**Installation of Appium in Windows for Android alone :**

# **Requirements to install appium in mac for Android & iOS**

1. **Java Development Kit (JDK)**
2. **Android SDK**
3. **Node.js**
4. **Microsoft Webdriver**
5. **PDANet+**
6. **Appium**
7. **ADT Plugin**
8. **Java Client Drivers**
9. **Appium Client Libraries**

**Steps :**

## **1. Download and install Java (JDK) and set path of jdk and bin folder**

a) Download the “.exe” file from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>(Version: jdk1.8.0\_151 or whichever is the latest you find there.)

b) Install the “.exe” file.

c) Set JDK folder path in your system’s environment variable.

* The JDK software is installed on your computer, for example, at C:\Program Files\Java\jdk1.8.0\_151
* Right click My Computer and select Properties.
* On the Advanced tab, select Environment Variables, and then edit JAVA\_HOME to point to where the JDK software is located, for example, C:\Program Files\Java\jdk1.8.0\_151

d) Set JDK bin folder path in your system’s environment variable.

1. Click Start, then Control Panel, then System.
2. Click Advanced, then Environment Variables.
3. Add the location of the bin folder of the JDK installation for the PATH variable in System Variables. The following is a typical value for the PATH variable:
4. C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files\Java\jdk1.7.0\bin

**Refer below link to know about jdk setup:**

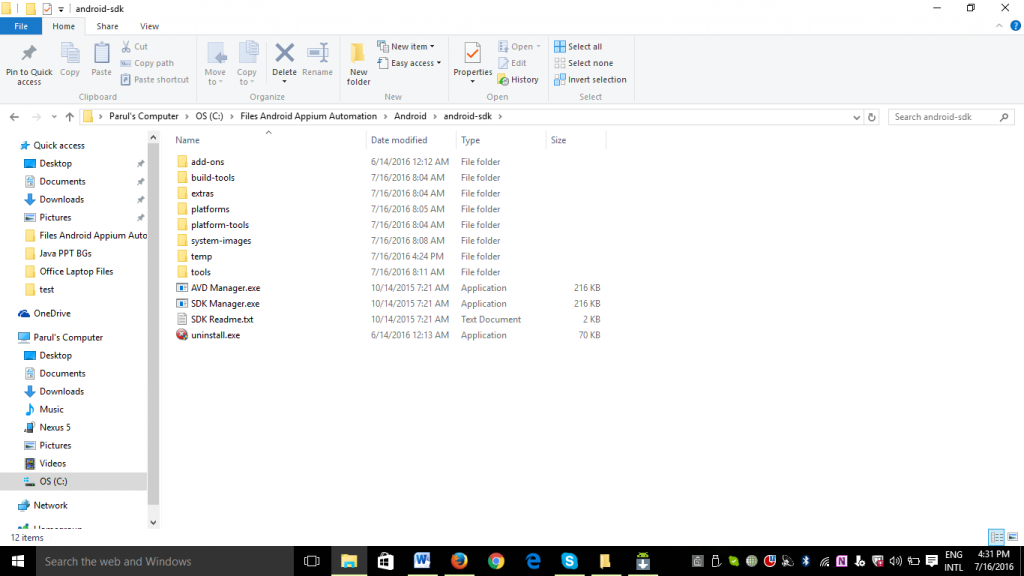
[**http://www.software-testing-tutorials-automation.com/2015/09/steps-to-download-and-install-javajdk.html**](http://www.software-testing-tutorials-automation.com/2015/09/steps-to-download-and-install-javajdk.html)

## **2. Download and install Android SDK**

a) Download Android SDK from <https://developer.android.com/studio/index.html>

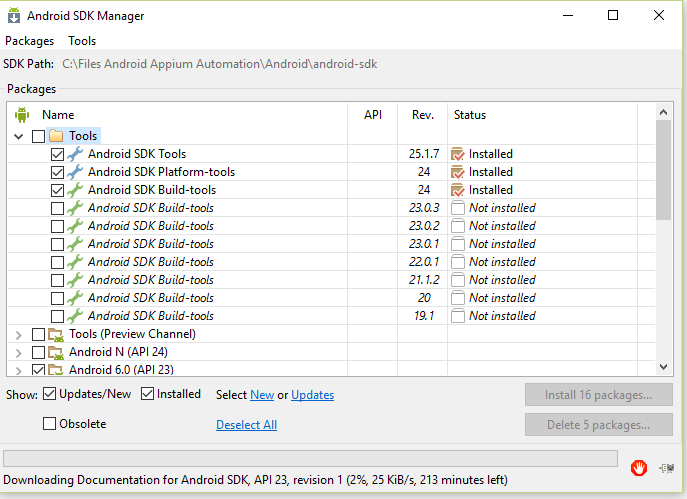
b) Click on the link “android-sdk\_r24.4.1-windows.zip” (or whichever is the latest you find there) and then click on the download button.

c) Once a zip file gets downloaded, unzip the folder. You will get the following list of folders inside the Android SDK.

****

d) Now click on the “SDK Manager.exe” file.

e) This opens the Android SDK Manager window. Select the first 3 packages under “Tools”, select the package under Tools (Preview Channel) and finally the Android platform based on the platform of the device on which you will be performing your tests

****

Only after all the selected packages are installed you would be able to write and execute your test scripts.

## **3. Set Android SDK path in Windows**

After all the packages are successfully installed, you need to set the SDK path.

* Click on Environment Variables -> Create a new user variable ANDROID\_HOME -> and set the Android SDK path as the value for it. (e.g.: C:\Files Android Appium Automation\Android\android-sdk)
* Set 2 paths in the system variable Path:
* Path of “platform tools” folder in the SDK (e.g.: C:\Files Android Appium Automation\Android\android-sdk\platform-tools)
* Path of “tools” folder in SDK (e.g.: C:\Files Android Appium Automation\Android\android-sdk\tools)
* Now, to check whether or not Android is configured properly in your system, run command “android” in the command prompt. This will open the Android SDK Manager dialogue box, which verifies successful configuration of Android in your system.

## **4. Node.js**

Follow these steps:-

a) Go to the link <https://nodejs.org/en/download/>

b) Click on the Windows Installer tab.

c) Download starts.

d) Install it.

## 5. Microsoft Webdriver

Follow these steps:-

a) Go to the link <https://www.microsoft.com/en-us/download/details.aspx?id=48212>

b) Click on the Download link on this page.

c) Install it.

## **6. PDANet+**

To download and install PDANet+ on your machine, follow these steps:-

a) Go to the link [pdanet.co/](http://pdanet.co/)

b) Download and install it.

## **7. Appium**

This is a test automation tool for mobile applications.

a) Go to the link [appium.io](http://appium.io/)

b) Click on “AppiumForWindows.zip” link from here.

c) Download starts.

d) Unzip the downloaded zipped folder.

e) Install the exe file “appium-installer”.

## **8. ADT Plugin**

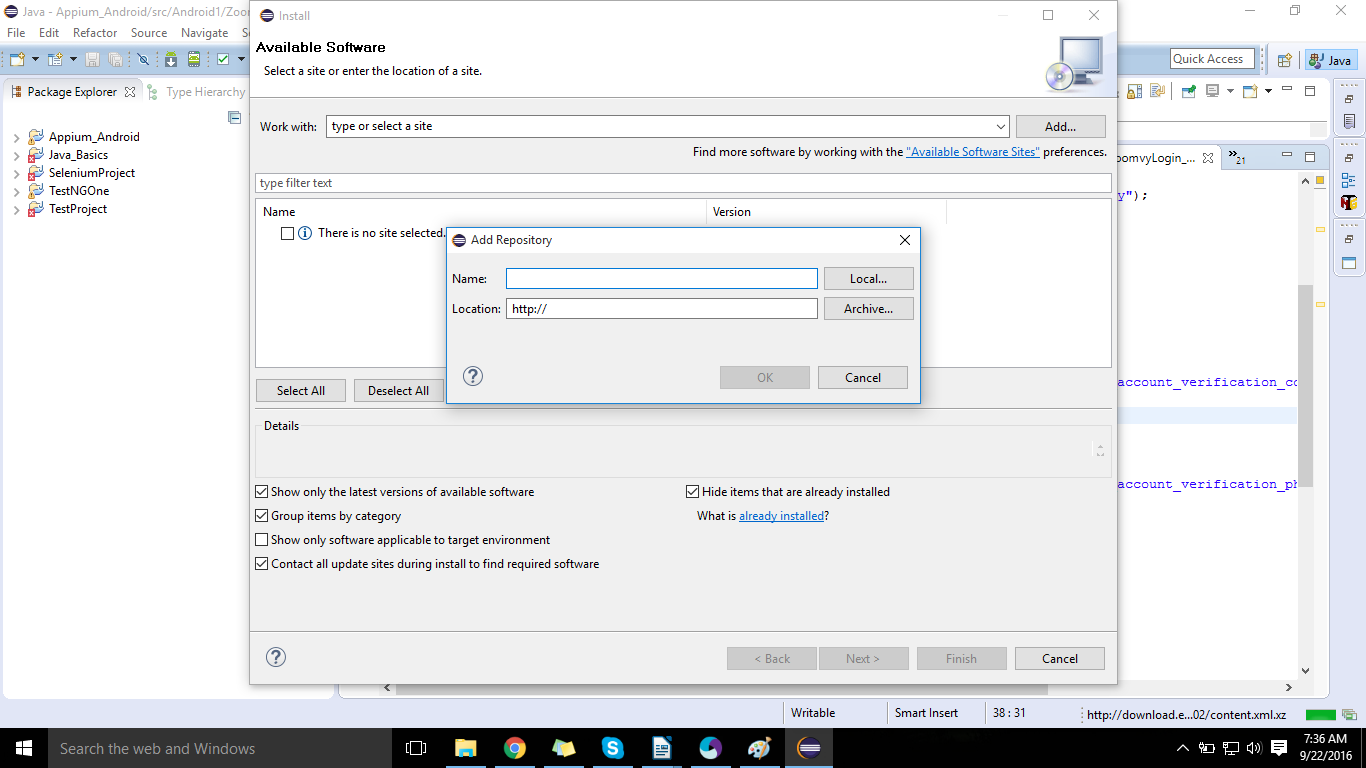
It is used to access Android SDK within Eclipse. ADT Plugin tool is basically used to launch Android Emulators in Eclipse using the test script that we have created. By installing this plugin you can basically launch AVD Manager from Eclipse.

-**Steps to install ADT Plugin in Eclipse:**

a) Open Eclipse IDE.

b) Go to Help -> Install New Software. This will open Install Software dialogue.

c) Click on the add button here. This will open a new dialogue.

****

d) Set URL “https://dl-ssl.google.com/android/eclipse/” in location text box and click on OK button.

e) It will open Developer tools option along with a checkbox.

f) Select it and click on the Next button.

g) Click on the next button from here, accept the Terms & Conditions and click on the Finish button.

h) This will start installing the Plugin.

i) After successful installation, restart Eclipse.

**Set SDK Location:** You need to set the SDK folder path after installing ADT Plugin

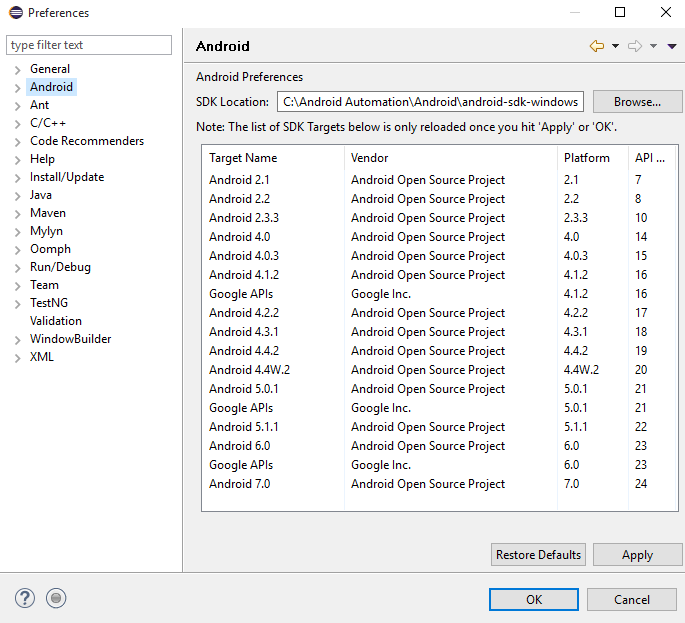
a) Open Eclipse.

b) Go to Window -> Preferences

c) Select “Android” from the list on the left

d) Set SDK folder path in the “SDK Location” box.

e) Click on “OK” after this.

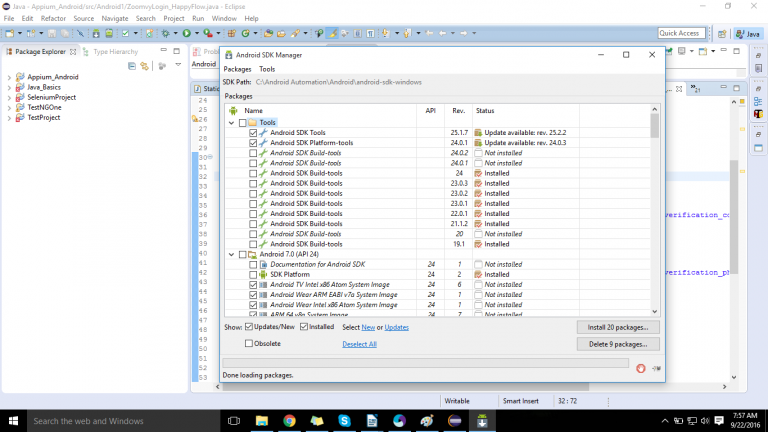
****

**To verify if Android SDK is integrated properly or not**

a) Open Eclipse IDE.

b) Go to Window -> Android SDK Manager.

c) This will open the Android SDK Manager dialogue.

****

**This verifies that Android SDK has been properly integrated with Eclipse IDE.**

## **9.Creating Appium Scripts**

**Now, before you can begin creating and running your own test scripts using Appium, there are some configurations that you need to make for your Java Project in Eclipse.**

**Eclipse Configuration with Selenium WebDriver:** This is needed for the interaction between your test scripts and the Selenium WebDriver. For the same, you will be needing language-specific client drivers. Since we will be working on Java, we will be needing the Java client drivers.

a) Go to this link: <http://docs.seleniumhq.org/download/>

b) Click on the Download link here for the latest JAR available (selenium-2.53.0 in my case)

c) Extract the downloaded zipped folder

d) Open Eclipse IDE

e) Create a new Java Project -> Create a new package under this project -> Create a new class under this package

f) Right click on your Project name -> Select Build Path -> Select Configure Build Path

g) Click on “Add External JARs” button -> go to the path where you had saved the Selenium WebDriver zip folder (e.g. in my case: C:\Android Automation\selenium-java-2.53.0\selenium-2.53.0)

h) Select both .jar files from here.

i) Now select all the .jar files inside the libs folder here.

**Don’t close the Properties Dialogue, since there are a few more JARs which need to be added to your project.**

## **10. Eclipse Configuration with Appium**

a) Go to this link: <http://appium.io/downloads.html>

b) Click on the Java link under the Appium Client Libraries section.

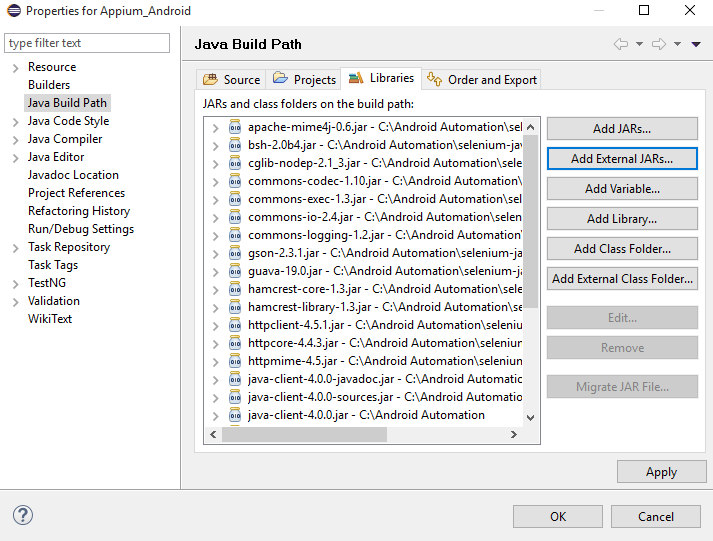
c) Click on the JAR link here.

d) Download starts.

e) Follow the same steps as above to import Appium client libraries into your Eclipse project.

f) Click on OK after importing all the JAR files.

So this is how it’ll look:

****

## **11.Sample Script**

* Create a new project in eclipse by clicking on **File->New->Java Project**
* Enter project name(Eg: sample project)
* Right click on the created project and create new package (Right click on **Sample Project -> New -> Package**) - Enter a new name for package **(eg : pack)**
* Create a new class inside the package- Right click on package (Right click on package **Pack -> New -> class/TestNG class**)

**Now type you appium scripts in the class and run it as Testng or normal java application**

**package pack;**

**import io.appium.java\_client.android.AndroidDriver;**

**import java.io.File;**

**import java.net.MalformedURLException;**

**import java.net.URL;**

**import java.util.concurrent.TimeUnit;**

**import org.openqa.selenium.remote.CapabilityType;**

**import org.openqa.selenium.remote.DesiredCapabilities;**

**public class StartApplication {**

**private static AndroidDriver driver;**

**public static void main(String[] args) throws MalformedURLException, InterruptedException {**

**File classpathRoot = new File(System.getProperty("user.dir"));**

**File appDir = new File(classpathRoot, "/Apps/Amazon/");**

**File app = new File(appDir, "in.amazon.mShop.android.shopping.apk");**

**DesiredCapabilities capabilities = new DesiredCapabilities();**

**capabilities.setCapability(CapabilityType.BROWSER\_NAME, "");**

**capabilities.setCapability("deviceName", "Micromax A311");**

**capabilities.setCapability("platformVersion", "4.4.2");**

**capabilities.setCapability("platformName", "Android");**

**capabilities.setCapability("app", app.getAbsolutePath());**

**capabilities.setCapability("appPackage", "in.amazon.mShop.android.shopping");**

**capabilities.setCapability("appActivity", "com.amazon.mShop.home.HomeActivity");**

**driver = new AndroidDriver(new URL("http://127.0.0.1:4723/wd/hub"), capabilities);**

**driver.manage().timeouts().implicitlyWait(80, TimeUnit.SECONDS);**

**Thread.sleep(10000);**

**driver.quit();**

**}**

**}**

Installation of appium in mac for both Android & iOS

# Requirements & setup

**1.Java**

**2.Eclipse**

**3.Android SDK**

**Setup:**

## **1.Java environment Setup**

* Open terminal and type **“emacs .profile”** - emacs editor will be opened

Enter the following commands in terminal

**Export JAVA\_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0\_151.jdk/Contents/Home**

**Export PATH=$JAVA\_HOME/bin:$PATH**

Now we want to save the emacs terminal editor. Press **(Command+X , Command+S)**

-Another method to set home variable

Type following in terminal **Vim .bash\_profile**

Export all the home path in .bash\_profile & save it.

## **2.Set Android sdk ,tools & platform tools- environment variable**

Type following in terminal **Vim .bash\_profile**

Export all the home path in .bash\_profile & save it.

**export ANDROID\_HOME=/Users/aravindhakumar/Library/Android/sdk**

**export PATH=/Users/aravindhakumar/Library/Android/sdk/tools:$PATH**

**export PATH=/Users/aravindhakumar/Library/Android/sdk/platform-tools:$PATH**

Download all the sdk platform tools via android studio:

## **3.Install external dependencies**

Link to know more details on external dependencies:

<https://github.com/appium/appium-xcuitest-driver>

**Commands to install external dependencies:(Open terminal & enter following commands to install dependencies)**

* **brew install libimobiledevice –HEAD**
* **brew install ideviceinstaller**

**If it is already installed in your machine, update the required item as mentioned in the terminal window**

* **brew upgrade ideviceinstaller**
* **brew install carthage**

**If it is already installed in your machine, update the required item as mentioned in the terminal window;**

* **brew upgrade carthage**
* **npm install -g ios-deploy**

**You should get \*\* BUILD SUCCEEDED \*\* information in your terminal window once run the above command.**

**gem install xcpretty**

## **4.Configuring WebDriver Agent in Mac**

**What is WebDriver Agent?**

WebDriverAgent is a WebDriver server implementation for iOS that can be used to remote control iOS devices. It allows you to launch & kill applications, tap & scroll views or confirm view presence on a screen. This makes it a perfect tool for application end-to-end testing or general purpose device automation. It works by linking XCTest.framework and calling Apple’s API to execute commands directly on a device.

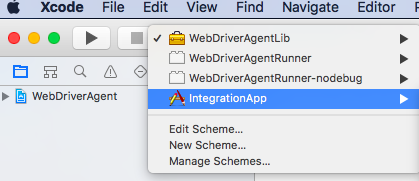
**Refer the below URL to learn about WebDriver agent;**

[**https://github.com/facebook/WebDriverAgent**](https://github.com/facebook/WebDriverAgent)

Please find the below location for accessing the webdriver agent available in your Mac;

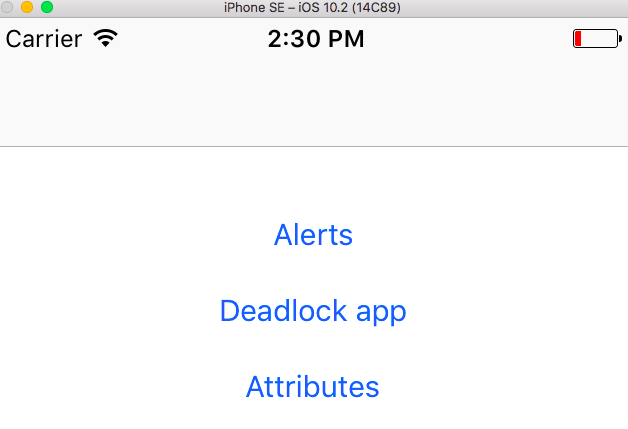
/Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent

Open the WebDriverAgent.xcodeproj using XCODE and ensure WebDriverAgentLib, WebDriverAgentRunner and IntegrationApp showing as expected based on the below screenshot;



Select Integration App and iPhone SE as Simulator and build the code by clicking the Play button;

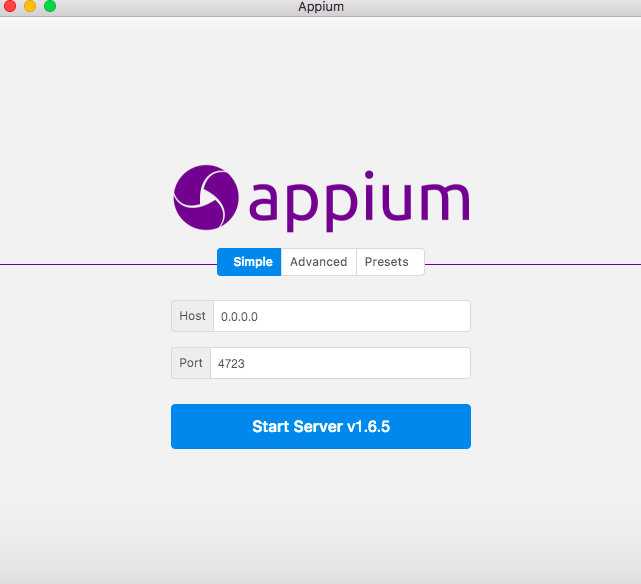
You should see the “Build Succeeded” message in the XCODE for the webdriveragent project and iPhone SE Simulator open with the webdriver agent application as mentioned below;



Step 4:

Your installation task got completed, now we need to set the required capabilities in the appium server and start inspecting the elements followed by writing the automation scripts;

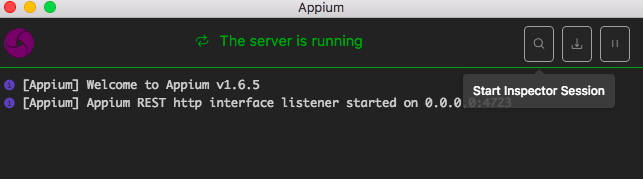
Start the server by clicking the Start Server v1.6.5 button as mentioned below;



Ensure server started message as follows in the console window;

[Appium] Appium REST http interface listener started on 0.0.0.0:4723

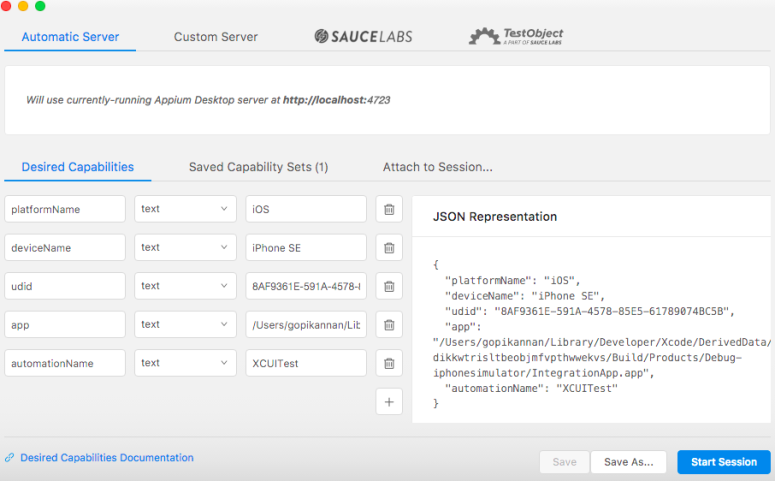
Click Start Inspector Session as mentioned below;



Please provide the desired capabilities information as mentioned in the below screenshot;

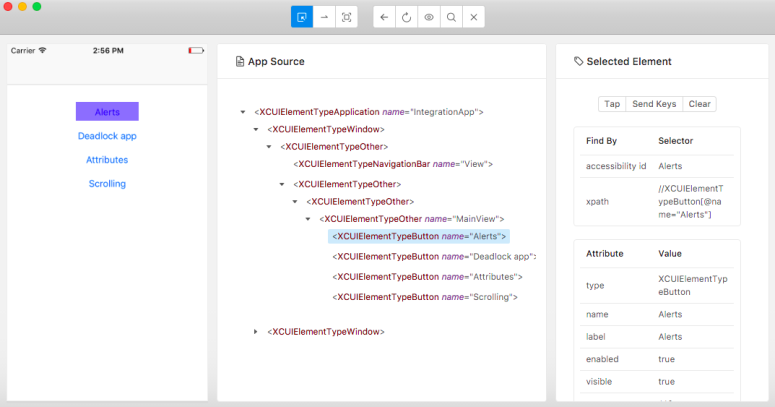
IntegrationApp.app file location should be taken from the XCODE

automationName parameter is one of the important stuff to handle the latest iOS 10.2 and above including the latest XCODE 8.2



Click Start Session button in the bottom right of the window and click the Allow button in the dialog box;

You can see the inspector window as mentioned in the below screenshot;



Now you can use the above desired capabilities in your script prepared by using the Eclipse IDE.