**Requirements**

**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.**

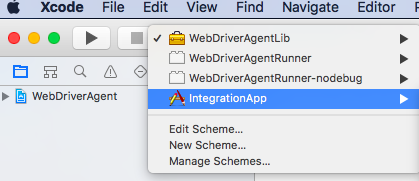
**Refer the below URL to learn about WebDriver agent;**

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

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

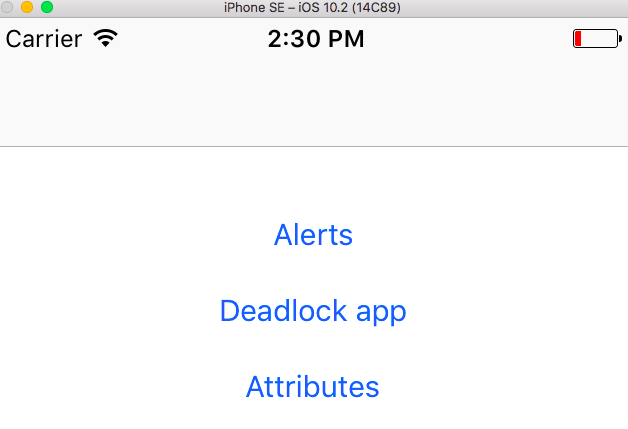
**/Applications/Appium.app/Contents/Resources/app/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent**

**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;**

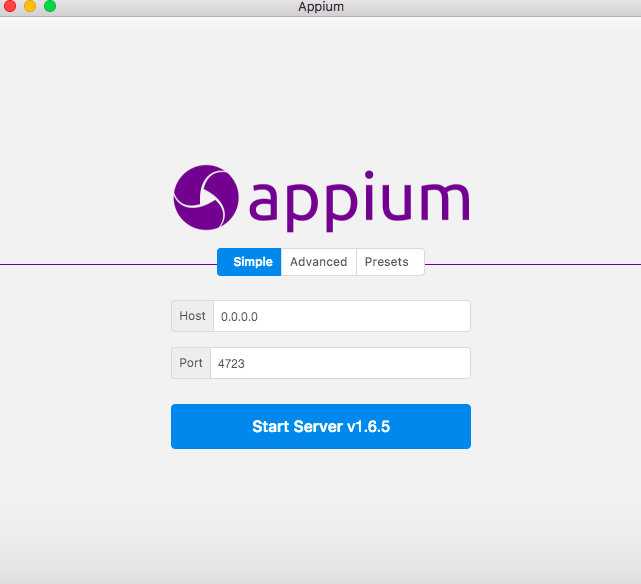
**You should see the “Build Succeeded” message in the XCODE for the webdriveragent project and iPhone SE Simulator open with the webdriver agent application as mentioned below;**

****

**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;**

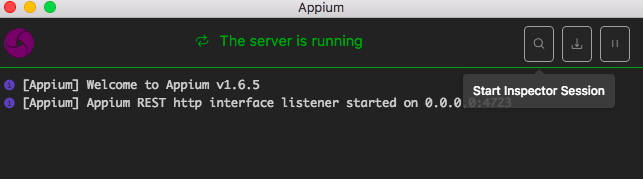
**Start the server by clicking the Start Server v1.6.5 button as mentioned below;**

****

**Ensure server started message as follows in the console window;**

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

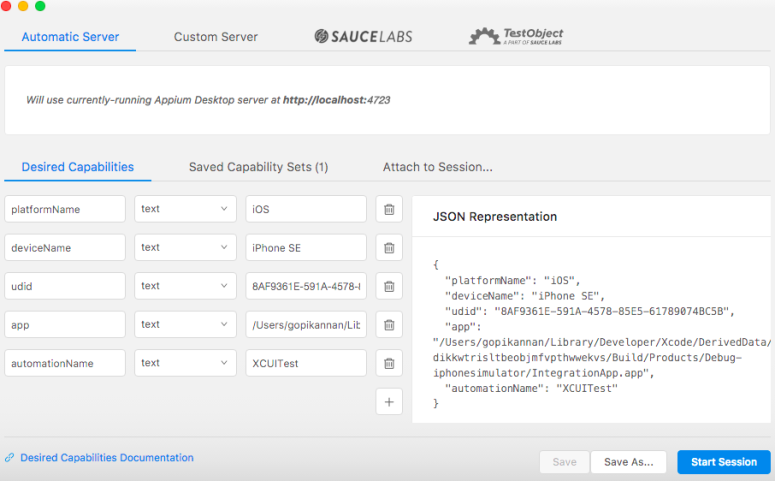
**Click Start Inspector Session as mentioned below;**

****

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

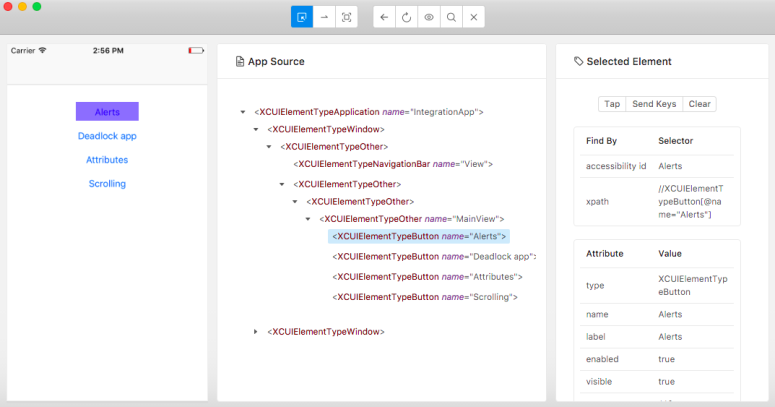
**IntegrationApp.app file location should be taken from the XCODE**

**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.**