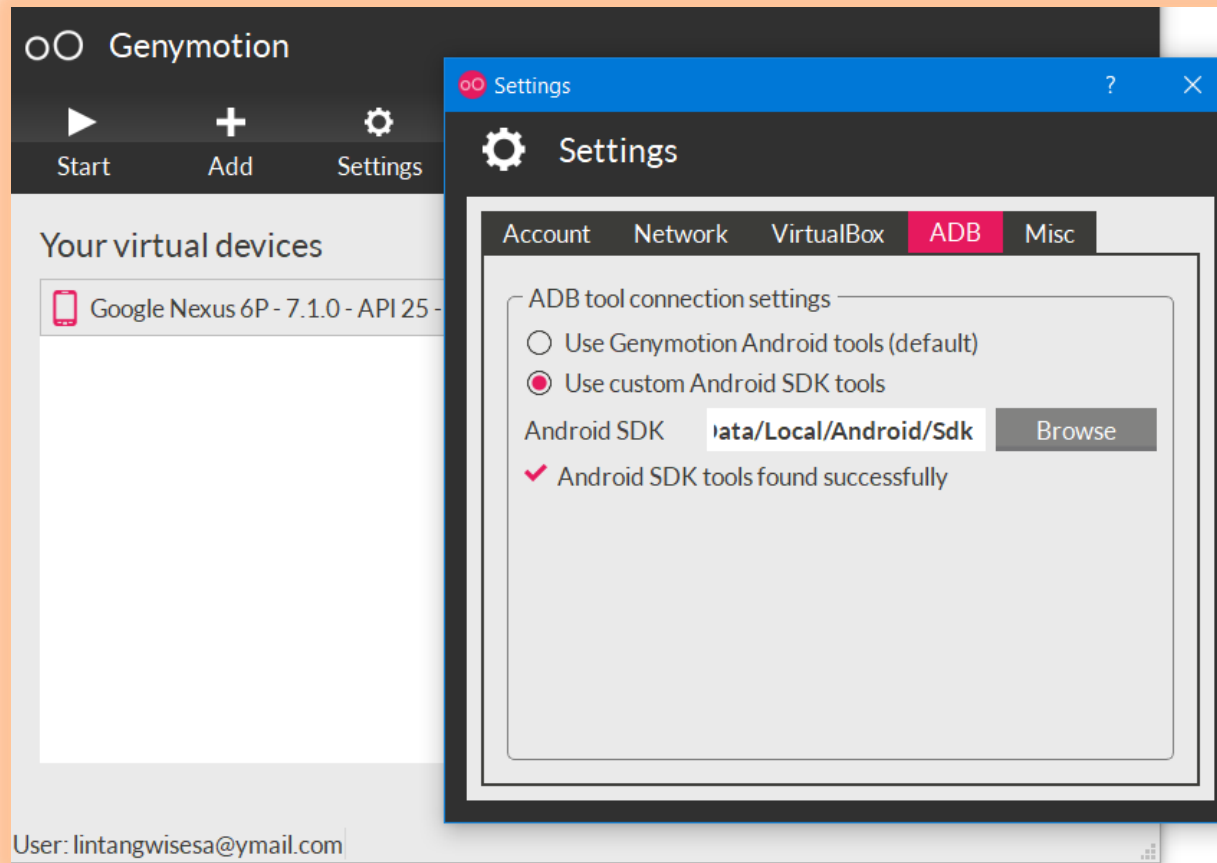
- Once done you'll see your devices listed under Virtual devices:





#2 ADB to SDK

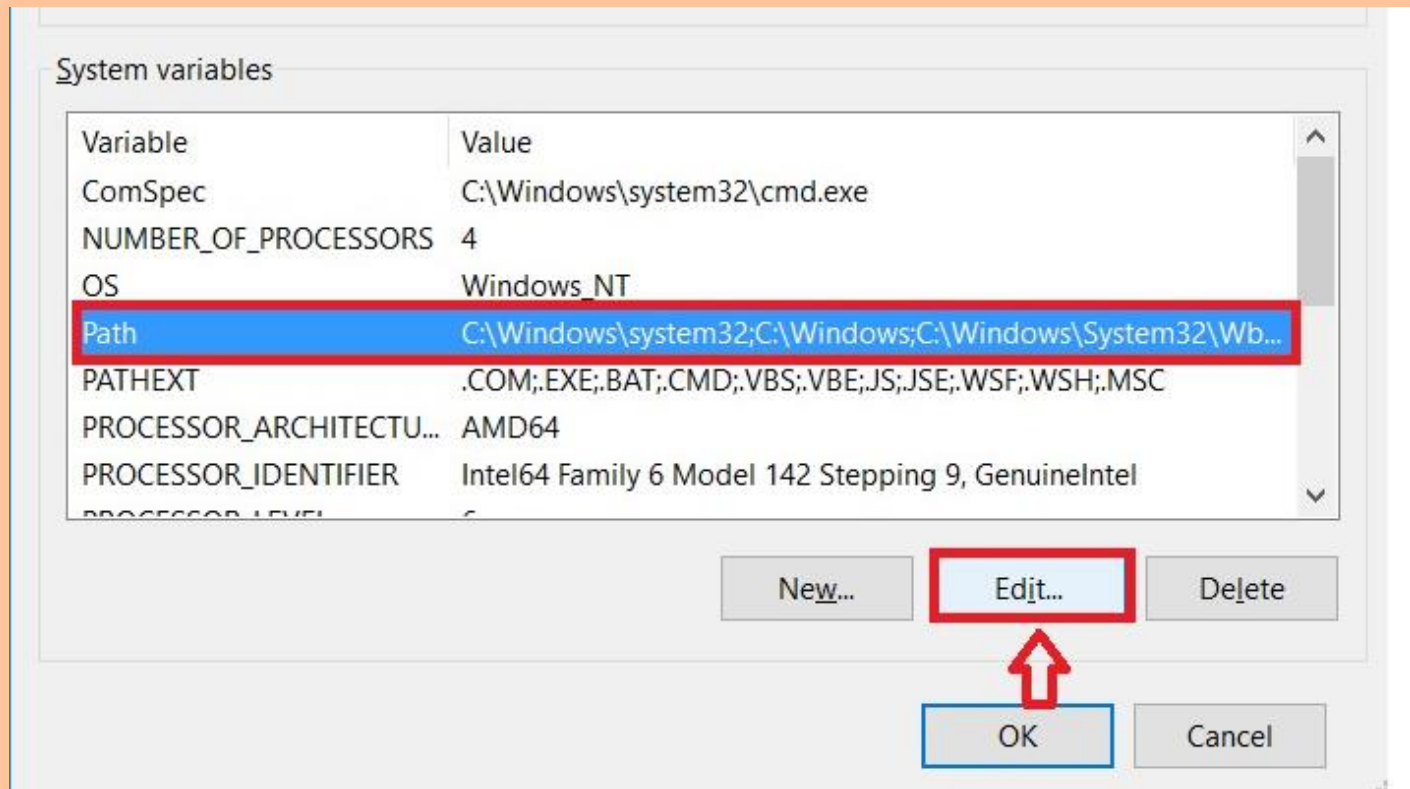
- On Genymotion, click **Settings**, on tab **ADB** choose **Use custom Android SDK tools** then browse your Android SDK's path, e.g:
`C:\Users\name\AppData\Local\Android\Sdk`





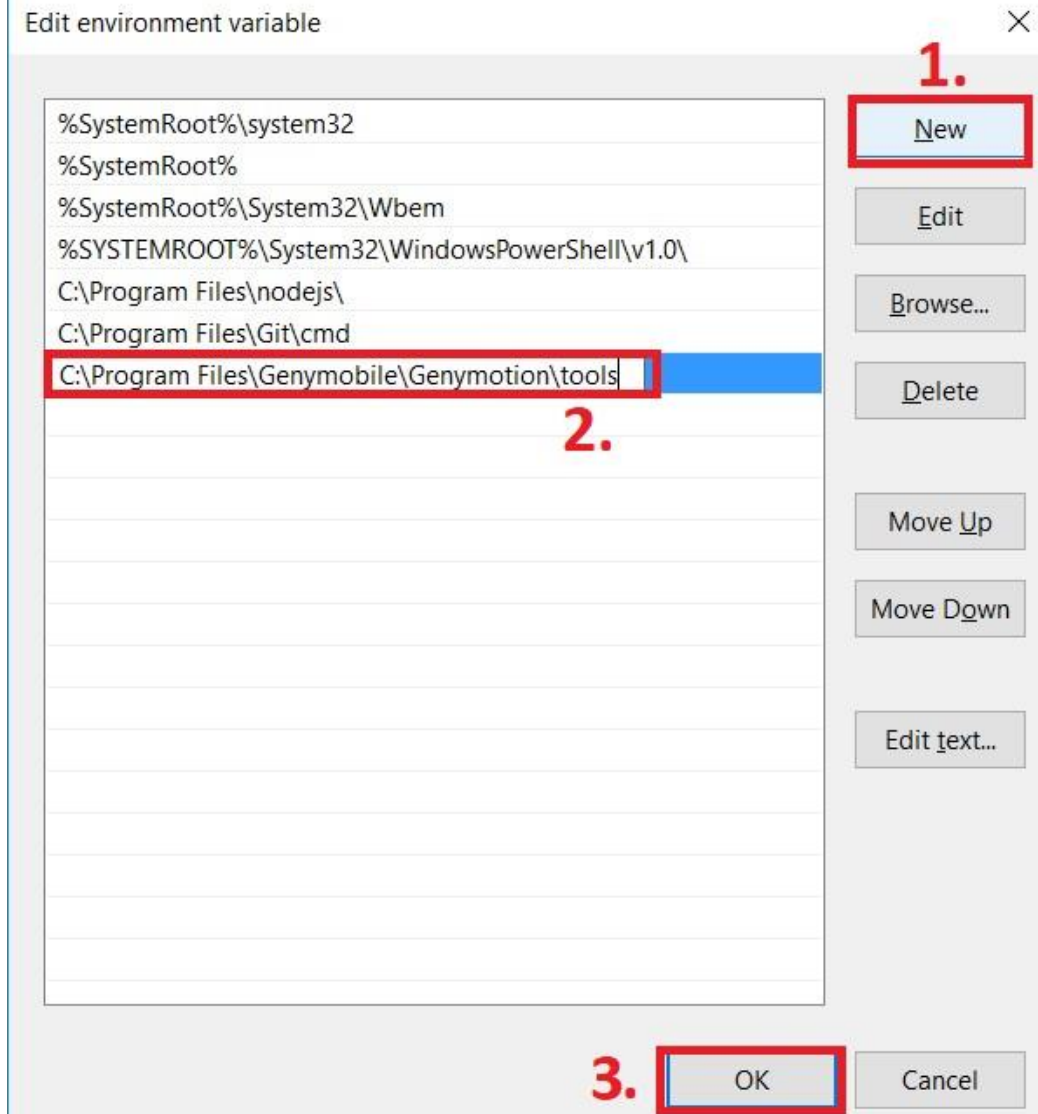
#3 Environment Variables

- 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

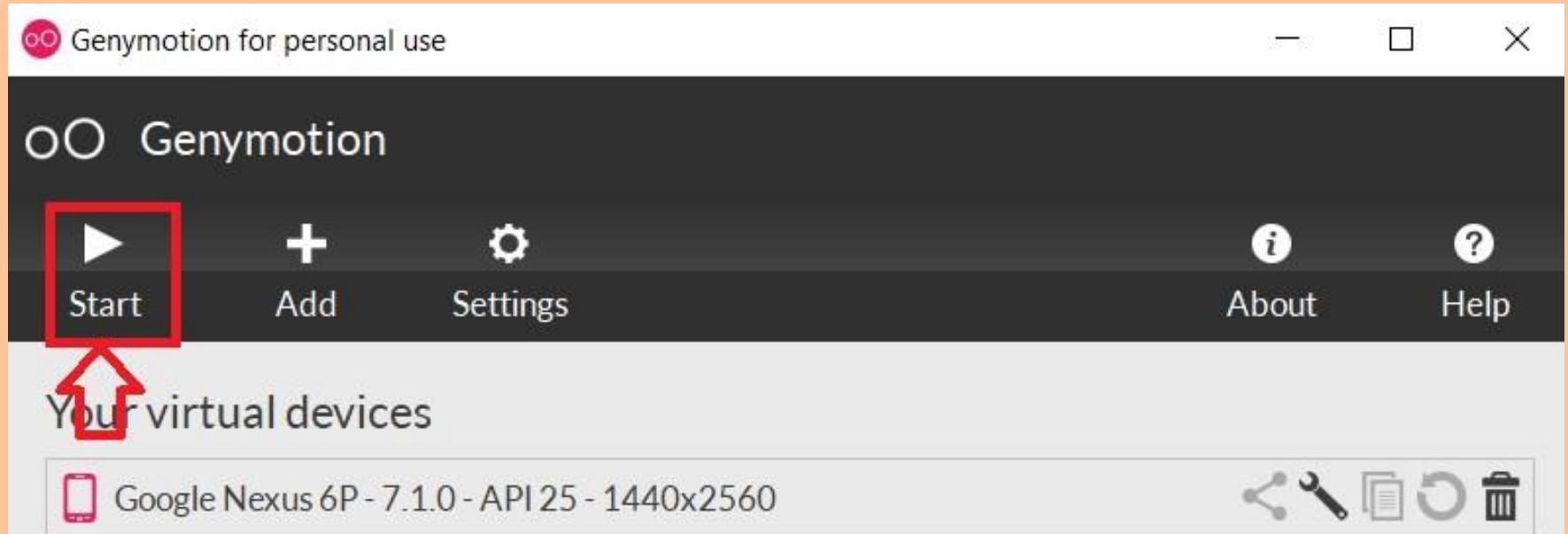
**C:\Program Files\
Genymobile\
Genymotion\
tools**

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



#4 Starting Emulator

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

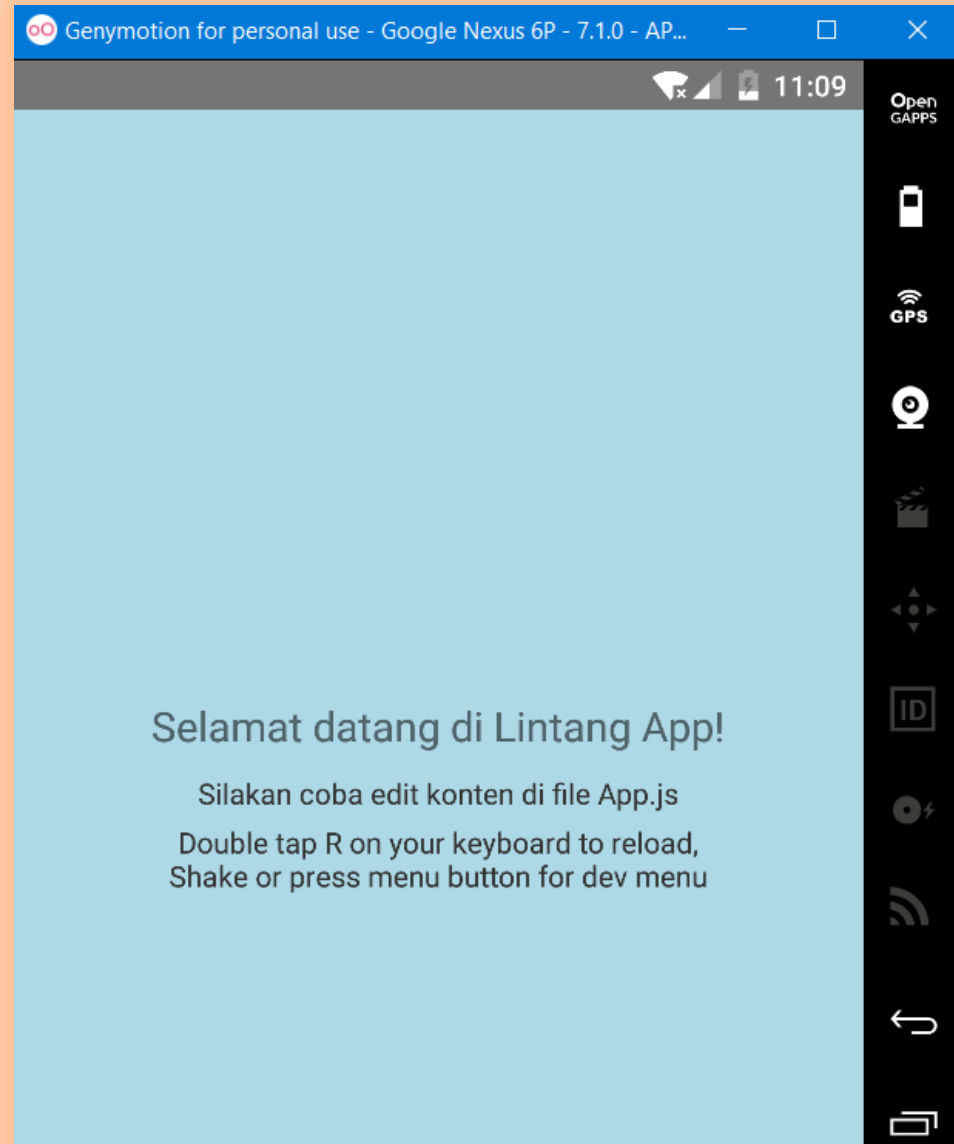


- You should see an Android emulator, then start your React Native project! Don't forget to give it permission, by enabling "Permit drawing over other apps"!



#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

Front-End Development



React Native

#1 Getting Started