**Mobile Application Testing Documentation**

# Prerequisite

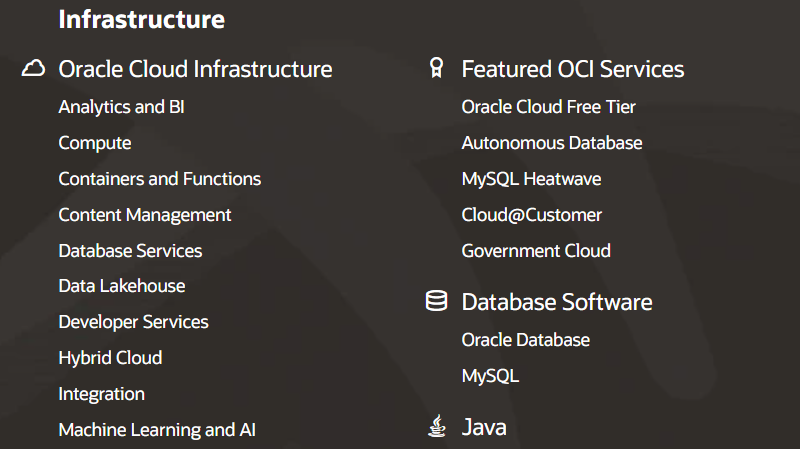
1. Install JDK (Java Development Kit) set jdk path in windows.
2. Install Android SDK (Software Development Kit) set sdk path in windows.
3. Install Node.js
4. Install Appium
5. Install Appium Inspector
6. Install IDE (IntelliJ or Eclipse)
7. Appium client Library

* **Install JDK and set JDK path in windows**

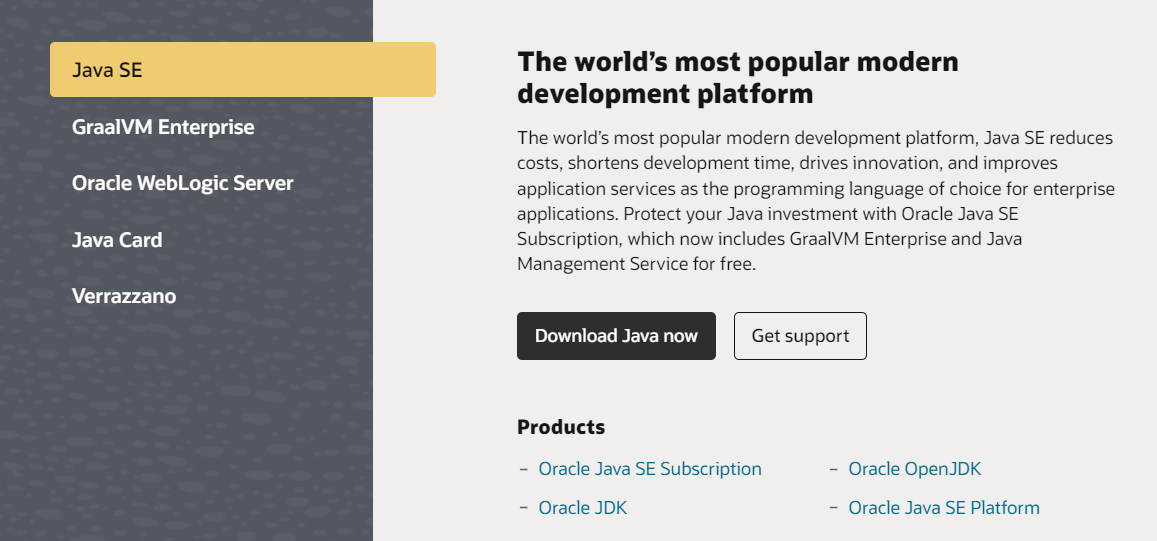
Step 1: Visit the oracle site <https://www.oracle.com/index.html> and click on Products as shown in figure below



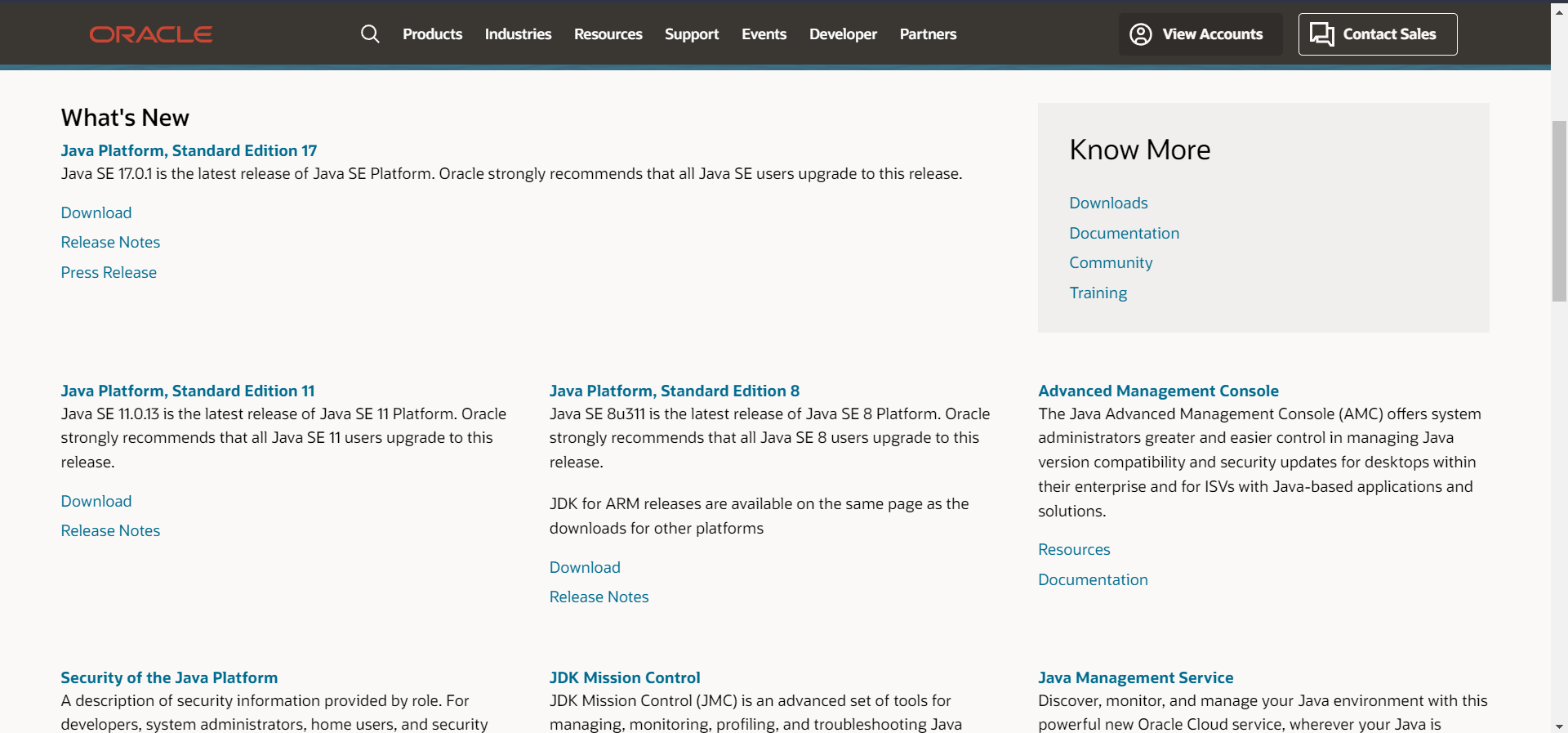
Step 2: Under product section click on Java.



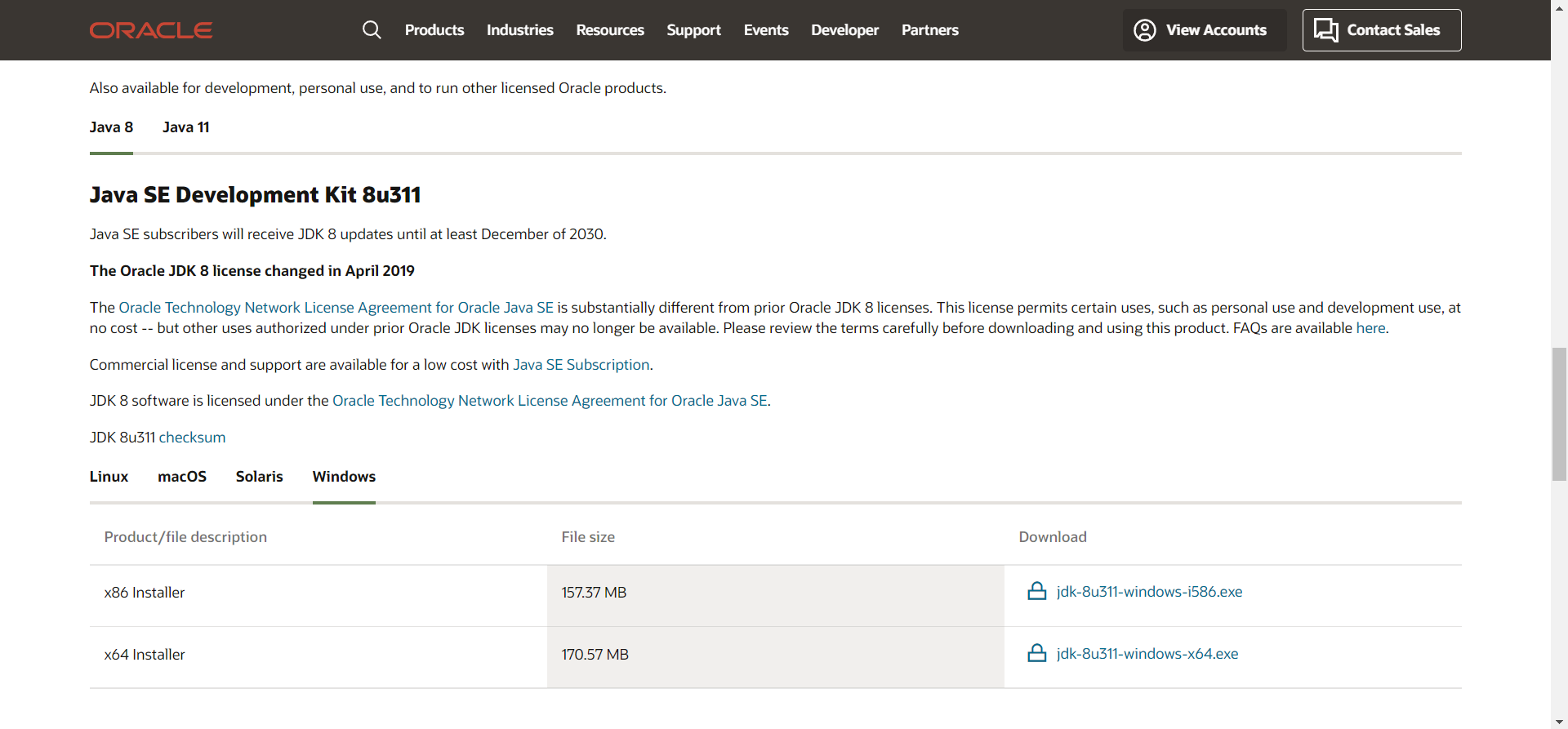
Step 3: Scroll down and click on Oracle Java SE Platform



Step 4: Scroll down and click on Java Platform, Standard Edition 8

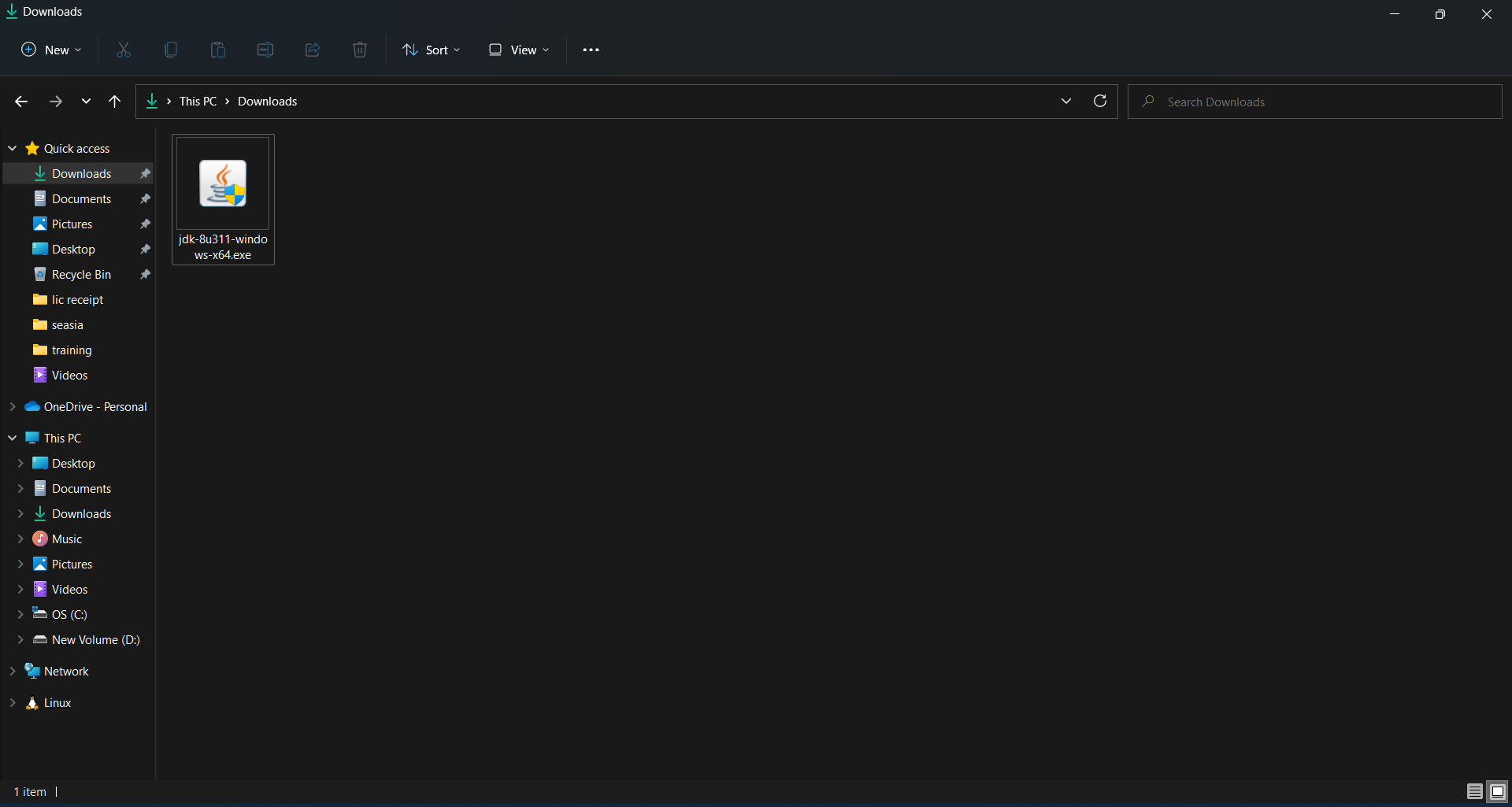


Step 5: Download the Jdk executable files.

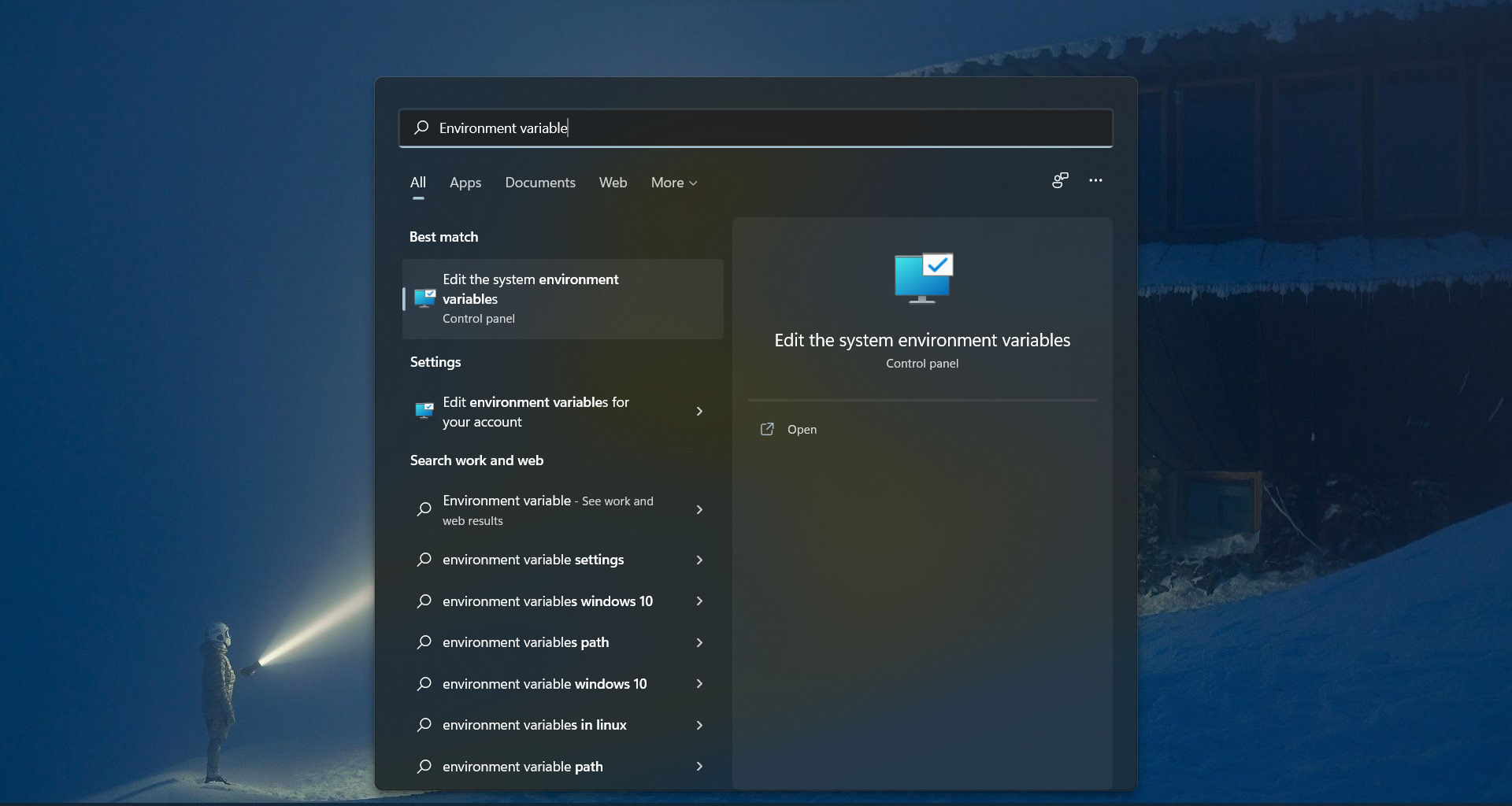


* **Installation:**

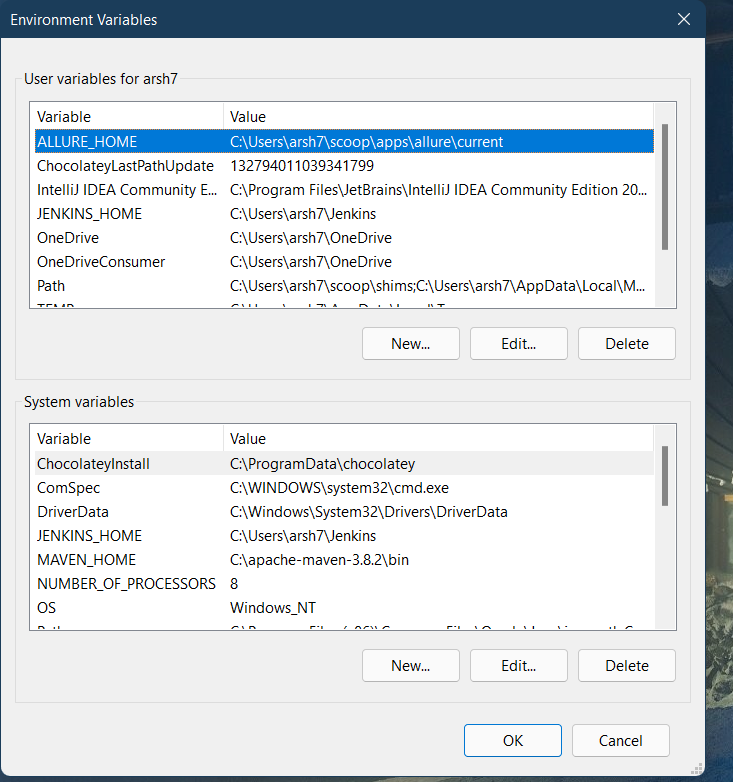
Step 1: Install the JDK in to your system by double click on the executable file.



Step 2: After successful installation search Environment variable and open it.



Step 3: Click on the new button to create a new variable for Java.

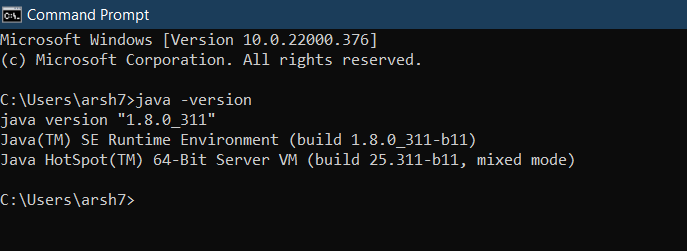


Step 4: Enter Variable name JAVA\_HOME and value where the JDK is installed that is **C:\Program Files\Java\jdk1.8.0\_311** for my system



Step 5: Done now you can check that it is successful installed and configured in your system by running this command in windows terminal.

**java -version**

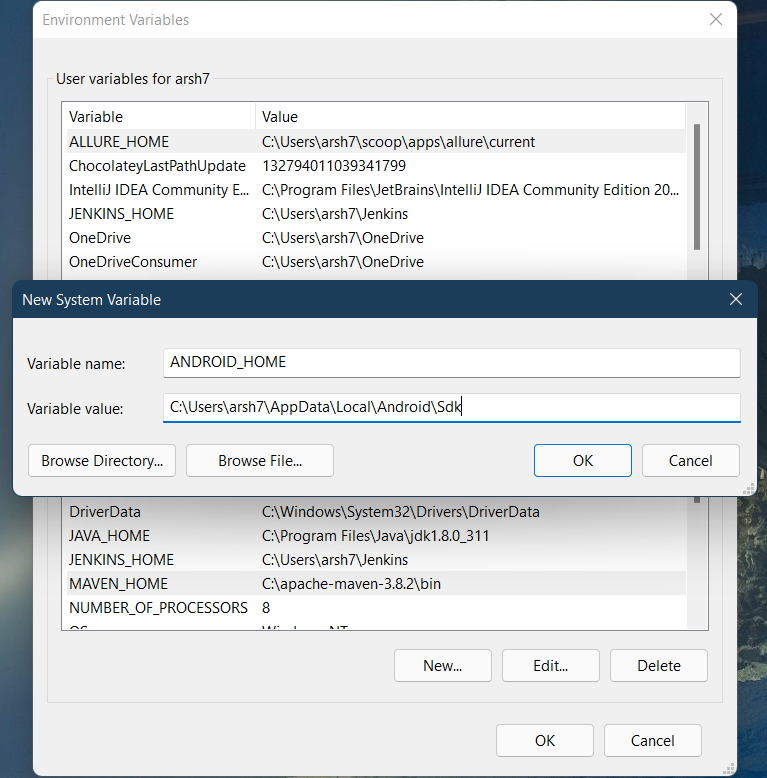
****

* **Install Android SDK and set sdk path in windows**

Step 1: Visit the URL <https://developer.android.com/studio> and download the android studio.

Step 2: Double click on Android studio executable file and install the Android studio.

Step 3: After installation again search environment variable and open and click on new and enter variable name ANDROID\_HOME and value **C:\Users\{your-username}\AppData\Local\Android\Sdk** it is same for every system.



* **Install Node.js**

Step 1: Visit the URL <https://nodejs.org/en/download/> and download the Nodejs executable.

Step 2: Double click on the file and install it.

Step 3: By default, node.js set the environment variable by itself.

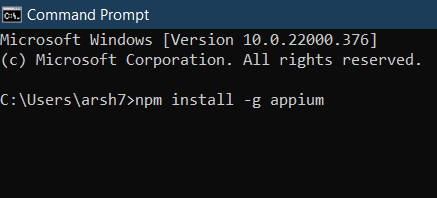
* **Install Appium**

Appium can be installed in one of two ways: via NPM or by downloading Appium Desktop, which is a graphical, desktop-based way to launch the Appium server.

From NPM

Step 1: Open Command Prompt and write this command it will install the appium in your system.

**npm install -g appium**

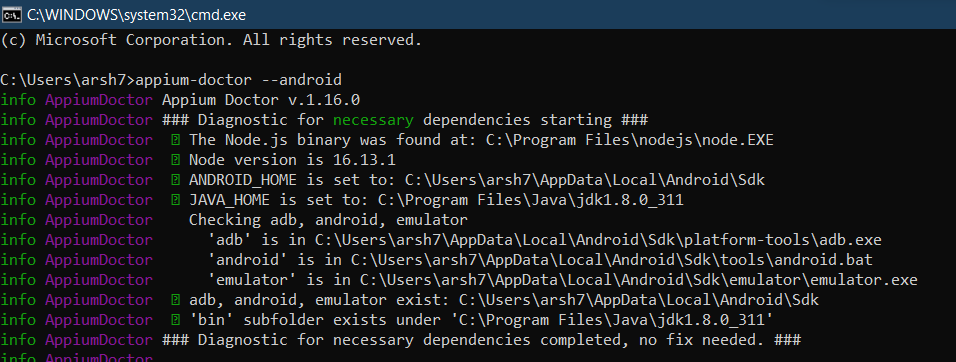
****

Step 2: You can check that appium is install successfully with required dependencies by first installing appium doctor use this command to install it.

**npm install appium-doctor**

Step 3: Run this command and it will show result.

**appium-doctor --android**

****

* **Install Appium Inspector**

Appium Inspector is a tool that help to locate the elements for both Android and IOS. To use Appium Inspector, we have to download it into our system and install it by just double clicking on the file.

Latest version of Appium inspector can be downloaded from the below mention URL.

<https://github.com/appium/appium-inspector/releases>

* **Appium client library**

Appium library are essential to write test with java and it can be imported to project by various methods like Maven, Gradle or we can manually download it and add to the projects build path.

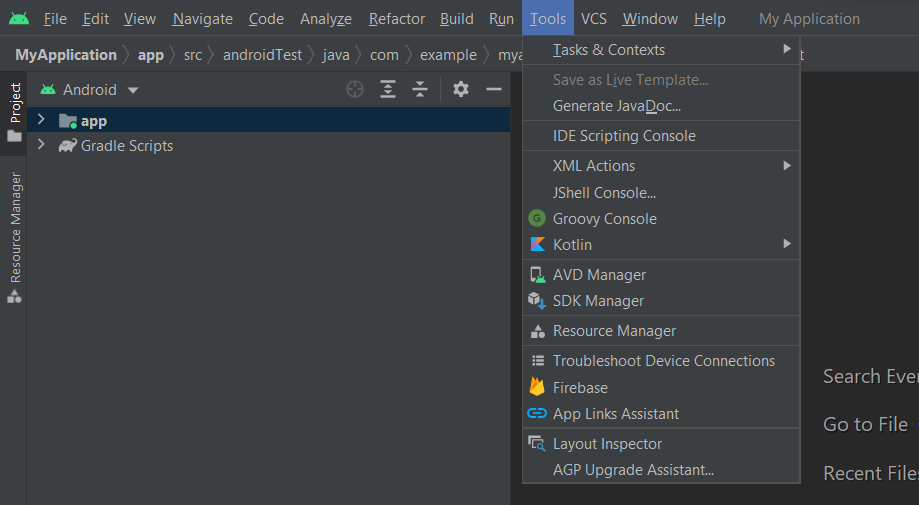
For maven this is the URL and jars can also be found in this web page <https://mvnrepository.com/artifact/io.appium/java-client>

# Writing the first Test with Appium

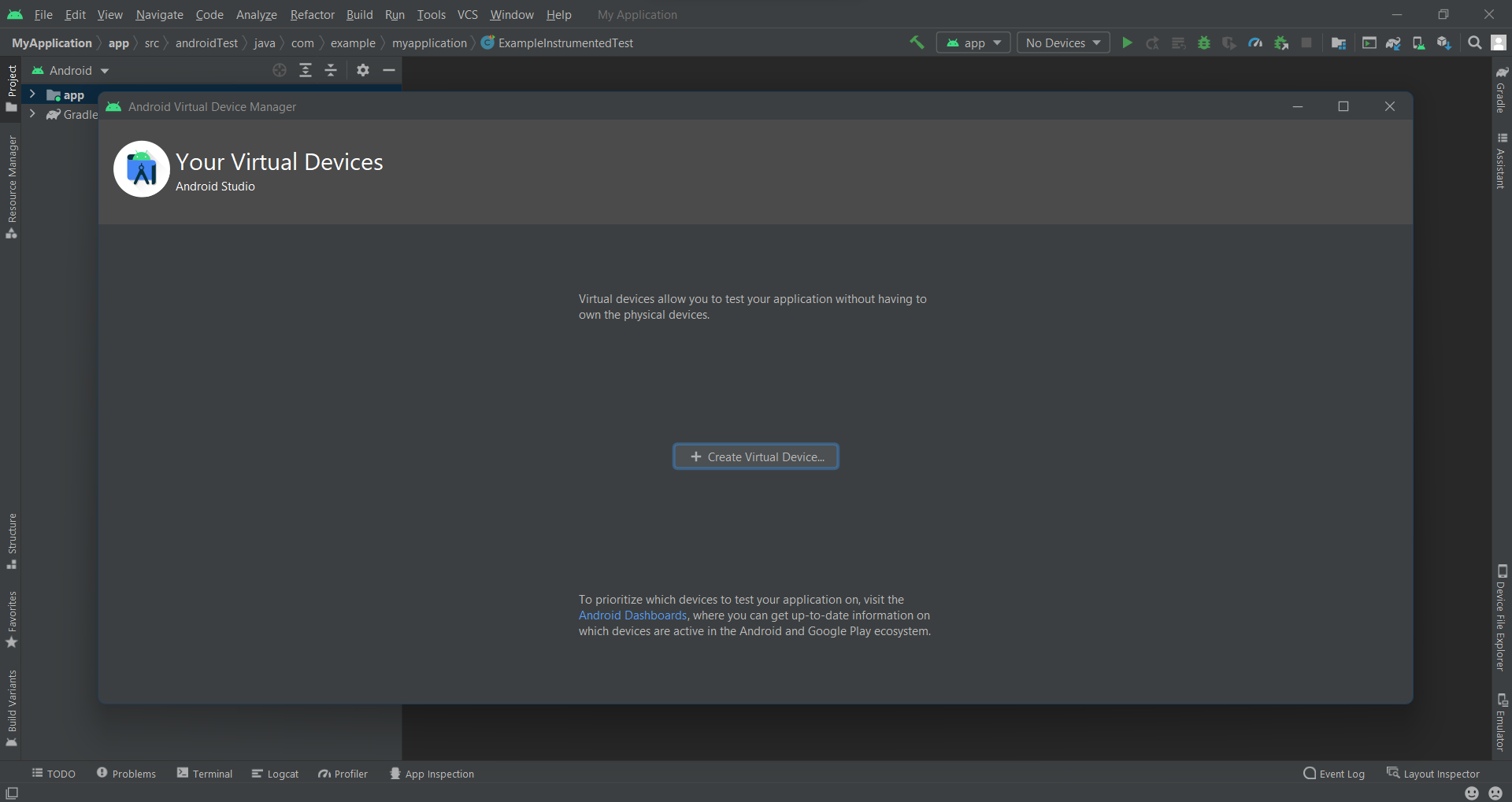
## Create Virtual Device

Step 1: Open Android Studio and create an emulator.

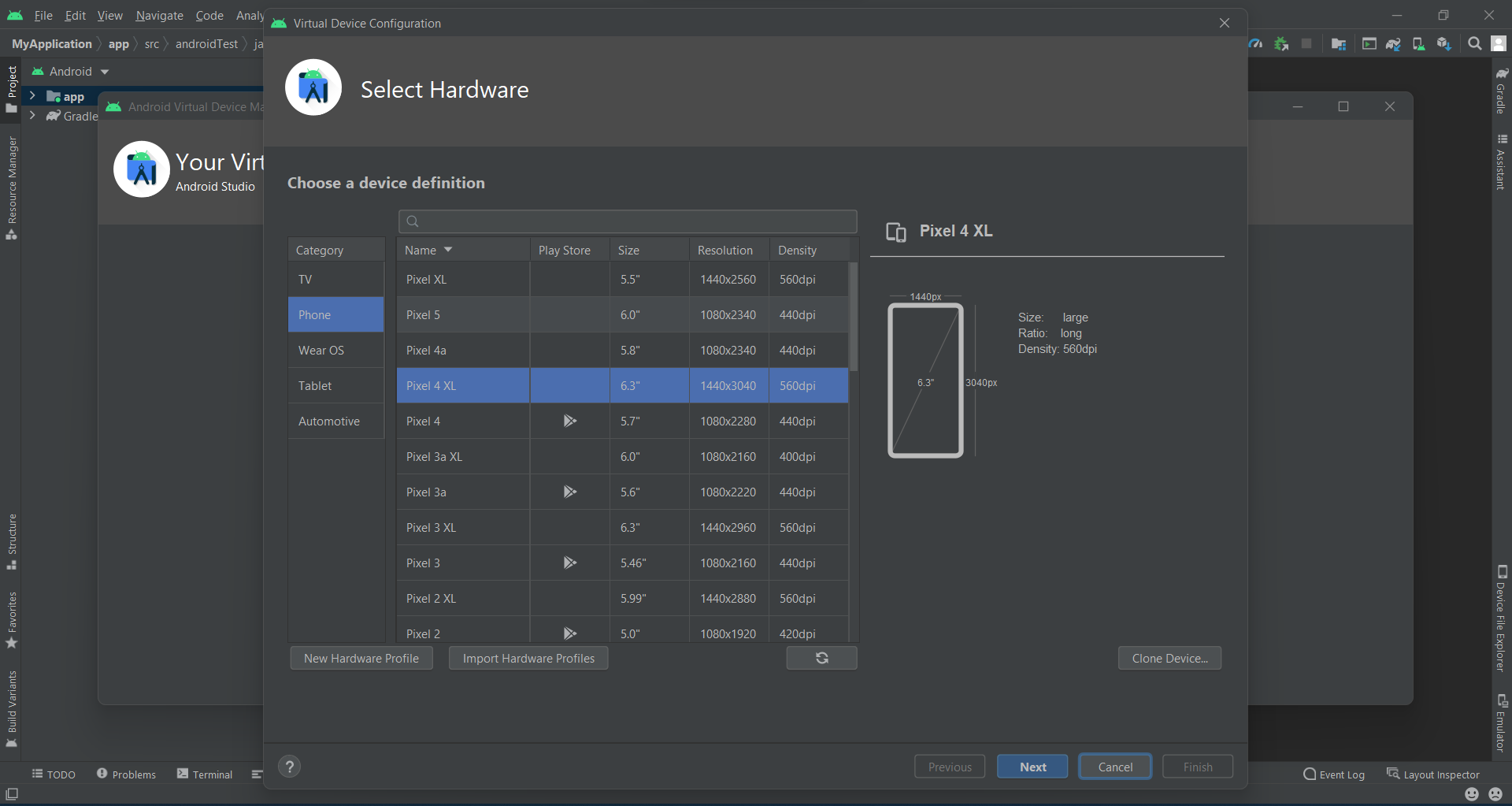
Step 2: Click on tools and open AVD manager.

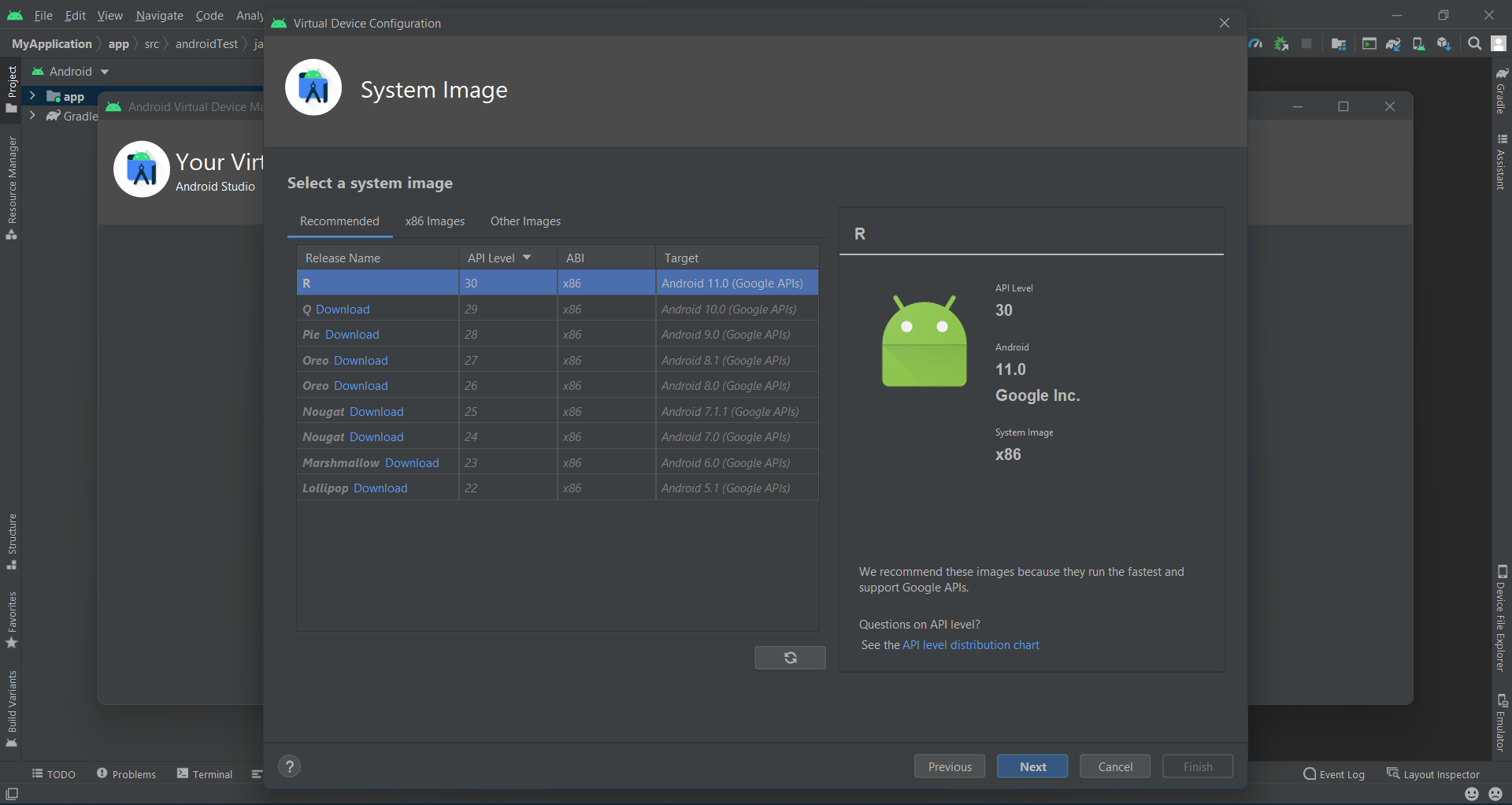


Step 3: A dialog box will open and click on Create virtual device.



Step 4: Select the hardware and os version for the virtual device and give a name to your virtual device and click on finish.



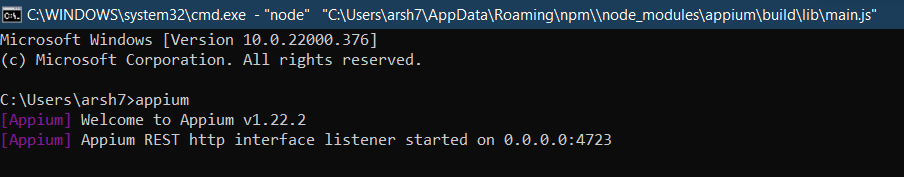


Step 5: Click on play icon to run the emulator.



## Start Appium Server

* + Appium Server GUI
    - Open the Appium Server and click on start server.
  + Node.js
    - Open the Command prompt.
    - Type the command **appium** it will start the server in default port which is 4723. It can be run on other port by using -p with the command like (appium -p 8000)

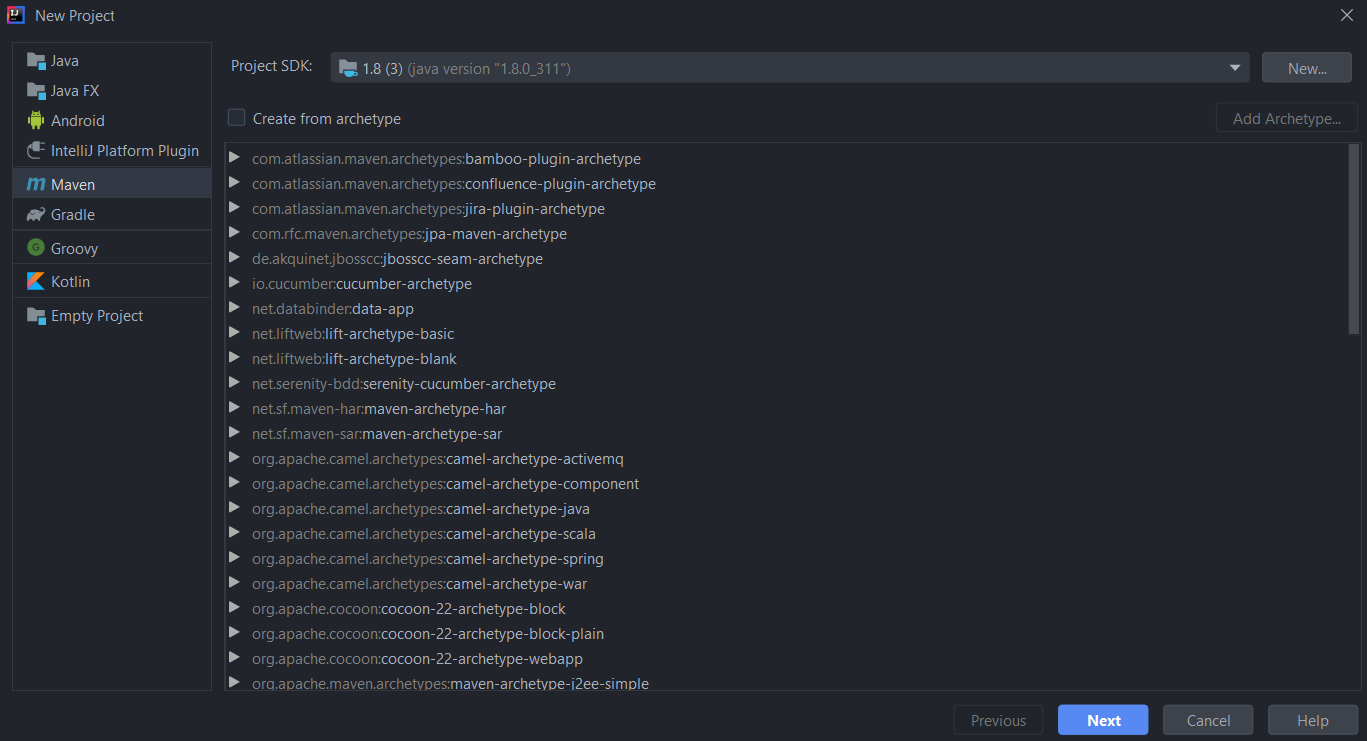


## Create Project with maven

Step 1: Create a new project.

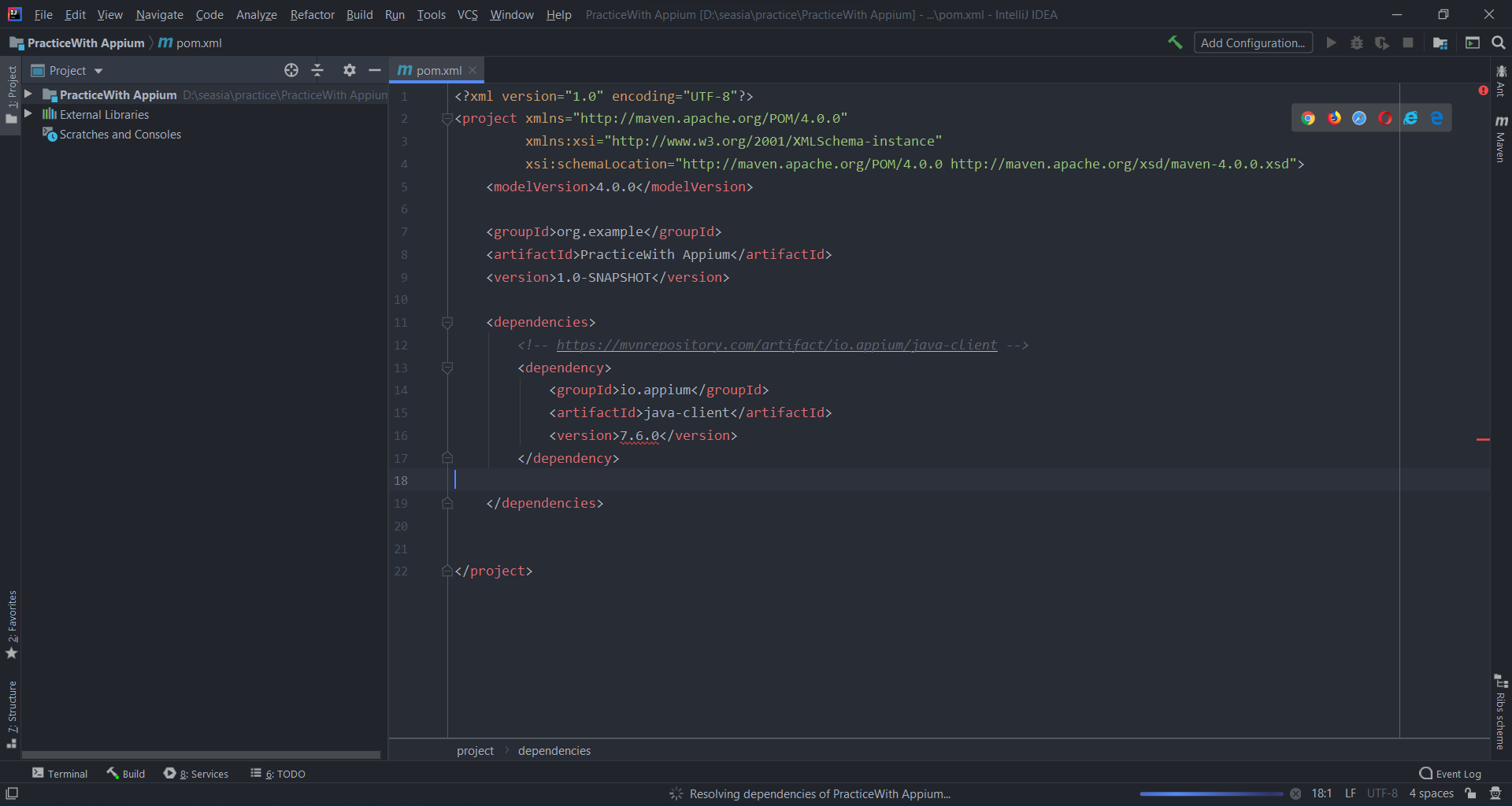


Step 2: Click on Maven and next.

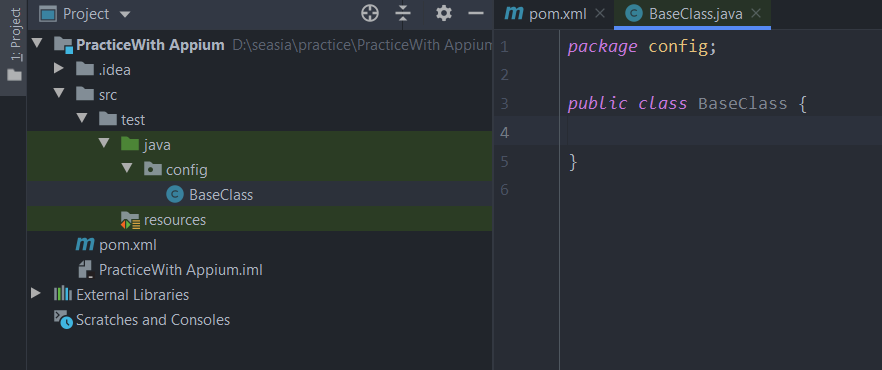


Step 3: Give your project a name and click on finish button.

Step 4: Add the dependency for appium java client in your pom file.

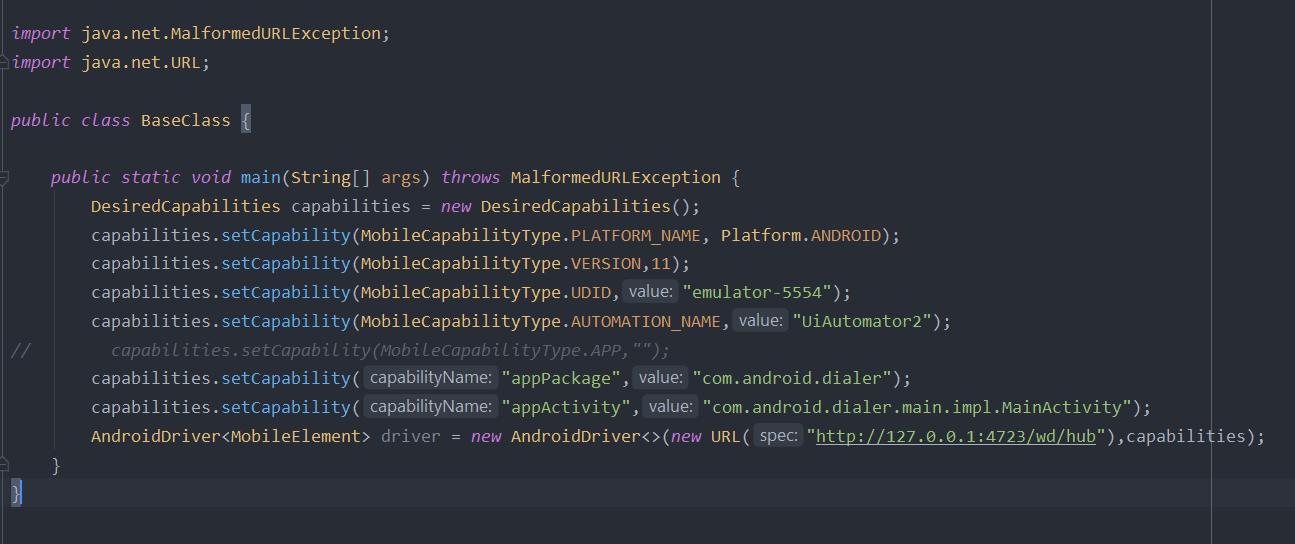


Step 5: Create a package inside the java folder and create a class in it. As shown in the figure.



Step 6: Start Appium Server please refer to the steps mention above for starting a Appium server.

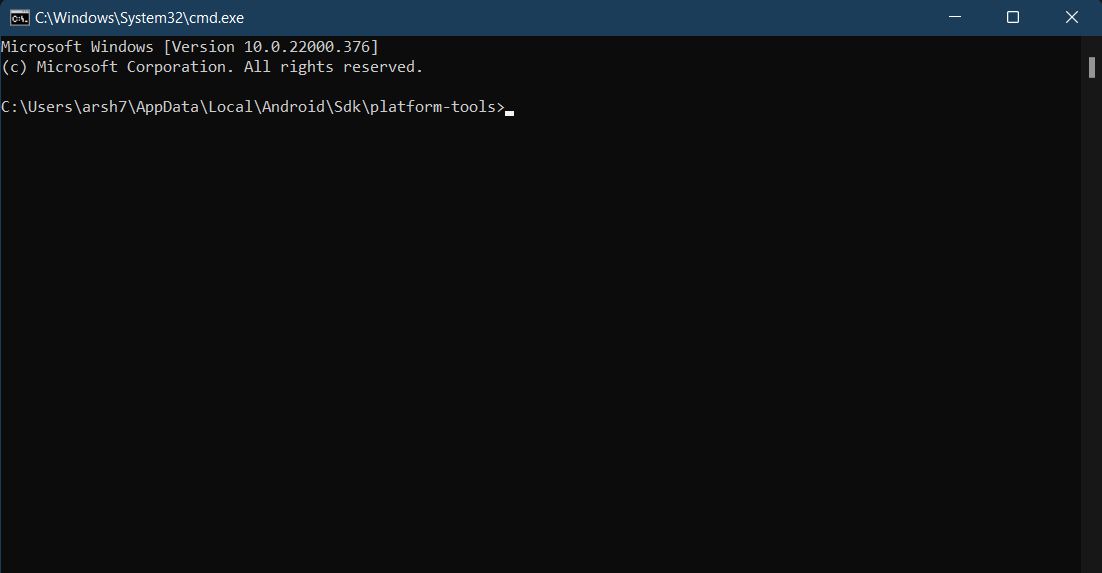
Step 7: Create a main method inside the BaseClass in which we write code to create session with appium server and these are the mandatory capabilities in order to create a session with appium server.



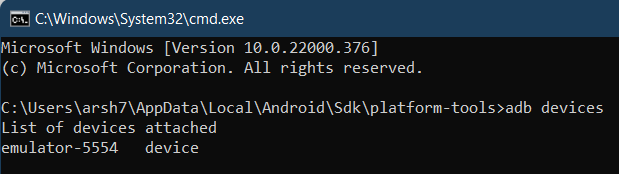


* + - Find UDID

To find UDID of device we have to first go to the Android sdk location and open the platform-tools folder and open command prompt in that particular folder like shown in the figure.



And type **adb devices** there you will see UDID

****

* + - Find appPackage and appActivity

To find appPackage and appActivity of application first you have to open the application in device and then open command prompt on platform-tools folder and type below command. It will provide you the appPackage and appActivity of current opened application.

**adb shell dumpsys window | find "mCurrentFocus"**

****

**appActivity**

**appPackage**

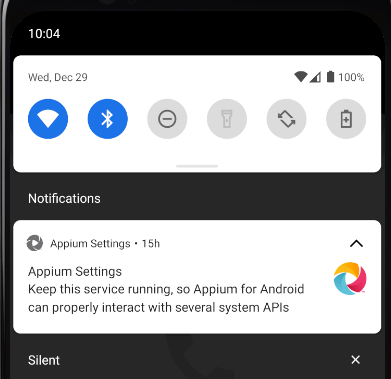
or you can download an application called **apk.info** from play store this application provides the application details like activities, version, directories etc.

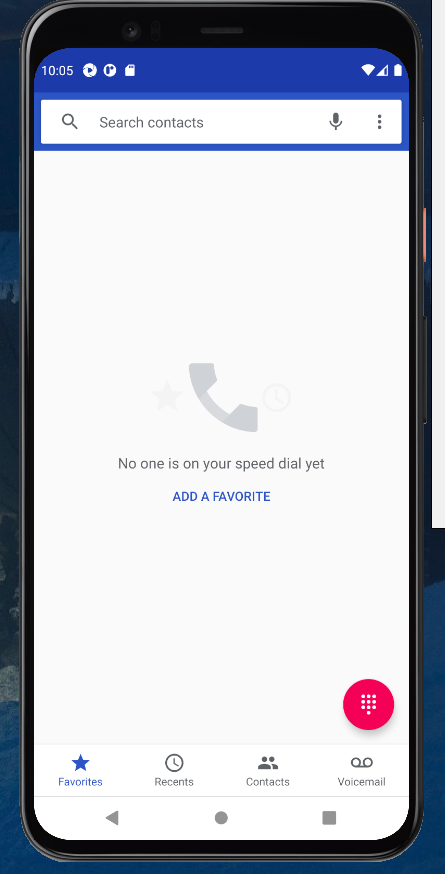
* + - Automation Name

This capability tells the Appium server to which automation to be used for android if you are working on device which has version lower that 5 or 5 than you can use UiAutomator1 or Espresso and if you are planning to test in higher versions than provide UiAutomator2.

The list of desired capabilities that can be used with Appium for android can be found in this URL <https://appium.io/docs/en/writing-running-appium/caps/> with their descriptions.

Step 8: Now run the main method make sure your Appium server is up and running and also your virtual device is up and running. Now you can open your virtual device and can see that Appium is setting things up and after that it will open the application that you have mentioned in desired capabilities.

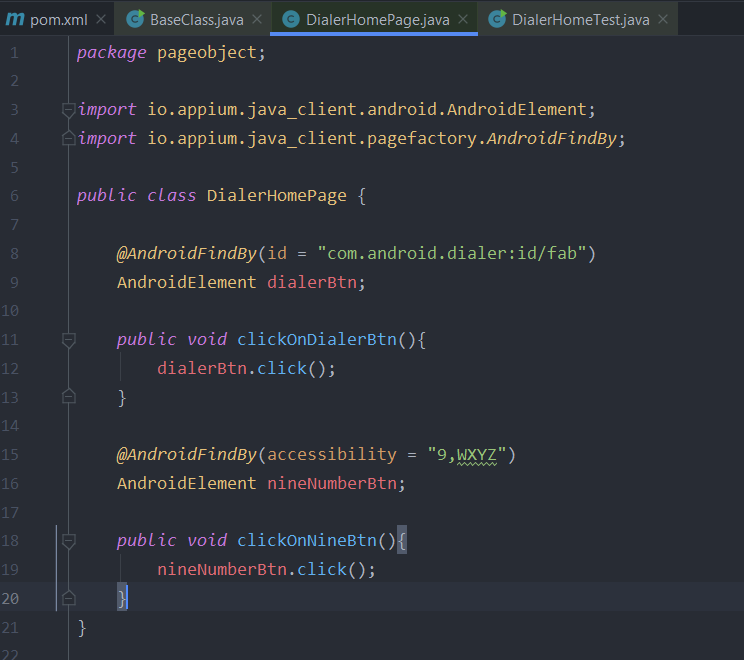




For writing the tests any tool and framework can be used like TestNG you just have to add the testng dependency to the pom file and rest of the things are same.

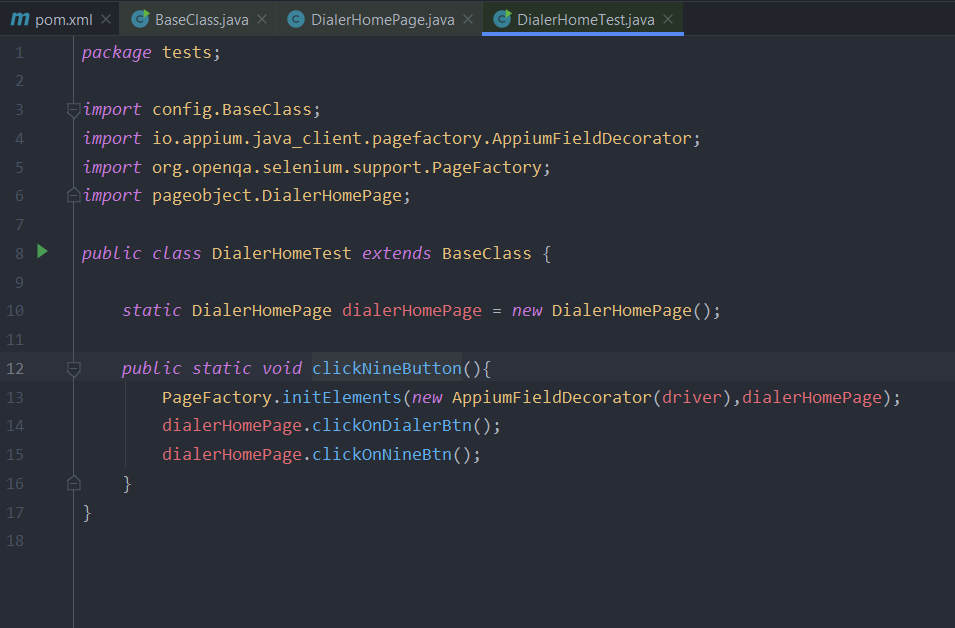
### Page Object Model with Appium

In Appium instead of using the Selenium WebElement we use AndroidElement for native applications because it provides us the methods that can be perform on android elements. We can use WebElement if we are automating the web application in mobile. Instead of using the Selenium FindBy Annotation we use AndroidFindBy Annotation of Appium.



**Page Factory**

1. First create an object of page class.
2. Invoke the static method of PageFactory class.
3. Pass two parameters as shown in the figure below.



You can keep this method under test annotation of TestNG, I have invoked this method in my main method in BaseClass that is why it is running for me.

## Switching Between Views of Application from Native to Web and vice versa

As you know Appium is used to Automate the Native, Hybrid and Web Applications. So, if the Application is Hybrid than we have to check that when it is switching from Native view to Web and vice versa. For that Appium provides us two ways:

* **Automatically entering the web view context on session start** If your application begins in a web view, and you do not want to automate the native application before entering it, you can have Appium automatically enter the web view context on session initialization by setting the **autoWebview** desired capability to **true**.



* **By invoking the getContextHandles()** This method return the Set<String> of available views of application like (Native\_App, WebView) and by using context(parameter) method we can switch between views by passing the id of view as parameter in context method. Below is the sample code.

Sample Code



## Start Appium Server Through Program

**Steps needs to follow to run Appium server through code are as following:**

1. Install node into the system and with node install Appium.
2. Create an object of AppiumServiceBuilder class



1. With the help of appiumServiceBuilder object call method withAppiumJS(location) and give the location of main.js file located at

C:\Users\{yourusername}\AppData\Roaming\npm\node\_modules\appium\lib\main.js

**Note:**

Give the path to executable appium.js (for version of Appium 1.4.x and lower) or main.js (version of Appium 1.5.x and higher). Appium.js file can be found in the same location as main.js.

1. With the help of withIPAddress(IP address) method you can specific the like in which IP address server will run if not called default value will be used.
2. With the help of usingPort(port number) method you can tell Appium in which port your Appium server will run if not called than default port will be used or use can use method usingAnyFreePort() it will use any free port available.
3. After all these 6 steps than you have to create variable of class AppiumDriverLocalService. AppiumDriverLocalService objectName; after that we will call the static method buildService(appiumServiceBuilder) of AppiumDriverLocalService class and pass the object of appiumServiceBuilder



1. With the help server object we will call methods of AppiumDriverLocalService class like to start the server we will call server.start(); after that if you want stop the server than call server.stop();

For reference you can refer to this website

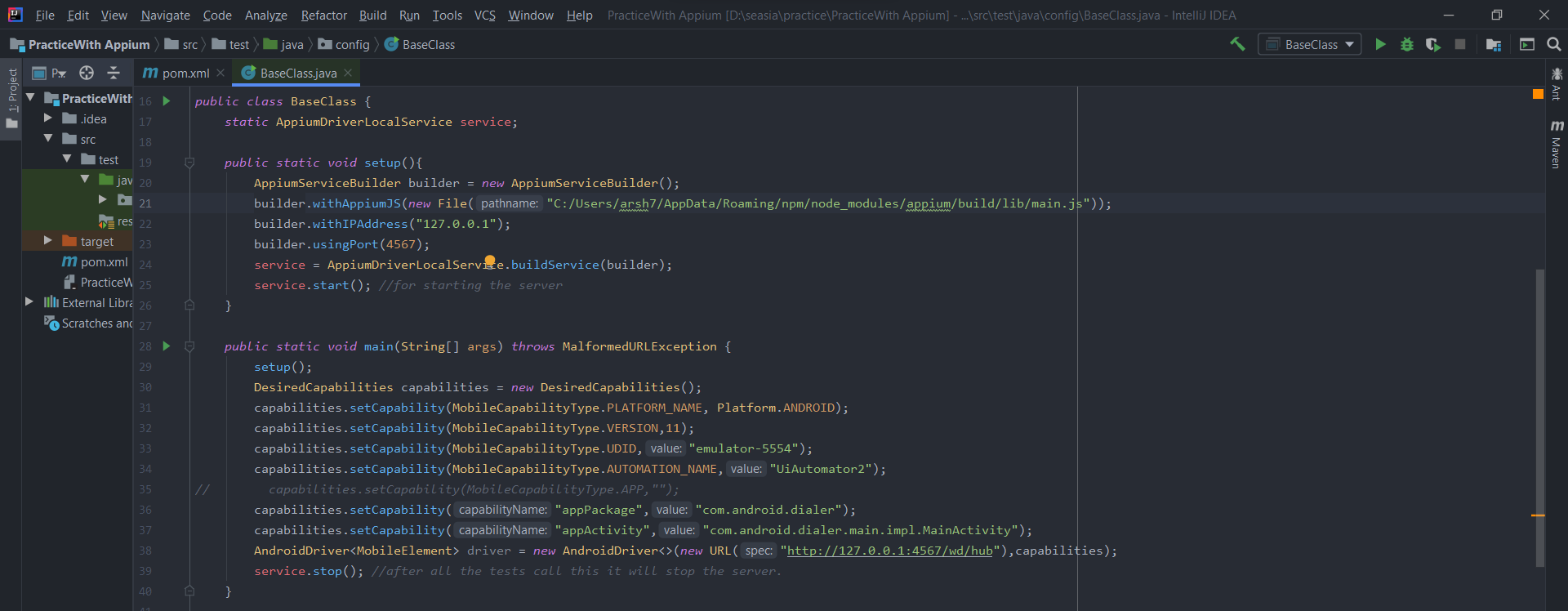
<https://www.seleniumeasy.com/appium-tutorials/how-to-start-appium-server-programmatically>

**Sample code:**

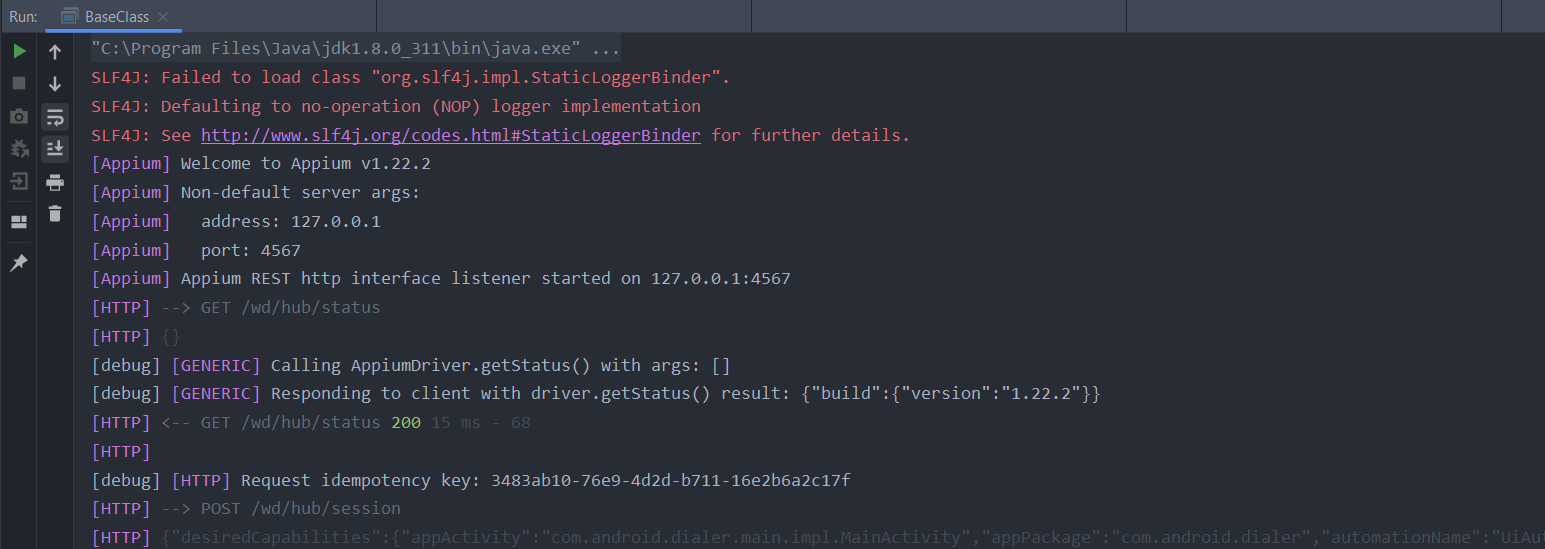


**Note:**

While running the above code if any issue occurs in the main.js file than you can use the other main file located at this location C:\Users\arsh7\AppData\Roaming\npm\node\_modules\appium\build\lib\main.js



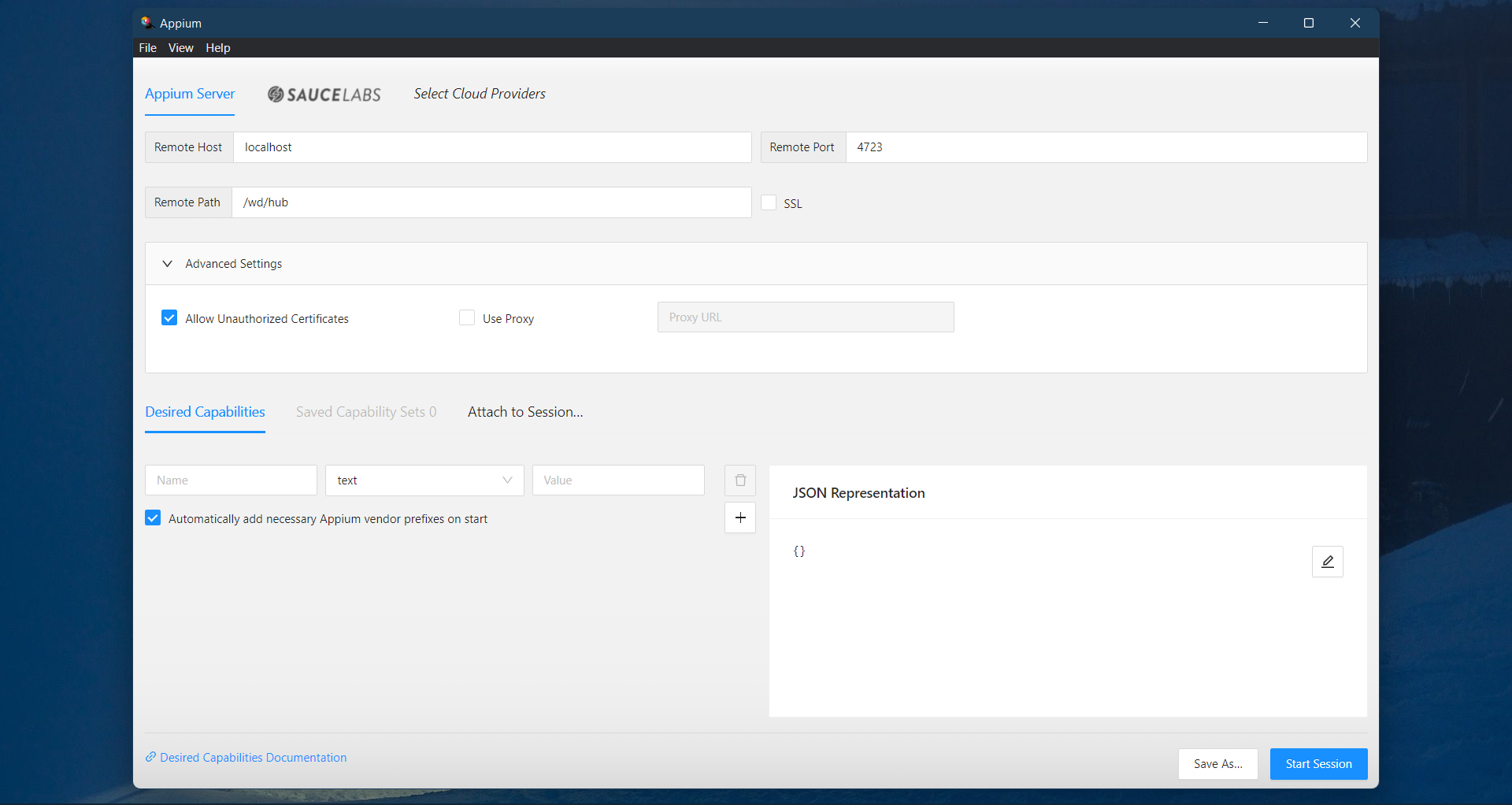
Now run this code and as you can see in figure below the Appium server log will start to print in the console.



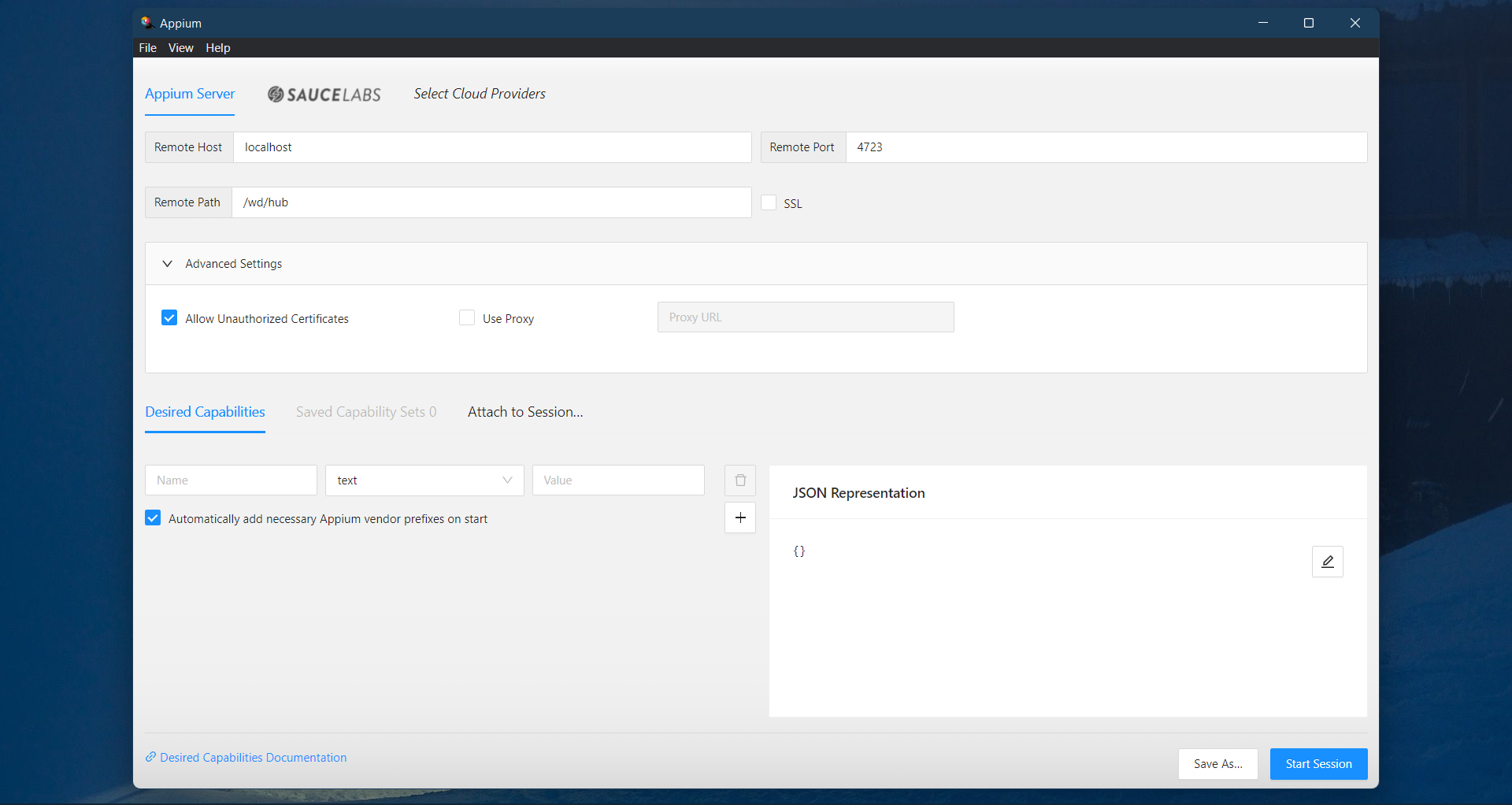
## Locating Elements with Appium Inspector

Step 1: Start the Appium server.

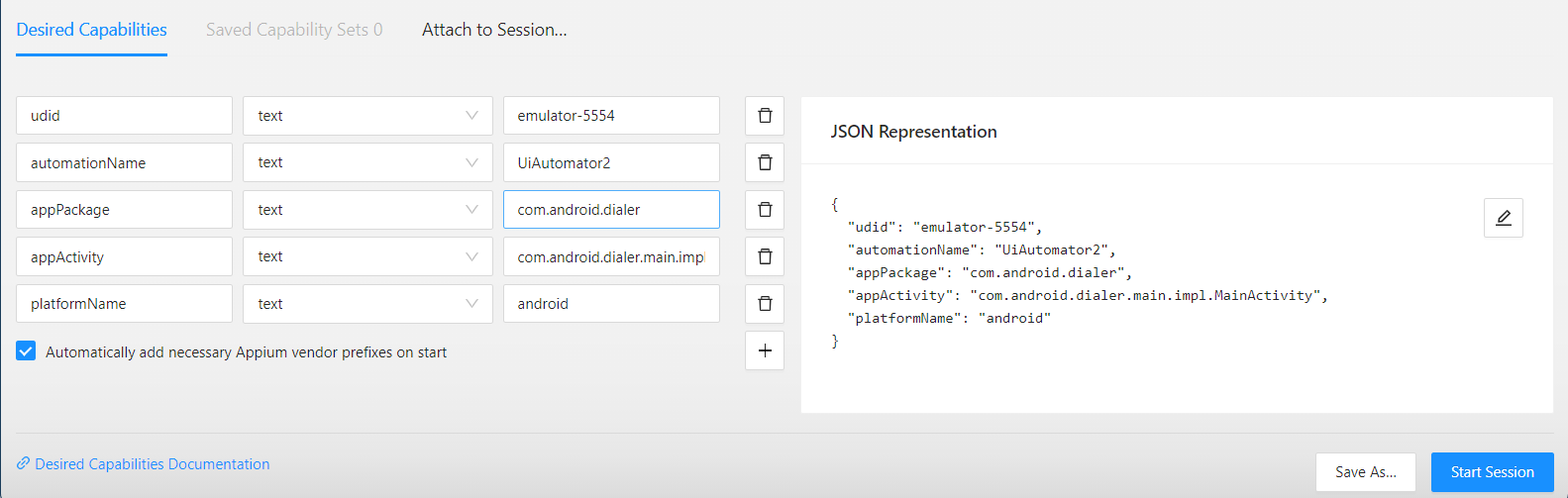
Step 2: Open Appium Inspector by click on its desktop icon. A window will open which will look as shown in the figure below. If you have started the server in default settings than no need changes anything and if not than if you have to mention the remote host, port on which server is up and running.



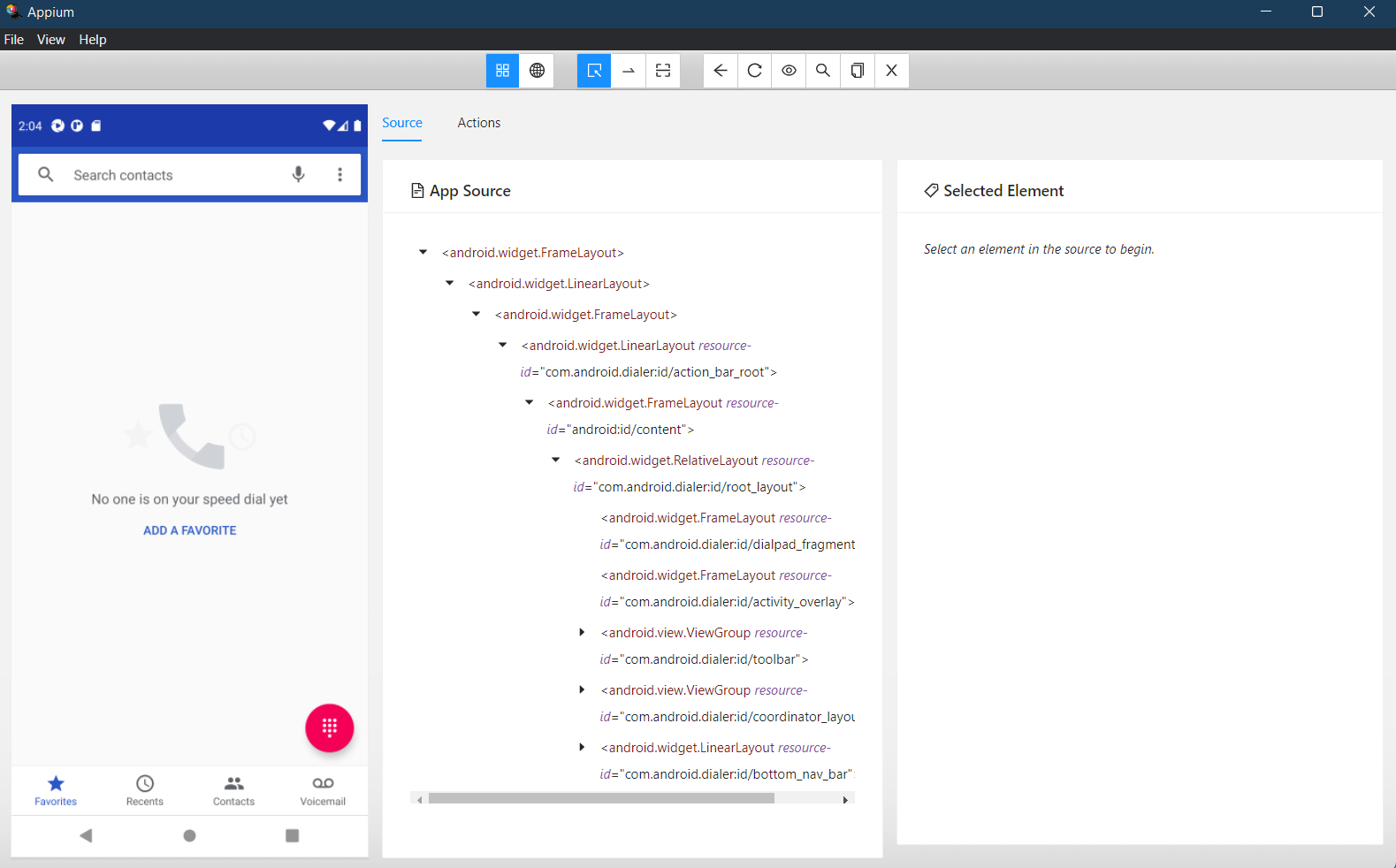
Step 3: Add the mandatory Desired Capabilities you manually type the capabilities in the box or you can directly copy paste it under the JSON Representation box by clicking on pencil button and click on Start Session.



After doing that it will look like this as you can see in the figure below. You can also save the desired capabilities by click on Save As button.



Step 4: After session is created it will show you screenshot, hierarchy, details about Selected Element like shown in the figure below.



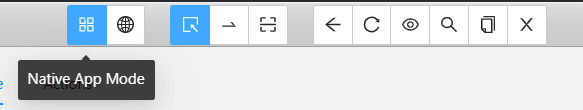
Tools

Details about Selected Element

Element Hierarchy

Screenshot

The Various tools are available to perform certain task as you can see in figure. You can hover over the buttons and it will show you the small description about the tools like shown in the figure below.



Appium Inspector provide a record built in that if enabled than it records steps and create a code for that. Just click on eye button and it will start recording. Also, you can specify the language for which code need to be generate.

