**Automation Mobile App Testing using Android driver in Java**

First important thing is you need an Android SDK (Software development kit) <http://developer.android.com/index.html>

**Day4**:

**Android SDK** – software development kit that will help Android developers to develop new mobile apps and be able to deploy them on the mobile. This is created for the Android OS. The applications developed are created as APK’s.

* This is primarily for embedded apps such as the apps on play store. Every time Android
* After installing SDK, go to program files and type SDK – Open SDK manager.
* Install further more – select Android 4.2 and Android 4.1.2 (example versions) also select “Extras’ and install package.

Emulator and simulator: Emulator is realistic. It will perform the tasks. Simulator looks like its original but it does not perform the task, it just shows you that UI.

Android Virtual Device : Is a way to add a new virtual emulator.

To launch AVD, open tools on the SDK manager and click manage tools.

**Virtual Device:**

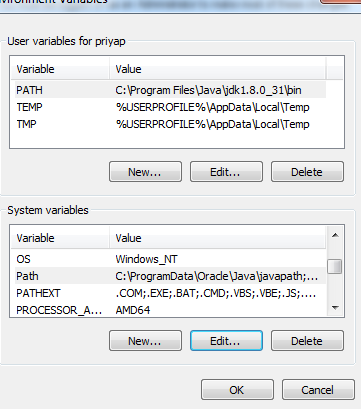
* tools – Manage AVDs – you can emulate real device. To Add, select **New** – Add a name. Select the **device type**. Choose **target** such as Android version. **CPU** – select anything such as ARM. **Keyboard**- Hardware keyboard present is important- check it. It helps you type using keyboard on the device. **Skin** can be checked. **Memory options**: RAM 512**; internal storage**: 2000(2GB). Click OK. The device appears in the Android virtual Device manager.
* Select the device and start. It will launch the device.

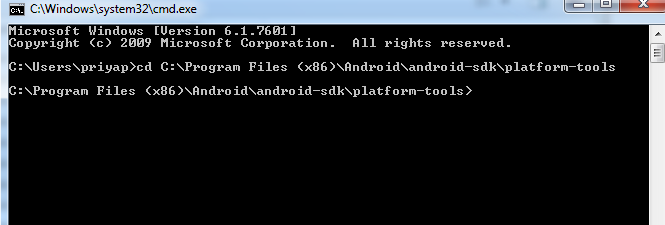
**Day 5: Testing on the real devices.**

**To Test Applications that use APK on the devices.**

* What is an Android APK file? The people who develop Android application. Once they complete developing, people can download it from market place onto a local device.
* The apps that do not use browser becomes native apps.
* Developers package all into the apk file. The entire application is zipped in that.
* Go to where the Android sdk – platform-tools is installed on your local machine, here is where you place your apk files. Those are the applications that you install on your emulators if you need to test these applications. (//programfiles//Android-sdk-Platform-tools)
* Copy this path and paste it on the path of your system environment variable.

Put a semilcolon and paste the path to the end.



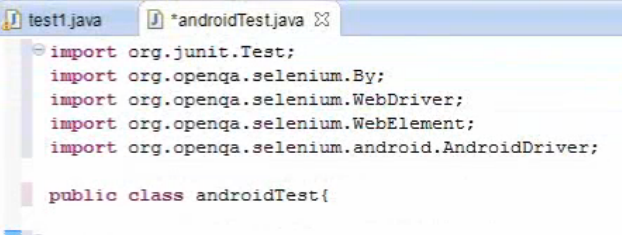
* Launch cmd prompt go to the directory where the sdk is installed and paste the same platform tools path.
* 
* Test ADB (Android developer bridge)Type adb and hit enter. This will print a lot of things such as if your system has the Android developer bridge or the Android debug bride and its version and its different features.
* Download a sample APK and paste in the path under platform-tools.
* Go to cmd – to install the apk – go to the platform-tools path – type adb install sample.apk

**Day 9: Automation of Mobile Applications.**

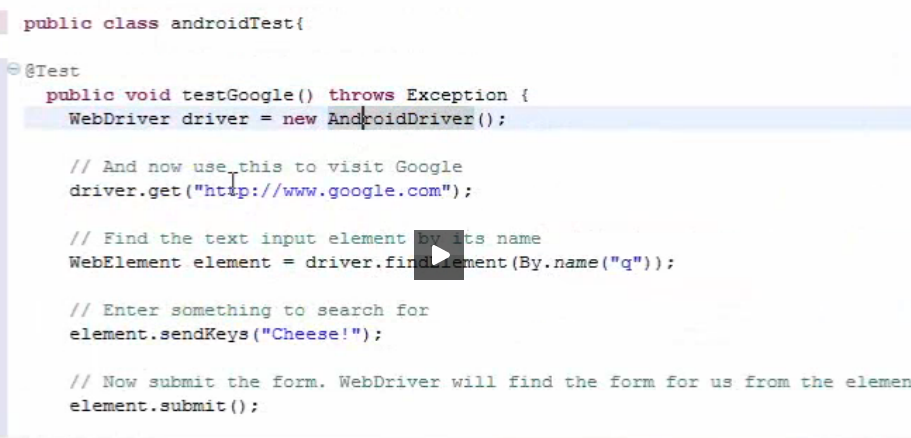
* **SDK manager –** Tools, latest version, previous version of Android and Extras needs to be checked and installed.
* There are 2 types of applications: web based apps and native apps.
* We use Android Device Bridge which is the connection to take the APK files and install on the virtual device.
* **Selenium Android driver :** visit <https://code.google.com/p/selenium/wiki/AndroidDriver> (reference as needed)
* **Download Eclipse. –** eclipse.org. Eclipse is an IDE, a development environment which will let you do coding on lot of technologies, primarily Java. It is also like a notepad on steroids. Extract to a folder.
* **Download latest JRE(java runtime environment) :** you need basic java to run java applications, require JRE or JDK to create executable files of java code and then run it. To run the final applications only java is required.
* **Download Selenium Server Jar file from** [http:**//**seleniumhq.o**rg**](http://seleniumhq.org) **-** selenium lets you write programs that you use to automate. Download 1) selenium IDE. It installs as a add-on on firefox browser. 2)Then install selenium RC web driver(zip) for java client(, copy the zip folder of web driver). 3)download Java binding. 4)Download jUnit jar file. Create a folder called selenium and the selenium server(jar file) from http://junit.org (download latest junit jar files)
* **Download the Android server apk from selenium hq (Android 2.21.0 wiki )from** <https://code.google.com/p/selenium/downloads/detail?name=android-server-2.21.0.apk>Put in the selenium folder. **Install this Android driver APK file** and **ADB install xxxx.apk from platform** tools under cmd prompt.(c/programfiles/android/androidsdk/platformtools – adb (android device bridge) from system to device .
* copy the installed Android driver apk to the same location where adb is
* so connect the device. Open cmd prompt and navigate to that location c:\Program Files<x86>\Android\android-sdk\platformtools
* type the command adb install android.apk ( this is that adroid driver apk file renamed to android) this will install the web driver onto the device.
* On the real device – goto settings apps – webdriver
* Now we need to set up the port forwarding in order to forward traffic from the host machine to the device. In a terminal type : go to cmd cd c:\Program Files<x86>\Android\android-sdk\platformtools ]
* This is giving transfer control protocol telling which port number that this interaction will happen.
* **Cmd is *adb forward tcp:8080 tcp 8080*** this will tell where my test will be executing from and where my device is present. If your device is on a different port in a local network or over the internet, you should be giving that ip address.
* Open Eclipse, start a new project. Add all the related jar files, get the basic web driver code and run the code from eclipse.
* File – new – java project – give project name and finish.
* Go to the project just created and expan to the src folder. Go to src folder, right click on it and say new java class and give a name.
* It will create a basic class. And it displays as public class androidTest {

} (public class and name is the name of the test.)

* Get a basic web driver code from Selenium recording. Take the Java Junit web driver.
* Add the corresponding jar files. Right click on the project, go to properties, java build path – libraries – add external jars – put all the 3 files in the selenium folder.
* Import wherever required to tell eclipse how to recognize



* The following is a sample code



* In the above code, webdriver driver = new AndroidDriver. This means we are creating a new Android driver in this code.
* Driver.get is a command to go to a specific url
* Then I’m finding a web element on the browser with a attribute called name value is q
* Run the code. It should open Google on the device.