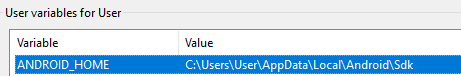
**Appium Framework Guide**

**By Henry Champion**

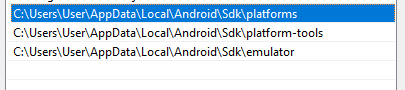
Setup steps:

1. **Install Appium Desktop Version from** <http://appium.io/>
2. **Install Java JDK and configure environment variable (JAVA\_HOME)**
3. **Install Andriod Studio (**<https://developer.android.com/studio>**)**
4. **Then set environment variables to android sdk :**

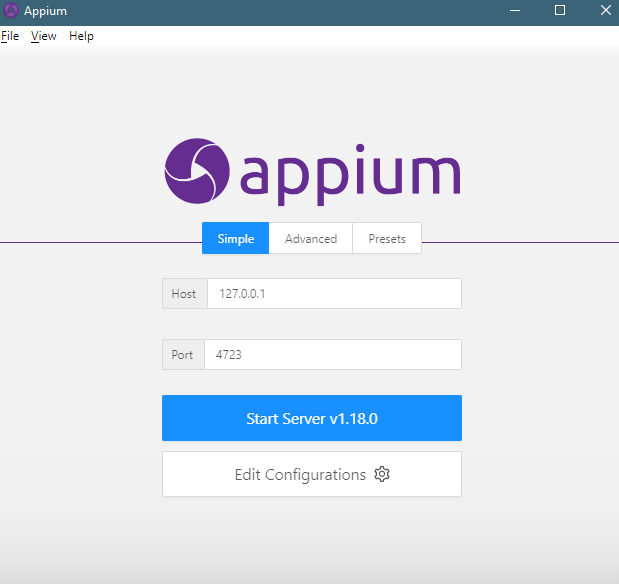


**‘Path’** system variable edited with following:





1. **Create an AVD in android studio (Pixel 2) and enable developer mode for this virtual device**
2. **Run Appium.exe (The Desktop Version you installed in step 1)**
3. **Set Host address to localhost (127.0.0.1) and start server:**



1. **Once the server is running, you can now start your maven/appiumDriver test creation**
2. **I used testNg/Junit/Appium Java-Client/extentReports:**

**File>New Project>Maven Project.**

**No archetype required.**

**Create the following directories/files once the project has been created:**

**Under ‘main/java’ create new package called ‘qa.mobile’.**

**Under qa.mobile create a new java class called ‘AppTest’.**

**Before continuing on this file (AppTest) first edit maven ‘pom.xml’ by adding the following dependencies:**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>AppiumClone</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.11</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>com.aventstack</groupId>  
 <artifactId>extentreports</artifactId>  
 <version>4.0.9 </version>  
 </dependency>  
 <dependency>  
 <groupId>io.appium</groupId>  
 <artifactId>java-client</artifactId>  
 <version>7.3.0</version>  
 </dependency>  
 <dependency>  
 <groupId>org.testng</groupId>  
 <artifactId>testng</artifactId>  
 <version>7.3.0</version>  
 <scope>compile</scope>  
 </dependency>  
 </dependencies>  
  
</project>

Ensure auto-import of dependencies is enabled. The dependencies should now be downloaded.

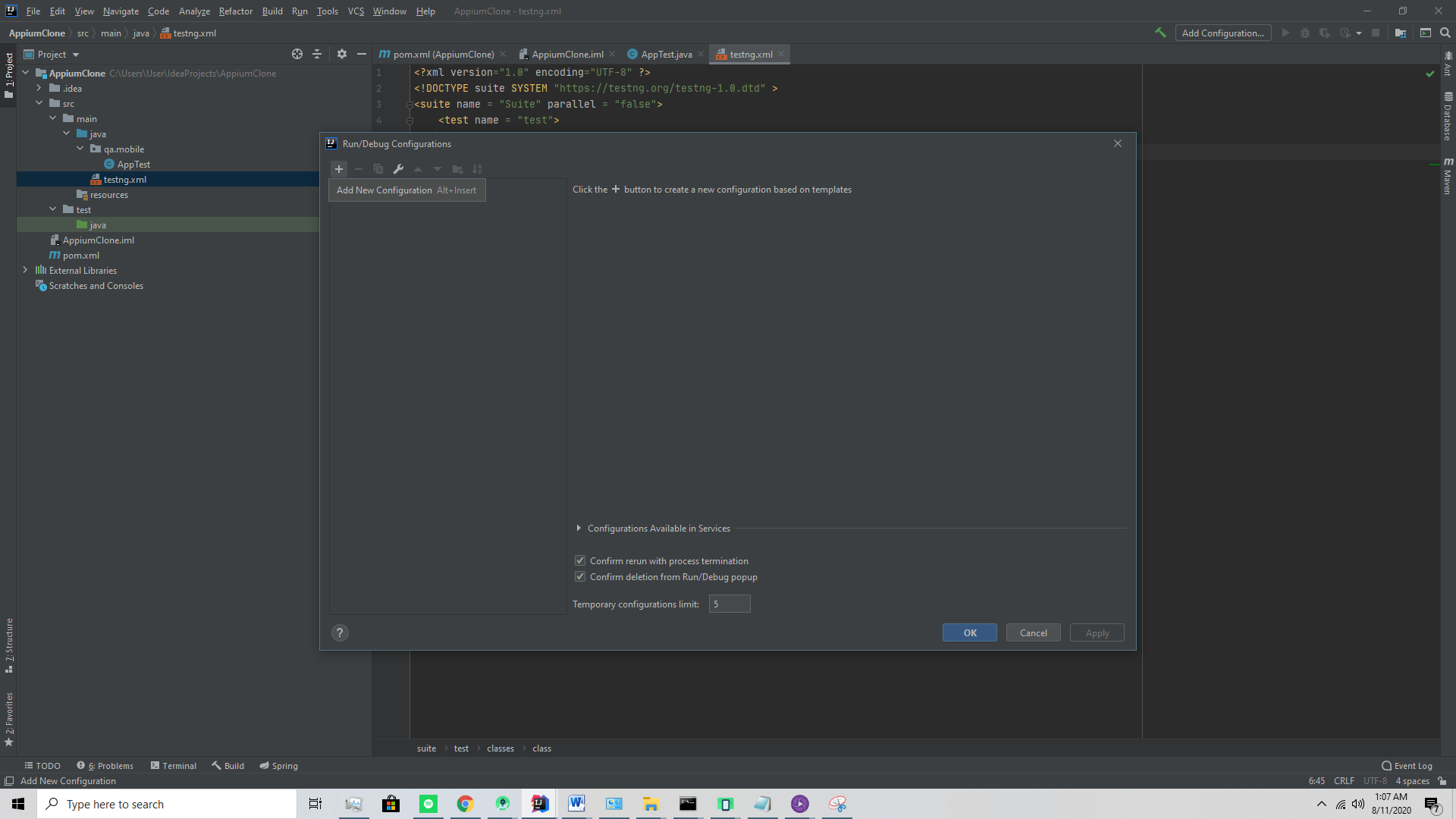
Once downloaded, head over to **src>main>java** and then under here **create a new xml file called ‘testng.xml’. Populate with the following:**

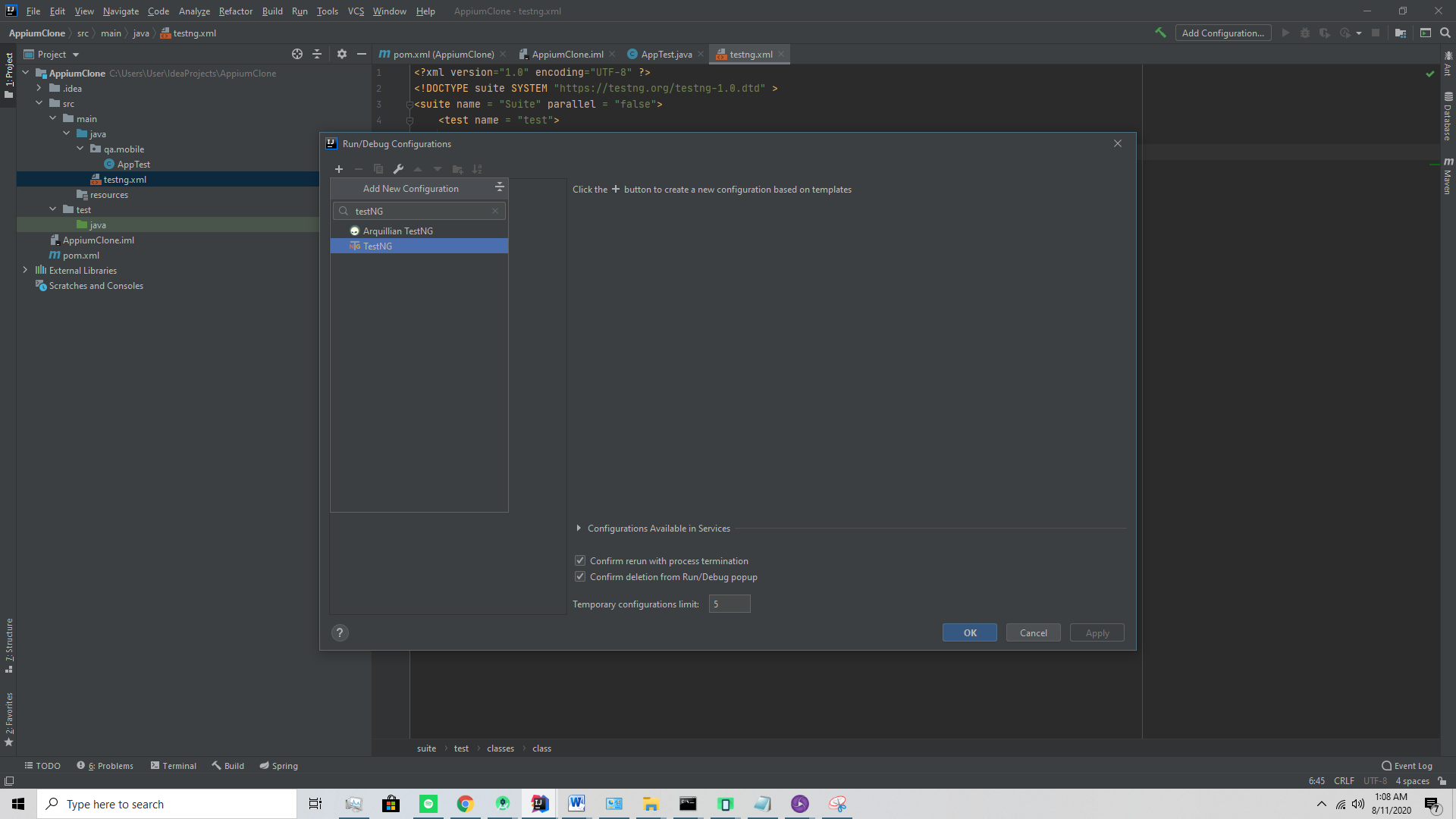
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >  
<suite name = "Suite" parallel = "false">  
 <test name = "test">  
 <classes>  
 <class name = "qa.mobile.AppTest"/>  
 </classes>  
 </test>  
</suite>

Now that your testNg file is done, you need to add this xml suite to the run configuration:

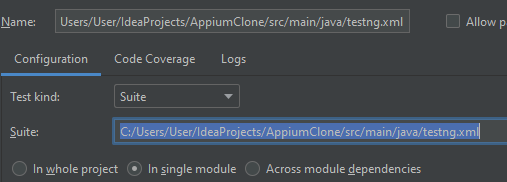


Add New > select testNG:

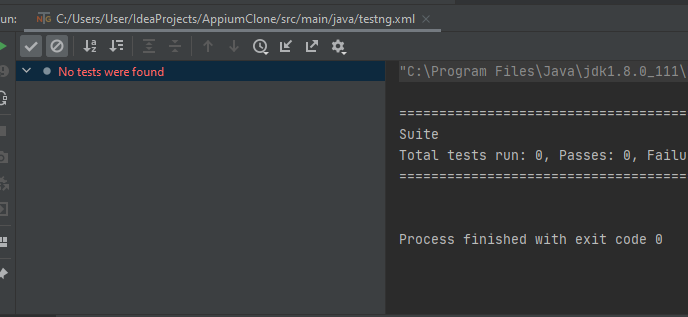




Ensure ’Suite:’ is pointing to the new testng.xml file.



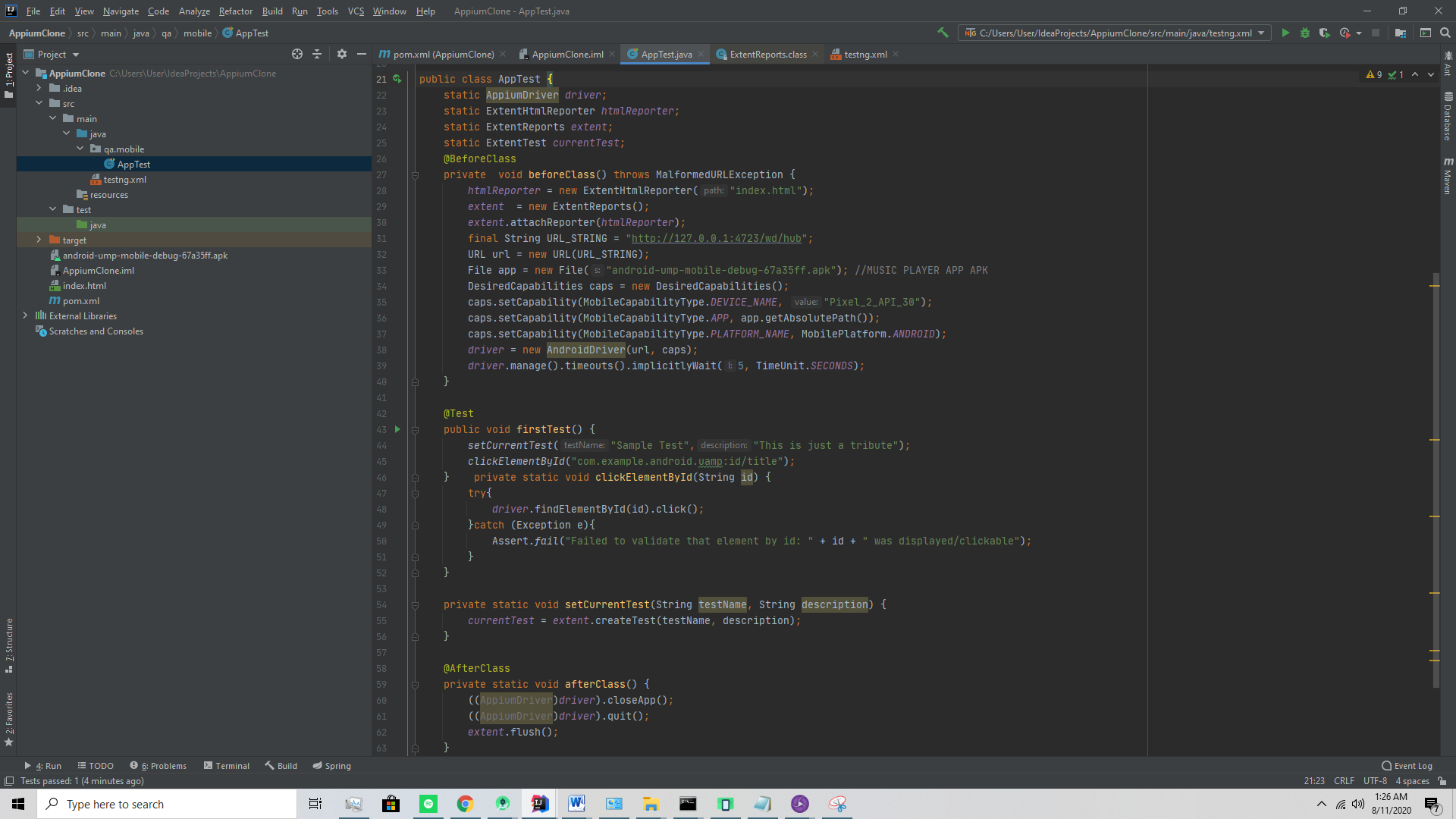
When you run now it will just do nothing and display message: “no tests were found”



The next step is to add the tests to your AppTest.java class.

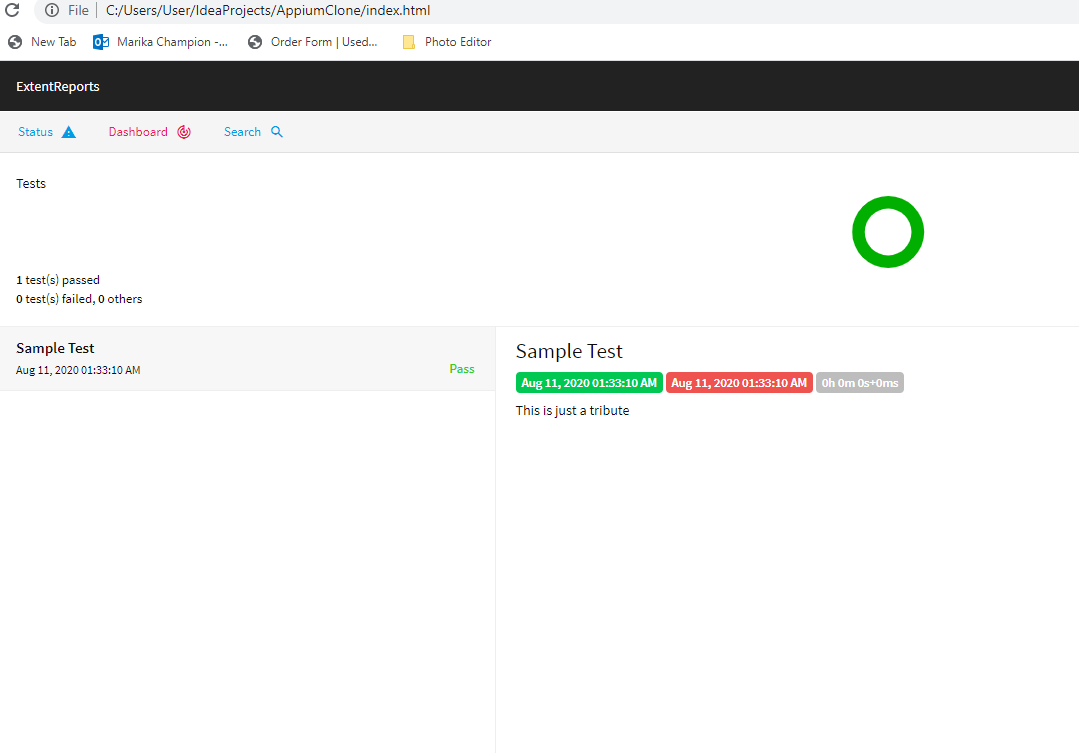
Here you will now be able to create a new AppiumDriver instance and start coding away at the new automation framework you just set up ☺

Below is a sample class for this tutorial with all the necessary functions:



Before you run this test, open your device in android studio’s emulator.

Then if you run the above code you will end up with an index.html file as a report when the program has finished:



**The full version I did is more elaborate and includes screenshots of all the actions.**