**Appium-**

**Mobile Application Testing Automation**

**Primary Author**

Name : Malathesh Jalagar

Linked In profile : https://www.linkedin.com/in/malathesh-jalagar-46b721b3/

FB profile link : <https://www.facebook.com/malathesh.jk?ref=bookmarks>

You are allowed to add images, code snippet, diagrams etc. If you are referring any image or material from the internet please do not miss on giving due credits and references. The font is Times New Roman , size 10 Point. Please do not change the font or font size.) All images have to saved in either .JPG or .PNG format and shared as separate files. One zip file with your name and this word file along with all the images should be uploaded on the registration page. Please do not delete this para

**Blog**: (Appium-Mobile Application Testing Automation).

I would like to share my experiance on working with appium automation testing tool where I have worked on mobility testing of the app.

**Overview**:

Appium is an open source, cross-platform test automation tool for native, hybrid and mobile web apps, tested on simulators (iOS, FirefoxOS), emulators (Android), and real devices (iOS, Android, Windows, FirefoxOS).

**Supported Platforms**

• iOS

• Android

• Windows

• FirefoxOS

**Why Appium**

1. No need to recompile mobile application or modify it in any way as it uses standard automation APIs on all platforms.

2. Write tests with your favorite development tools using any WebDriver-compatible language such as Java, Objective-C, JavaScript with Node.js (in promise, callback or generatorflavors), PHP, Python, Ruby**, C#**, or Perl with the Selenium WebDriver API and language-specific client libraries.

3. You can use any testing framework Requirements

**Requirements :**

1.      To run Appium via an npm install, you will need node.js and npm 4 or greater (use [n](https://github.com/visionmedia/n) or brew install node to install Node.js. Make sure you have not installed Node or Appium with sudo, otherwise you’ll run into problems). We recommend the latest stable version.

2.      To verify that all of Appium’s dependencies are met you can use appium-doctor. Install it with npm install -g appium-doctor, then run appium-doctor and supply the --ios or --android flags to verify that all of the dependencies are set up correctly.

3.      You also need to download the Appium client for your language so you can write tests. The Appium clients are simple extensions to the Web Driver clients,

Here, I have used C# driver for the appium using Nugget package manger console in VS by typing command: Install – Package Appium.WebDriver, it will automatically download all the references to project.

4.      Appium app to be installed on MAC machine

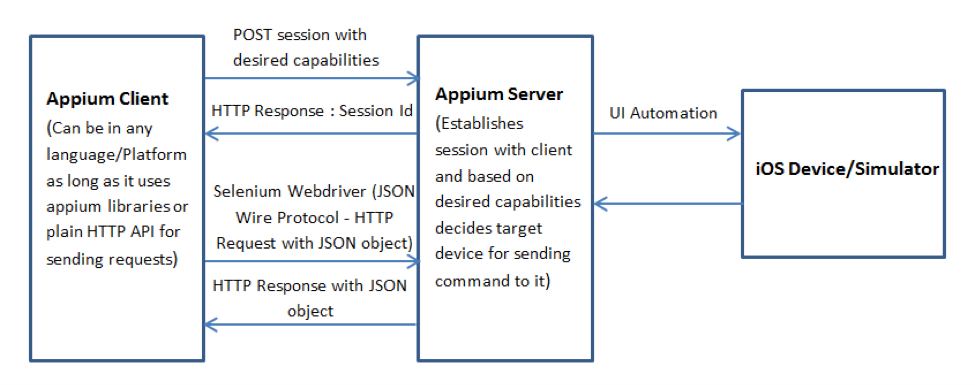
i**OS Requirements**

• Mac OS X 10.10 or higher, 10.11.1 recommended

• XCode >= 6.0, 7.1.1 recommended

• Apple Developer Tools (iPhone simulator SDK, command line tools)

**APPIUM CLIENT-SERVER ARCHITECTURE:**



•  We have established the Client server communication between windows and MAC machine

•  Appium app is installed in the MAC machine (Server) and in the windows we will be writing test cases (Client). And we access server from windows by Local IP address of MAC machine.

• Once connection is established based on desired capabilities, appium server starts sending commands to ios devices/ simulator

**Type of tests suitable for Appium**

• Appium is pretty handy for testing sceneries that users will go through when using app

•    Appium can perform UX interactions and Functional testing

**Setup process**

• Install latest version of Appium app in MAC machine

•  Follow the below commands to install appium server in the Mac machine. Type bellow commands in Terminal

> brew install node                       *# get node.js*

> npm install -g appium                   *# get appium*

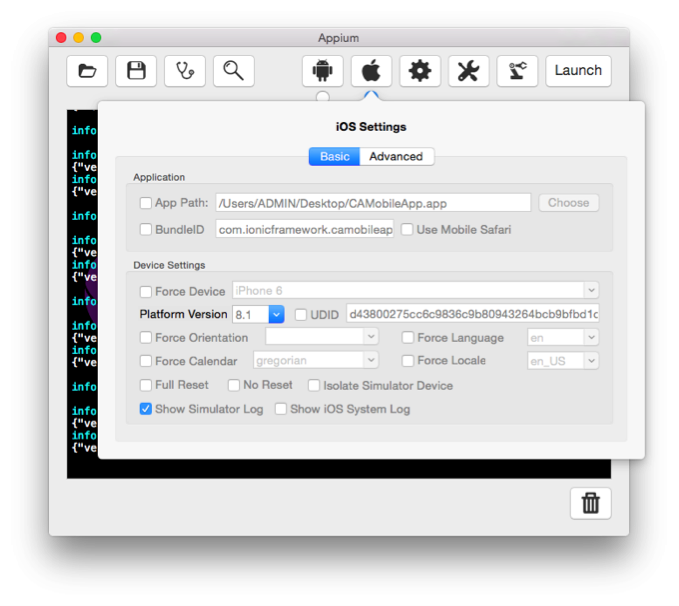
> npm install wd                          *# get appium client*

> appium &                                *# start appium*

•  Once appium $ command is called , Server will start and listing for appium commands at the port 4723 by default

Appium Server Setting For Simulator

• Verify that no checkbox is checked in iOS settings and General settings on the Appium server

•   
  
  


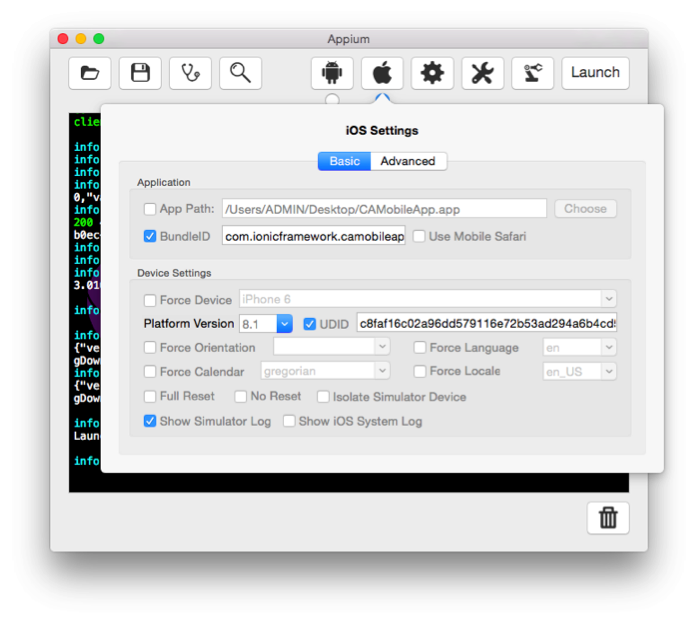
• Setting Local host as the IP address and Default port number has 4723 in General setting

**Appium Server Setting For Real Device**

]

Verify that both the “UDID” and “Bundle ID” checkboxes are checked in iOS settings (correct details are provided in corresponding text boxes) and no checkbox is checked in General settings on the Appium server:

•



**Setting up Desired Capabilities**

To initialize an Appium session you need to define certain parameters known as “desired capabilities” like PlatformName, PlatformVersion, Device Name and App path and so on.

•

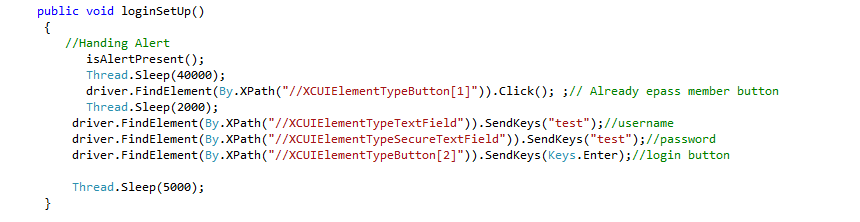
**Writing Test cases**

• Once installation is complete on both machines, you can start writing test cases for the app. In current example, Visual studio console application has been used.

• While writing test cases make sure you have correct XPath for the elements in the app using inspector in the appium app.

• Based on the workflow you can write test cases for the app, below image has simple example for log in for a sample app

•

Here is the another example for signup task in a sample app

•

• Finding elements in the app by xpath, which we can get it from appium inspector. Based on the type of element, you can pass the text using **sendkeys (“***some text***”)** method or **click ()** if it is button, and finally we can **assert** it is meets the test cases

**Running the Test**

Launch Appium server on Mac machine

•

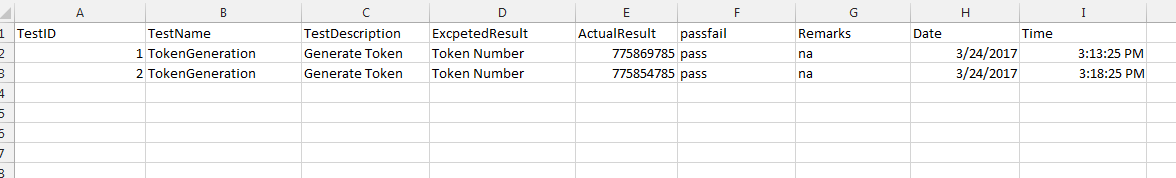
 Run the automation scripts on a Windows machine

2. Verify that the logs started appearing on Appium, that the app is launched on a Simulator/Device, & tests as written in scripts are being executed.

3. Verify the execution result in Visual Studio after all scripts are executed.

4. Generate the reports for the test cases based on the result. It can be customized based on the requirement

•

                 Fig- Sample test report

Thank you all for reading, share your comments/feedback.