**Appium Introduction**

1. Appium is an open source cross platform test automation tool for testing native, hybrid and mobile web apps.

2. Appium is supported for iOS, Android and Firefox OS.

3. Appium can be used on Simulators, Emulators and also on real devices.

4. Appium is an extended version of Selenium WebDriver. The features which are required for automating Mobile Apps are included in the WebDriver and the new version called Appium is created.

WebDriver + Mobile Features = Appium

5. Appium has inbuilt WebDriver features. WebDriver is subset of Appium.

6. There are three types of Mobile Apps. - Native, Hybrid and Mobile Web Apps.

Native: Pure mobile app. Cannot be switched to Webview. This app is specifically written in mobile platform.

Hybrid: Have a flexibility of switching to both native/mobile view and webviews.

Mobile Web App: Eg: Chrome, Firefox.

**Benefits of using Appium:**

1. No need to modify or recompile your app for automation APIs.

2. Can use any testing framework.

3. Can write code in any language supported by WebDriver.

For automating iOS apps Apple has provided library called Apple's UIAutomation library. Similarly we have Google's UIAutomator library for automating Android apps.

4. Appium library is a cross platform library which works for both iOS and Android.

5. Freeware.

**Internal Architecture of Appium:**

1. Appium follows Client/Server architecture.

Mobile Device where apps are present.

Appium Server

Automation Code (Editor)

Commands are sent to Appium server in the form of Json Wire Protocol (POST session request + json obj).

Triggers the invocation in the mobile.

Json Wire Protocol: Appium code can be written in any language. Json wire protocol sends that code to Appium Server as a POST session request with a json object called as Desired capabilities. This independent of any language.

Appium Server will have the capability to interpret the session object which is sent by the client(Editor).

Appium server will act as a proxy between the mobile device and the client libraries where we write automation code.

2. Desired Capabilities are used to declare the details of automation i.e. app, browser, OS etc.. Desired Capabilities object is provided to Appium Server in the form of http session object.

**Configure Appium and Android in Windows**

1. Download Android SDK from the link [Download SDK](http://developer.android.com/sdk/index.html).

2. Run .exe file of Android Studio.

3. Find the SDK location at C:/Users/<username>/AppData/Local/Android/SDK or any custom location in case if you have changed the default one.

4. Download Java from the link [Download Java](http://www.oracle.com/technetwork/java/javase/downloads/index.html).(Download JDK version based on system configuration).

In some cases without downloading java you may not be able to run Android Studio. In that case first download java and then Android Studio.

5. Set system variables path for both Java and Android so that your system detect these locations.

1. Go to jdk folder and copy the path till jdk. Go to Environment variables and create a new system variable called JAVA\_HOME and copied path as value.
2. Go to bin folder in jdk and copy path till bin folder. Go to Environment variables and add this value to the path variable in the system variables.
3. Cross check the setting by giving the command *java -version* in cmd.
4. Go to sdk folder and copy path till sdk. Go to Environment variables and create a new system variable called ANDROID\_HOME and copied path as value.
5. Inside sdk folder copy the paths of platform tools and tools and set both the values in path variable in the system variables.

6. Download Eclipse from [Download Eclipse](https://eclipse.org/downloads/) based on system configuration(64bit or 32bit).

7. Install ADT plug-in into eclipse following instructions at [Install ADT Plug-in to Eclipse](http://developer.android.com/sdk/installing/installing-adt.html).

8. Complete these settings in Eclipse once you configured ADT Plug-in t make it recognize: "*Open Custom Perspective and make Android SDK and AVD Manager enable in Toolbar and Command bar tabs and set the path of SDK location in Preferences window".*

9. Create Android Virtual Device which will be later used for testing.

10. Make sure DotNet Framework 4.5 is available on your system. [DotNet Framework 4.5](https://www.microsoft.com/en-in/download/details.aspx?id=30653).

11. Install Appium Server from the link [Download Appium](http://appium.io/downloads.html).(This automatically installs Node.js)

12. Download Appium and Selenium Jars from [Maven Repository](http://mvnrepository.com/).

13. Create Java Project in Eclipse by configuring all these jars in it.

Above mentioned steps 7,8,9 are for Android Emulator. In order to use Android device for automation, after following the above mentioned steps follow the below mentioned ones also.

For Automation with Android Device(Connecting Android Device to Computer):

1. Install PDANet in your computer. While installing PDANet it will ask you to connect your phone, after connecting PDANet will also be installed in your phone.

2. Make sure your mobile is connected to Wi-Fi and open PDANet application on mobile. Select USB Tethering.

3. Go to *Settings -> About Phone* and enable Developer mode by continuously tapping 7 times on the *Build Number*.

4. Proceed to *Settings -> Developer Options* and enable *USB Debugging*.

5. Open command prompt and enter command "adb devices". (You will see your device connected).

6. Install the apps App Info and App Backup & Restore on your mobile.

**First Android Program**

1. Download the app on your mobile and transfer it to your PC with App Backup & Restore app.

2. Open already created java project(by configuring Selenium and Appium jars) in Eclipse.

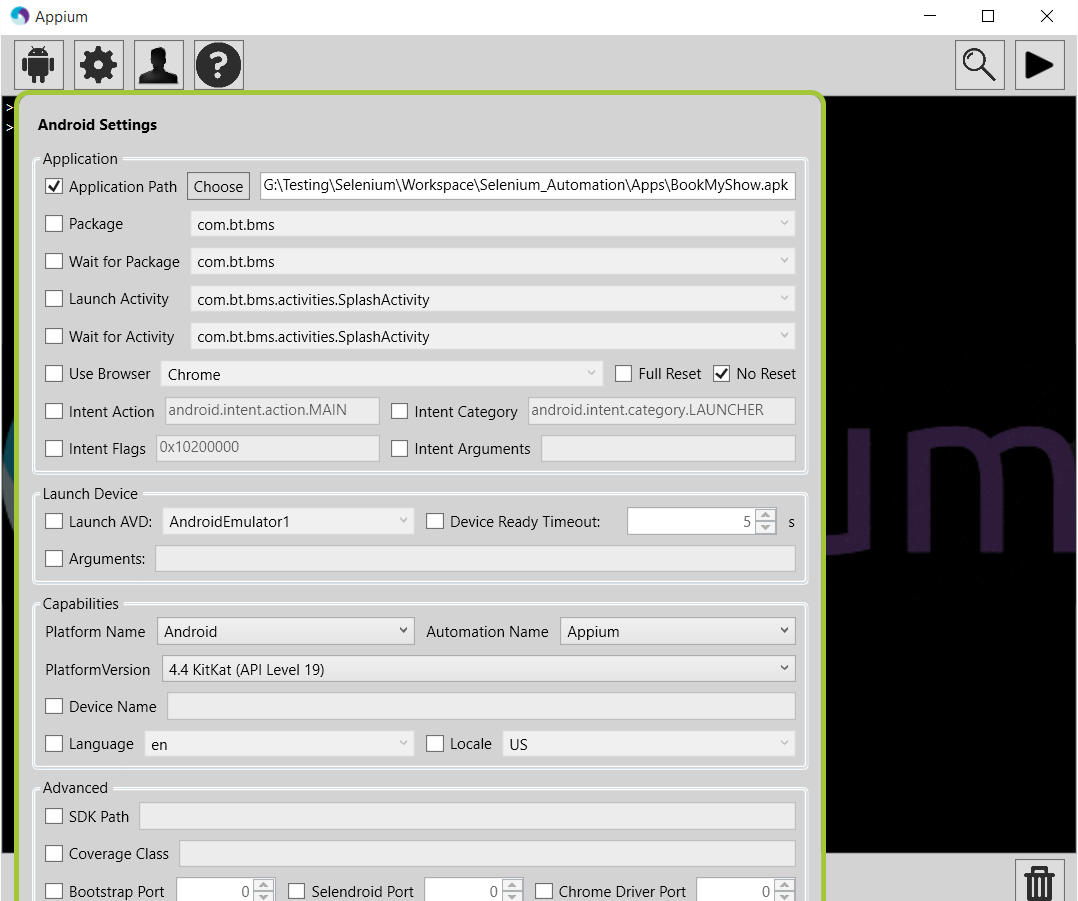
3. Open App Info app on mobile and select the app which is to be automated. Tap and hold on the app name to proceed to Detailed Information

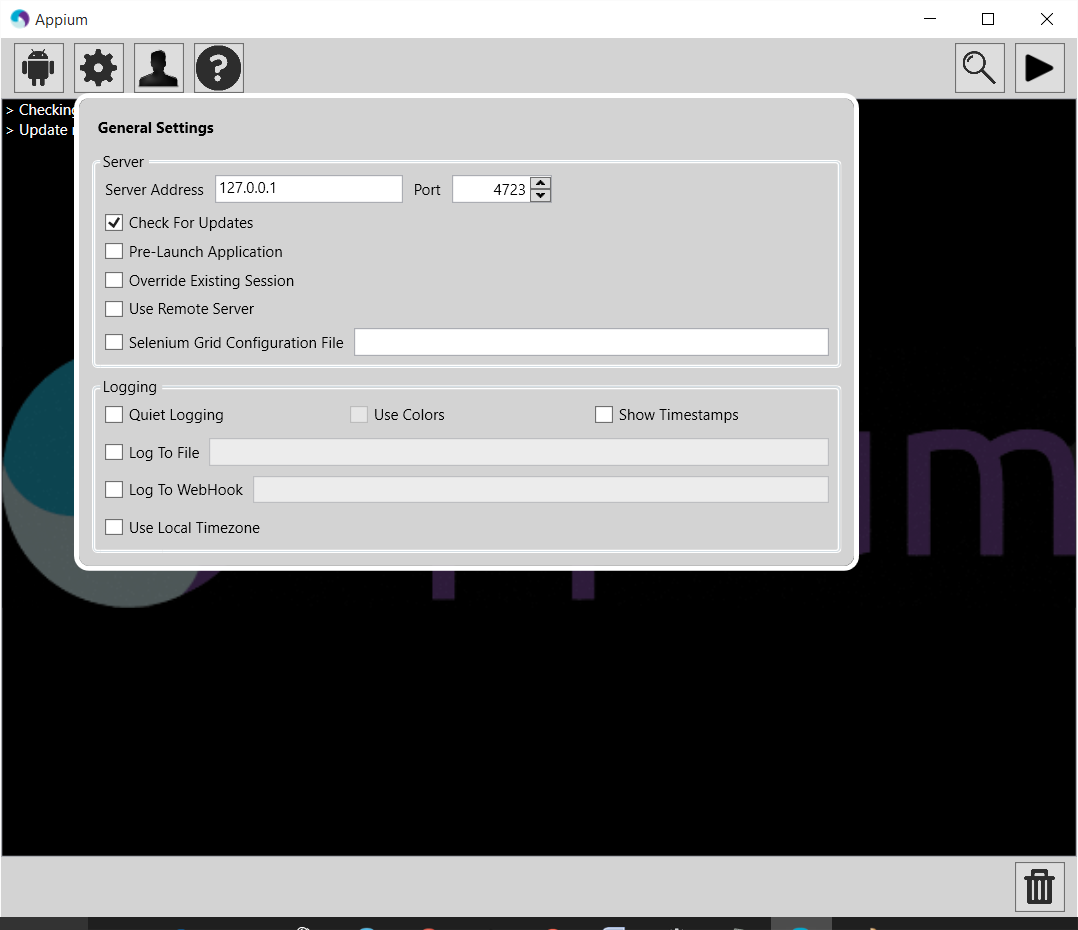
3. Copy the App package under APK Path section. For example if the APK path for the app is /data/app/**com.bt.bms**-2/base.apk, copy the highlighted text only.

4. Copy the App Activity under the App Activity section. (Note: You should copy the main activity or the first activity. For example: **com.bt.bms.activities.MainTabActivity**).

5. Open the Appium desktop application and configure the same.

1. Proceed to Android Settings and enter the path of the apk file and check the checkbox "*No Reset*".



1. Proceed to Settings and set the Server address and Port number.

6. Open Android Emulator or connect your Android device to computer by following above mentioned steps.

7. Desired Capabilities:

1. Desired Capabilities are options that are used to customize and configure a session

2. In order to open the app on either device or emulator you need to provide Desired Capabilities to the Appium Server. Below are the few mandatory Desired Capabilities that needs to be provided.

***Browser Name*** - Name of the browser to be launched on device/Emulator.

***Device Name*** - Name of the Android device/Emulator.

***Platform Version*** - Version of Android installed on device/Emulator.

***Platform Name*** - Android or iOS.

***App*** - Path where app is located in the system.

***App Package, App Activity*** - We already have these details from App Info.

Apart from the above there are many capabilities which can be referred at [List of Desired Capabilities for Appium](https://github.com/appium/appium/blob/master/docs/en/writing-running-appium/caps.md).

8. Below is the sample piece of code for opening the BookMyShow app on your device/emulator and clicking on a link.

