# **Workshop Setup Tutorial**

# **Android Studio Installation**

### **Step 1**

[Download android studio](https://developer.android.com/studio/)

Click download Android Studio and agree to the terms and conditions.

### 

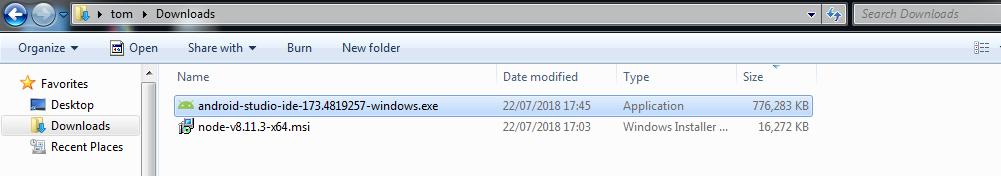
### 

### 

### **Step 2**

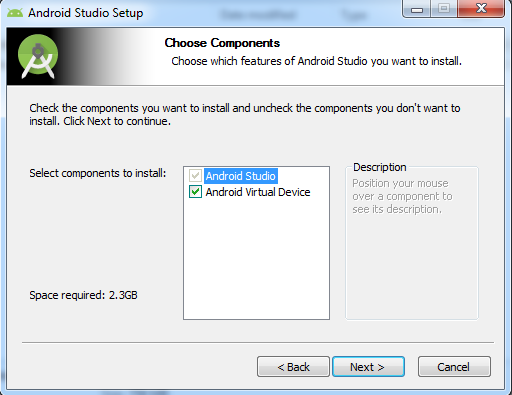
After the download is finished, navigate to the “Downloads” folder in the file explorer.

Double click on the .exe android studio file.



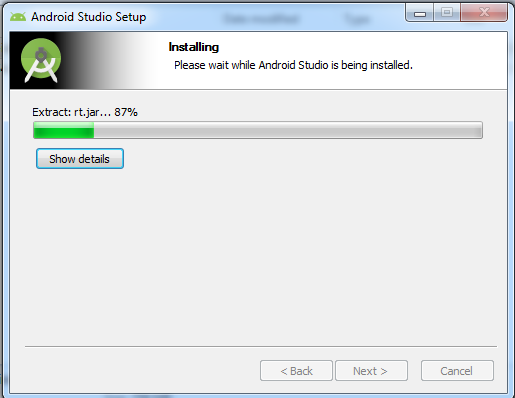
### **Step 3**

Ensure that Android Virtual Device is clicked before clicking next.



### **Step 4**

Wait for installation to complete.



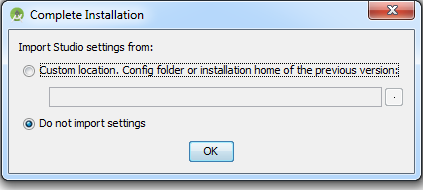
### **Step 5**

Android Studio has finished installing.



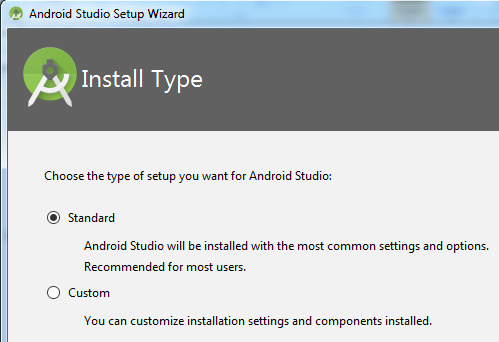
### **Step 6**

Click on the option “do not import settings” and then click ok.



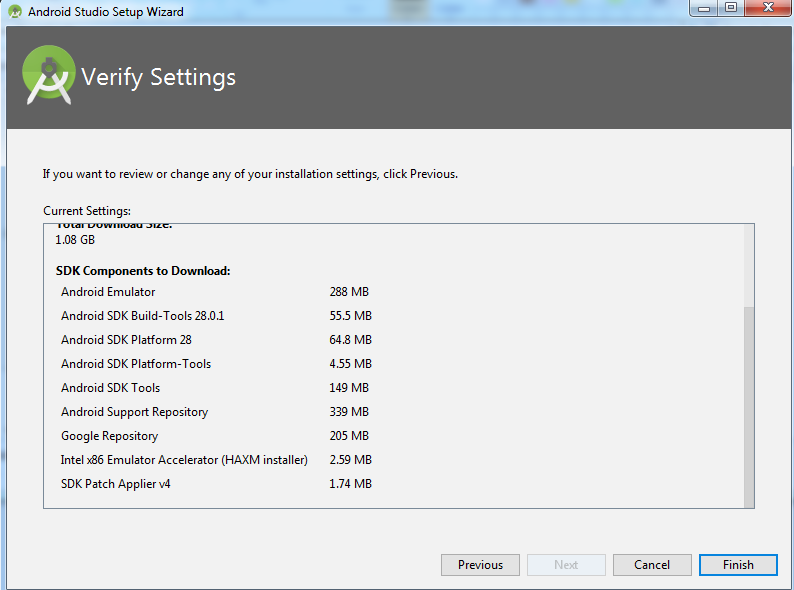
### **Step 7**

Click the standard option.



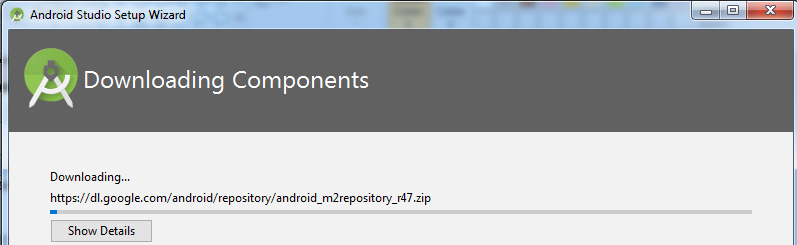
### **Step 8**

Click finish.



### **Step 9**

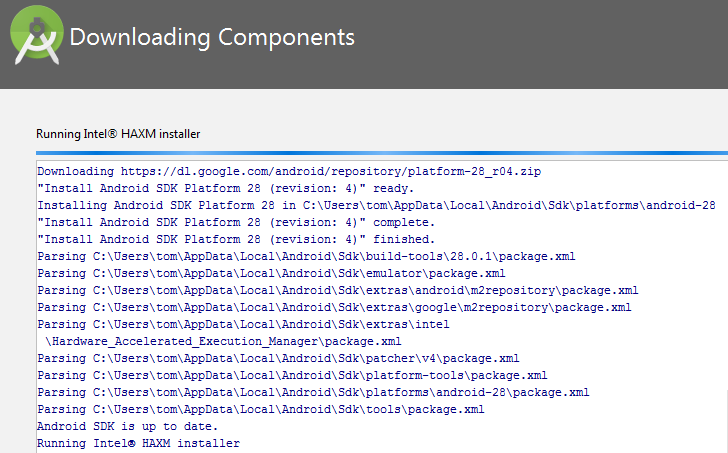
This stage will take 10-20 minutes depending on your internet connection.



### 

### **Step 10**

When your download is nearly finished, you will get a windows prompt “confirm changes to computer”, click ok.



### **Step 11**

Successful installation.

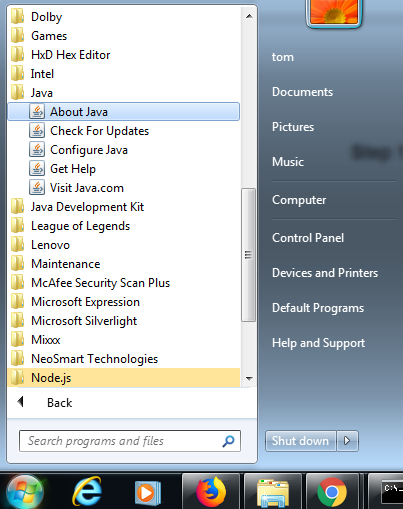


### **Step 12**

At this stage we want to check what version of JDK we have.

Launch the windows start menu and click on programs.

Find java and click About java.



### **Step 13**

If you see the version is 7, then you will need to uninstall the JDK for version 8.

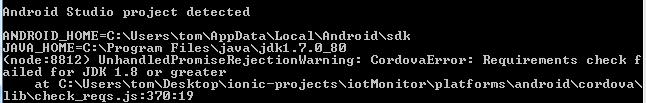


If you try to run the Ionic application on your mobile with the command

ionic cordova run android --device

You will get the following error.

“**CordrovaError: Requirements check failed for JDK 1.8 or greater**”



### 

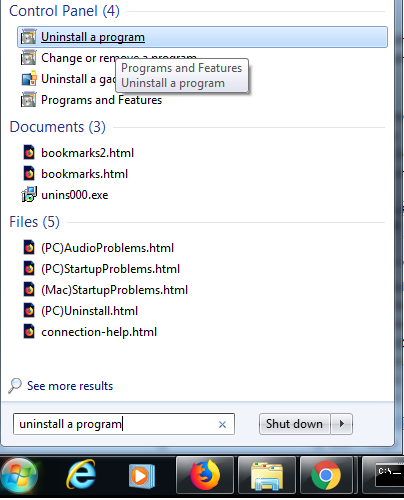
### 

### 

### **Step 14**

Warning: If you already have version 8 or “1.8” of the JDK, then you do not need to perform the following steps.

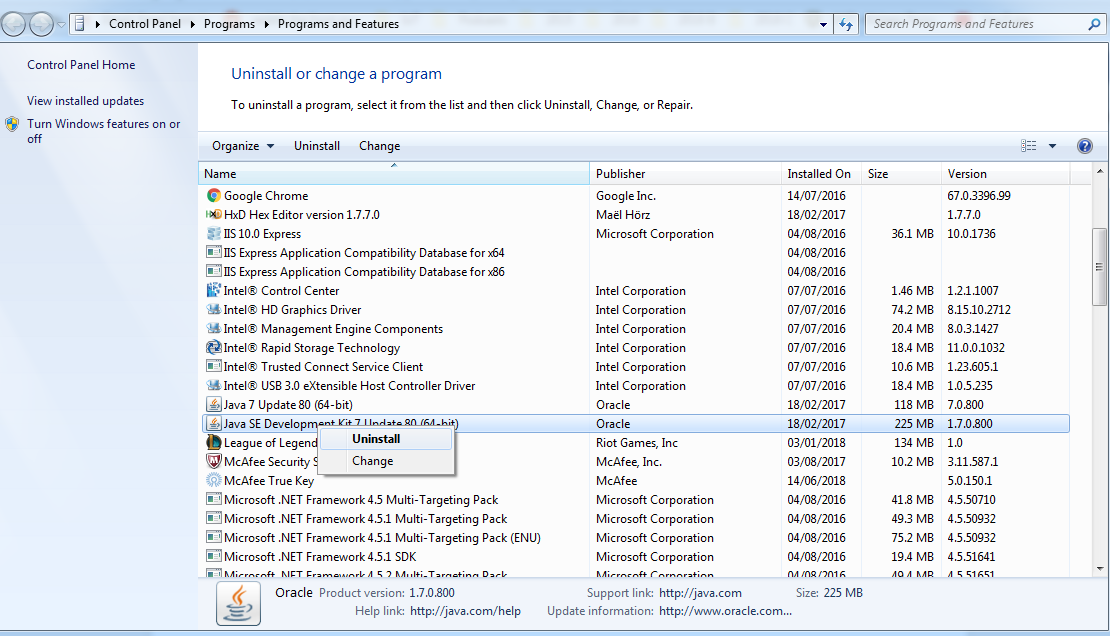
Click on the windows start button and search “uninstall a program”



### 

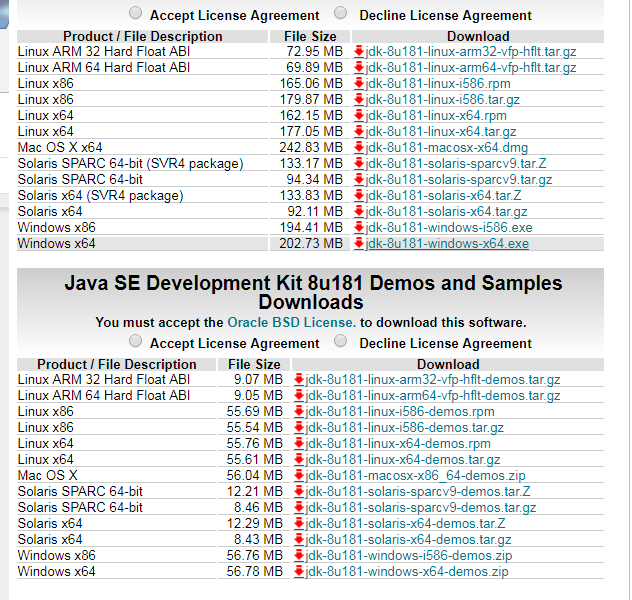
### **Step 15**

Search for java SE Development Kit 7 and click uninstall.



### **Step 16**

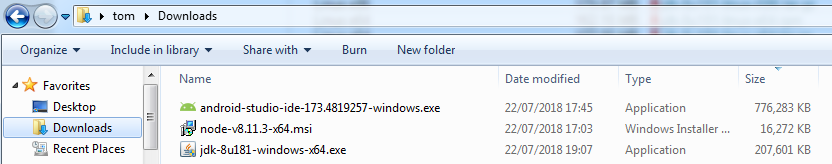
Go to the JDK [downloads page](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) and install Windows x64 of the JDK. Remember to click “accept licence agreement”.



### **Step 17**

Once the download has finished, navigate to the downloads folder using the file explorer.

Double click on the jdk-8u181.exe file.



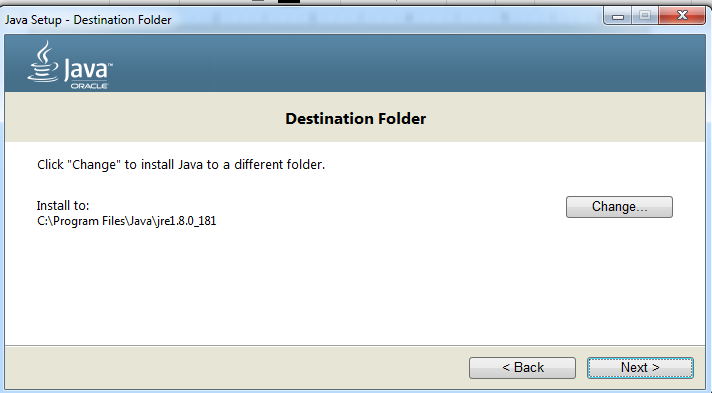
### **Step 18**

If successful, you should see this window, click ok.



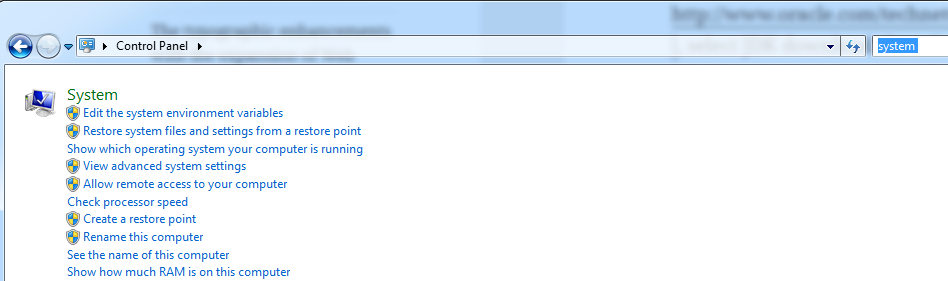
### **Step 19**

Click next.



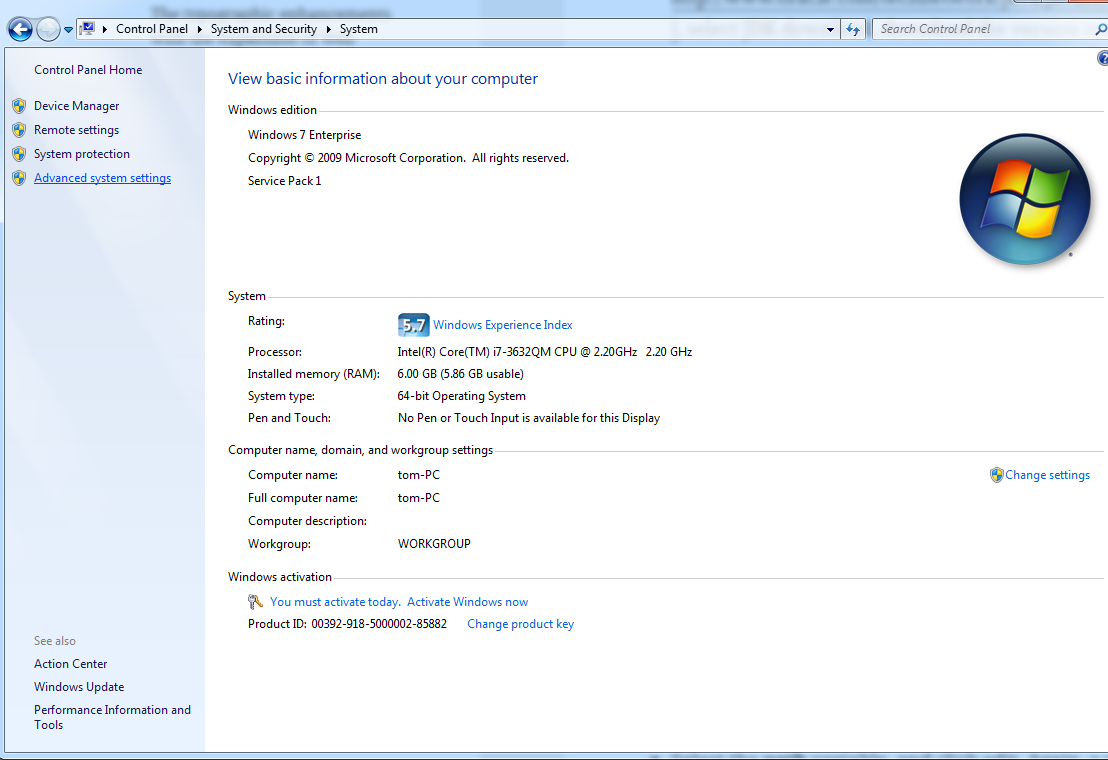
### **Step 20**

Now let’s create the **JAVA\_HOME** variable in the Windows environment. Open the “Control Panel” and search for “system”, then click on system highlighted in green.



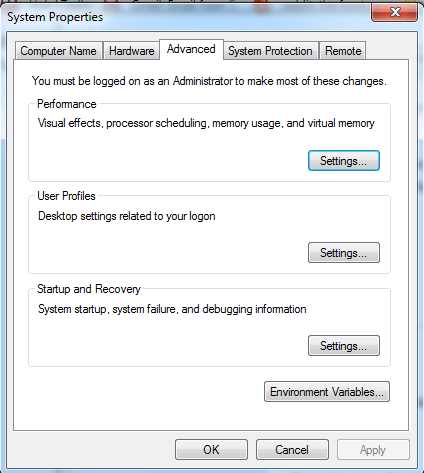
### **Step 21**

Click “Advanced System Settings”



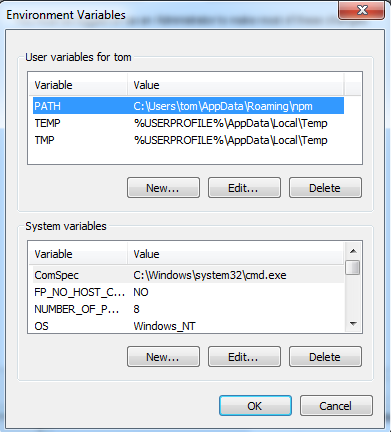
### **Step 22**

Click Environment Variables.



### **Step 23**

Click “new” under system variables.



### **Step 24**

Add the JAVA\_HOME Environment variable and give it the path to where you installed your JDK.

### 

### **Step 25**

Select the **path** variable under “system variables”, and click **edit**. Again, a window will open with the variable data. In the variable field values, go to the end of the *string* and add

**;%JAVA\_HOME%\bin**

*Do not forget the “****;****”.*

### 

### **Step 26**

Locate where the Android SDK is installed on your machine.

SDK folder by default is in

C:\Users\<user-name>\AppData\Local\Android

Using the file explorer click “Local Disk (C:)

Then click users, then followed by the user you are logged in as.

**Step 27**

You will not be able to see the AppData folder using the file explorer because it is hidden.

To get around this.

Add to the path at the top of the file explorer and hit enter.

**\AppData\Local**

### 

### 

### **Step 27**

From here you should be able to see the “Android” folder.

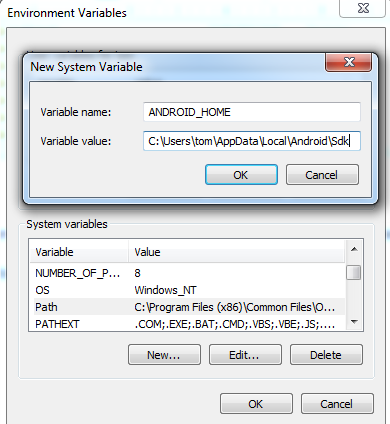
Verify that it has a folder called “sdk”inside it with the following files.

### 

### **Step 28**

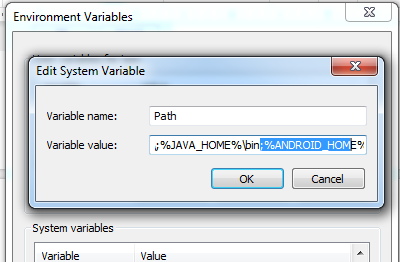
Go back to the system environment view in the control panel and click “new system environment variable”.

Name it “ANDROID\_HOME” and give it the path to your android sdk.



### **Step 28**

Returning to the environment variables, select the variable **path**, and click edit. In the new window in the variable value field, go to the end of the string and add **;%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools**

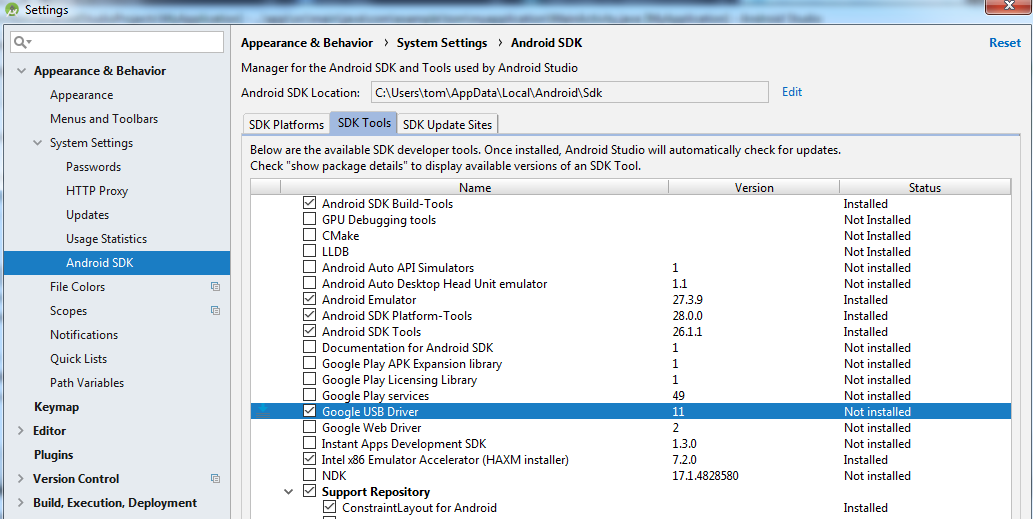


### **Step 29**

In Android studio, click file in the top left corner and then click settings.

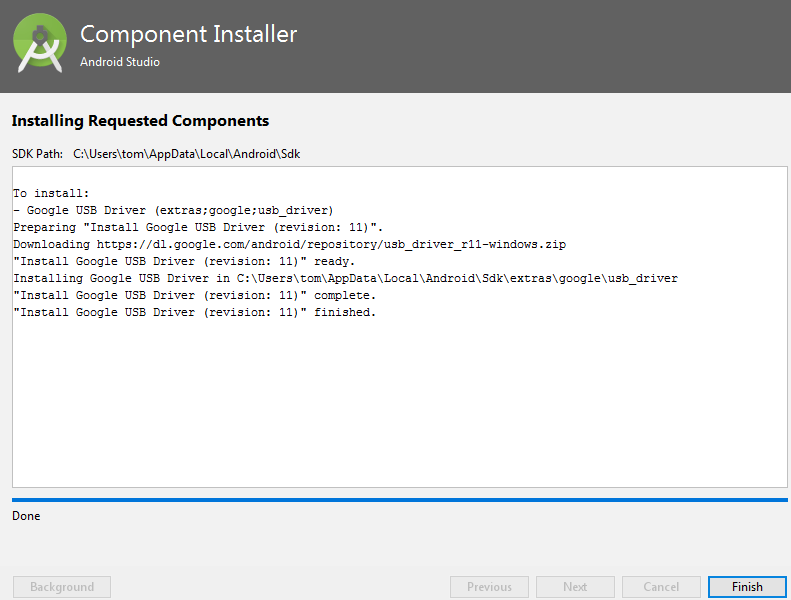
Under Appearance & Behaviour -> Android SDK

Click “Google USB Driver” and then click apply.



### 

### **Step 30**



### **Step 31**

[Install OEM USB drivers](https://developer.android.com/studio/run/oem-usb)

[Link to drivers](https://www.teamandroid.com/download-android-usb-drivers/)

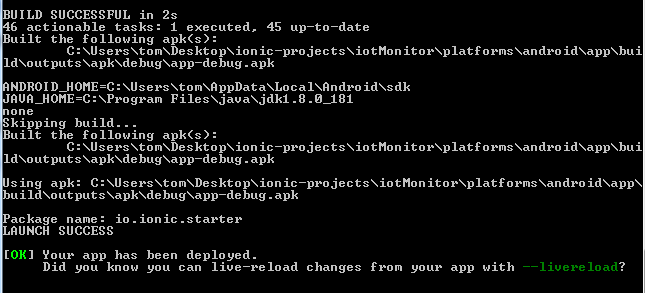
If you are developing on Windows and want to connect a device for testing, then you need to install the appropriate USB driver. This page provides links to the web sites for several original equipment manufacturers (OEMs), where you can download the appropriate USB driver for your device.

### **Step 32**

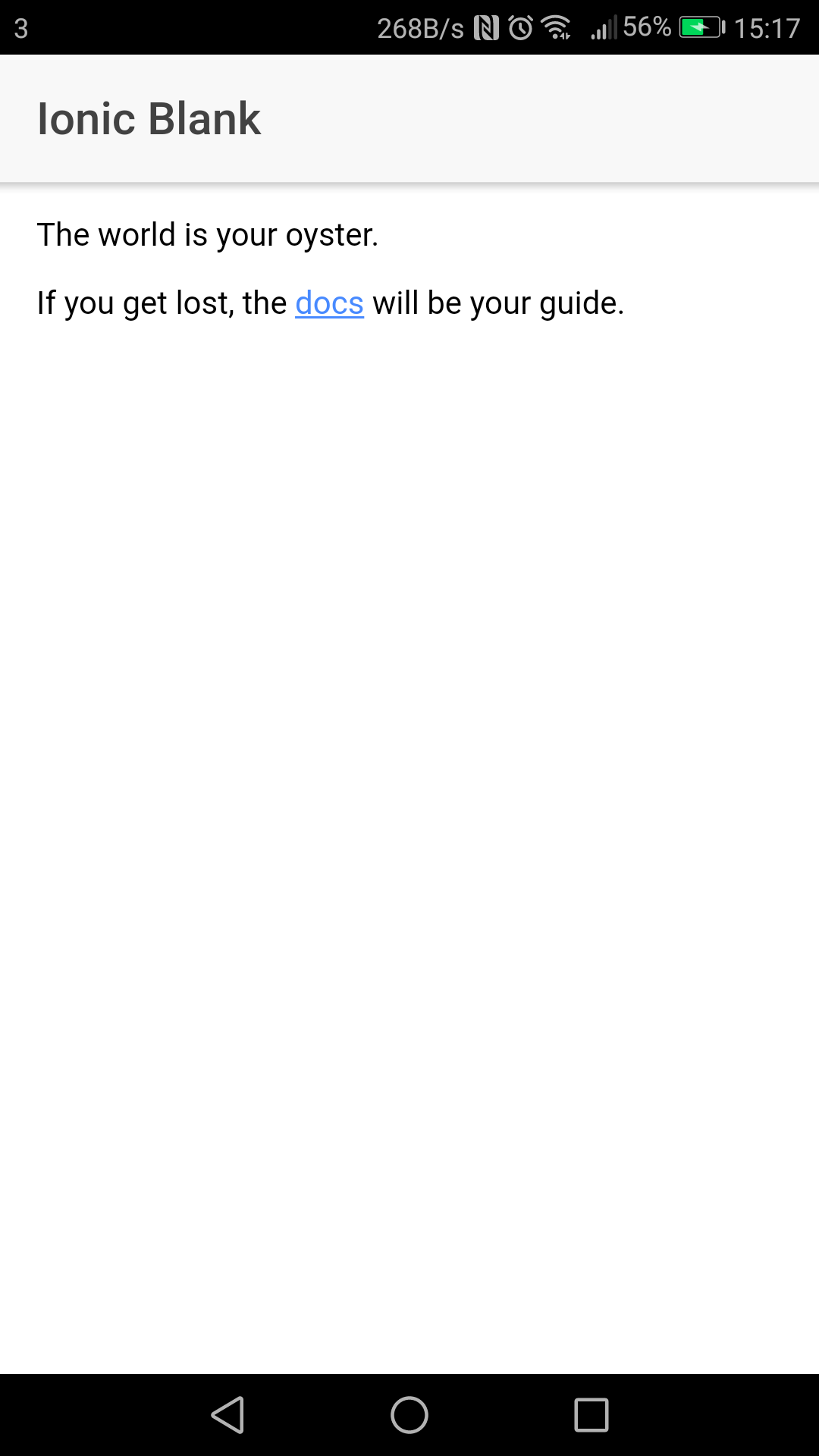
In the terminal that is opened in your ionic app folder, Run the following command

ionic cordova run android --device

You should see the following output



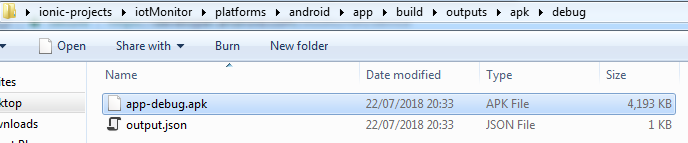
The Ionic Blank app should now be running on your phone as can be seen below.



**Step 33**

If the command did not work, open the following directory.

Desktop\ionic-projects\iotMonitor\platforms\android\app\build\outputs\apk\debug



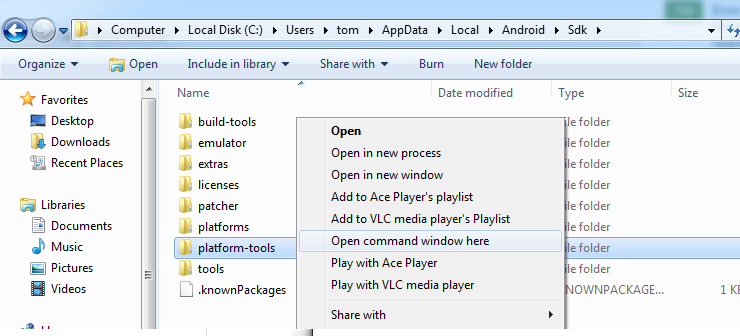
If you are unable to get USB-Debugging to work for your mobile device, you can copy the .apk file onto your phone. Then locate the .apk on your phone and open it. This will install the app on your phone.

**Debugging**

To figure out if the mobile device is being detected, open the following directory.

Make sure to hold the “shift” key when you right click on platform tools.

Click open command window here.

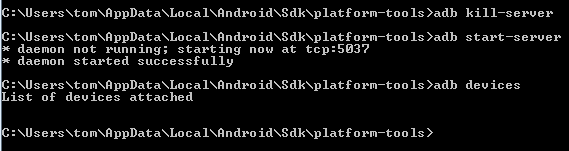


Run the following commands in the terminal.

adb kill-server

Adb start-server

Adb devices



The third command “adb devices” lists devices connected to your pc.

You should see something like this.

