APPIUM Tutorial For Beginners

APPIUM is a freely distributed open source mobile application UI Testing framework.

Appium allows native, hybrid and web application testing and supports automation test on physical devices as well as on emulator or simulator both.

It offers cross-platform application testing i.e. single API works for both Android and iOS platform test scripts.

It has **NO** dependency on Mobile device OS. Because, APPIUM has framework or wrapper that translate Selenium Webdriver commands into UIAutomation (iOS) or UIAutomator (Android) commands depending on the device type not any OS type.

Appium supports all languages that have Selenium client libraries like- Java, Objective-C, JavaScript with node.js, PHP, Ruby, Python, C# etc.

In this tutorial we will learn about

APPIUM Design Concepts

- Appium is an 'HTTP Server' written using Node.js platform and drives iOS and Android session using Webdriver JSON wire protocol. Hence, before initializing the Appium Server, Node.js must be pre-installed on the system.
- When Appium is downloaded and installed, then a server is setup on our machine that exposes a REST API.
- It receives connection and command request from the client and execute that command on mobile devices (Android / iOS).
- It responds back with HTTP responses. Again, to execute this request, it uses the mobile test automation frameworks to drive the user interface of the apps. Framework like:-
  - Apple Instruments for iOS (Instruments are available only in Xcode 3.0 or later with OS X v10.5 and later)
  - Google UIAutomator for Android API level 16 or higher
  - Selendroid for Android API level 15 or less

Prerequisite to use APPIUM

1. ANDROID SDK [Link]-
2. JDK (Java Development Kit) [Link]
3. TestNG [Link]
4. Eclipse [Link]
5. Selenium Server JAR [Link]
6. Webdriver Language Binding Library [Link]
7. APPIUM For Windows [Link]
8. APK App Info On Google Play [Link]
9. Node.js (Not Required - Whenever Appium server is installed, it by default comes with "Node.exe" & NPM. It's included in Current version of Appium.)

APPIUM Inspector

Similar to Selenium IDE record and playback tool, Appium has an 'Inspector' to record and Playback. It records and plays native application behavior by inspecting DOM and generates the test scripts in any desired language. However, currently there is no support for Appium Inspector for Microsoft Windows. In Windows, it launches the Appium Server but fails to inspect elements. However, UIAutomator viewer can be used as an option for Inspecting elements.

Steps to start with Appium Inspector on Mac machine:-

Step-1 Download and start your Appium server with the default IP Address 0.0.0.0 and the port 4725.

1. Select the source file or .app files from local to test.
2. Check the 'App Path' Checkbox to enable 'Choose' button.

Now, clicking on 'Choose' button will give the option to browse and select test file from the local drive.

Step 3- Start Simulator on Mac machine.

Step 4- Click 'Launch' button from top right corner, which enable a blue color icon. Again, click on this blue color icon, it will open the Appium inspector and Simulator with pre-selected application.

Step 5- Launching your Appium Inspector will show the element hierarchy in column-wise structure. Also, user can apply actions using buttons like Tap, Swipe etc.

Step 6- Click on 'Stop' button to stop recording.

APPIUM Installation on Windows

Step 1- Install Android SDK in your system.

a. Go to Control panel >> System and Security >> System and from the left panel click on 'Advance System Settings'. From 'System Properties' pop up, click on 'Advance' tab and then click on "Environment Variables" button.

b. Now, from 'Environment variables' pop up, 'double click on 'Path' and set ANDROID_HOME variable that point to your SDK directory. In the path append the whole SDK folder path.

e.g. - C:\User\ABC\Desktop\adt-bundled-windows-x86_64-20140321\sdk

Step 2- Start your Android emulator or any attach any Android device to your system (Make sure you have Android Debugging option enabled in your Android device. To check Debugging Option. Go to Device Settings >> Developer Options >> Check "Debugging Option").

Step 3- Open Command Prompt and navigate to your Android SDK's \platform-tools\ directory (E.g. D:\adt-bundle-windows-x86_64-20130514\sdk\platform-tools).

Step 4- Run 'adb devices' command. You can see your connected device listed in Command Prompt window. (In CMD write '>adb devices'- This command will list the connected emulator instances. E.g.: adb –s emulator-5554 install <Location of .apk file>)

Step 5- Run 'adb start-server' command. It will start ADB server that will be used by Appium to send commands to your Android device.

Step 6- Now, navigate to Appium directory in your system and start Appium by clicking Appium.exe file.

Step 7- Do not alter the IP address or port number and click 'Launch' button. Your Appium console start at 127.0.0.1:4723 as shown in below.

Steps 8- Click on 'Start' button, Appium server started running on your system.

Your First APPIUM Test Case for Native Android App

Step 1) Download ADT eclipse plugin or download ADT bundled separately here

Step 2) Open Eclipse and Create a new Project >> Package >> Class

Step 3) Import Selenium library and TestNG inside that new project.

Step 4) Now Create a small test Program for 'Calculator.app' to sum two numbers.

Download the above Appium Code

Appium Server and Android Emulator from 'AVD Manager' and Click Run >> TestNG. Above program will run the 'Calculator.app' on selected emulator and Result displayed under Eclipse console using TestNG framework.

Limitations using APPIUM

1. Appium does not support testing of Android Version lower than 4.2
2. Limited support for hybrid app testing. eg: not possible to test the switching action of application from the web app to native and vice-versa.
3. No support to run Appium Inspector on Microsoft Windows.

Common Encountered Errors and Troubleshooting Steps in Appium

Error	Troubleshooting Steps
1. error:- The following desired capabilities are required, but were not provided:device Name, platformName	1. Add desired capabilities: device Name, platformName in APPIUM script. e.g:capabilities.setCapability("deviceName","Emulator"); capabilities.setCapability("platformName","Android");
2. error: Could not find adb. Please set the ANDROID_HOME environment variable with the Android SDK root directory path.	2. You probably need to set up SDK root directory path in system 'Environment Variables' in 'Path' column
3.error:org.openqa.selenium.SessionNotCreatedException: A new session could not be created.	3. You need to set a correct App path and restart the Appium server.
4. How to find DOM element or xPath in mobile application?	4. Use 'UIAutomatorviewer' to find DOM element for Android application.