IOS Automation Setup - Appium

MAC Setup

Below mentioned are the tools required for Appium set up in MAC Machine

- MAC OS(10.12.6) - (Minimum Required OS)
- JDK (Java development Kit)
- Eclipse
- Appium (Any latest version)
- Appium Java client jar(4.1.2)
- IOS Device (10.3) (minimum requirement)
- Xcode (9.1)
- Node v4 or above (node -v) - (required only if you are using old appium server and in latest appium server this come pre-installed)
- Selenium jars

Download or ensure above mentioned tools present and update bash_profile file

Java Installation:

a. Download, install and configure Java/JDK (**JDK**)[Source: https://www.java.com/en/download/mac_download.jsp]
b. Setting the **JAVA_HOME** system variable:

on Mac, should update the **bash_profile** file with JAVA_HOME and PATH variables:

- vim ~/.bash_profile
- Include the following content into the file

Sample Bash Profile

Expand source

```
export M2_HOME=${HOME}/Lib/apache-maven-3.3.9
export PATH=$PATH:$M2_HOME/bin
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_102.jdk/Contents/Home
export PATH=${JAVA_HOME}/bin:$PATH
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=${PATH}:$ANDROID_HOME/tools:$ANDROID_HOME/platformtools
export PATH=/usr/local/bin:$path
export PATH=/usr/local/git/bin:/usr/bin:/usr/local/sbin:$PATH
```

- Save the file (<Esc> :wq)
- source your bash or zsh profile file after adding/saving to take the effect of updating the bash profile.
  source ~/.bash_profile

c. Test:

#1: echo $JAVA_HOME   Output: /Library/Java/JavaVirtualMachines/jdk1.8.0_102.jdk/Contents/Home

#2: type java -version Should give the installed Java version details

- if bash_profile does not exist
  - then create a new bash_profile file by typing in terminal > touch ~./bash_profile

  - As mentioned above, you can also use vi or vim editor for creating/updating the bash_profile

Procedure to install Home brew, Node & NPM through terminal

Step:1 Install Ruby, Homebrew

First, install Homebrew by typing below mentioned command in terminal:

ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Note:- Before Running Above command ensure ruby is installed in your system

Then, run brew update to ensure your **Homebrew** is up to date.

brew update

To validate the Brew installation, run brew doctor and follow the proposed recommendations from brew doctor.

brew doctor

Adding Homebrew to the $PATH:
export PATH=/usr/local/bin:$PATH

On Mac, the bash or zsh profile file should be updated by adding the above line & saving the bash profile file. But, to take the effect of updating the bash profile, need to source the file with the below CMD.
source ~/.bash_profile

Step:2 Install Node (npm will be installed with node) by typing below command in terminal:

brew install node

Step 3: Install APPIUM

Install Appium through terminal by typing below command; the latest version of appium will be installed

npm install -g appium

Step 4: Install web driver

Now also run below mentioned command(you might need to SUDO)

npm install wd

Note: Might need to sudo for the above cmd if current user is not the root user; but, Appium might not function correctly if it is sudo used. Need rerun above installations as a non-root user. If you had to install Appium using `sudo npm install -g appium`, the solution is to reinstall Node using **Homebrew**, that doesn't require sudo to install global npm packages.

Note: Incase if Appium is not launched after installing appium through terminal, you can download the UI version directly by navigating to link https://github.com/appium/appium-desktop/releases/tag/v1.2.7 and clicking .dmg file

- Incase if you want to **uninstall Appium** use below mentioned command

  sudo npm uninstall -g appium

Other Needed Dependencies:

Run below commands in terminal:

1. For installing ios deploy:

npm install -g ios-deploy
OR
brew install ios-deploy

2. For installing libimobiledevice

brew install libimobiledevice --HEAD

3. For installing carthage

brew install carthage

4. Other Needed setup:

Open Terminal at WebDriverAgent in Appium directory folder:

<…>/appium/node_modules/appium-xcuitest-driver/WebDriverAgent and enter following command to initialize WebDriverAgent project:

- First locate where appium is installed it would usually be located in Applications,
- Navigate to below mentioned path
  /Applications/Appium.app/Contents/Resources/app/node_modules/appium/node_modules/appium-xcuitest-driver/WebDriverAgent
  To navigate to the above path go to Applications -> Right click on Appium -> click on "**Show Package contents**" and navigate to above path

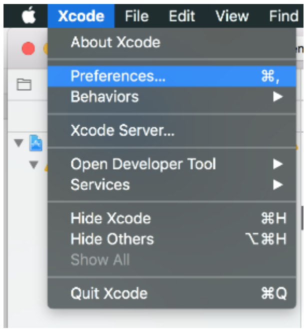
```
mkdir -p Resources/WebDriverAgent.bundle
sh ./Scripts/bootstrap.sh -d
```

Code Signing WebDriverAgent App:

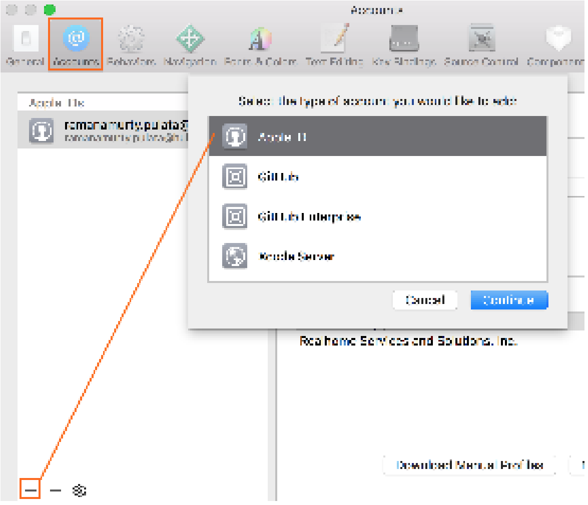
Pre-requsite:

ios developer account to be update in Xcode as given below:

1. Open Xcode
2. Under the Xcode menu select "Preferences"



3. Under **Accounts** tab click on **+** at the bottom left corner



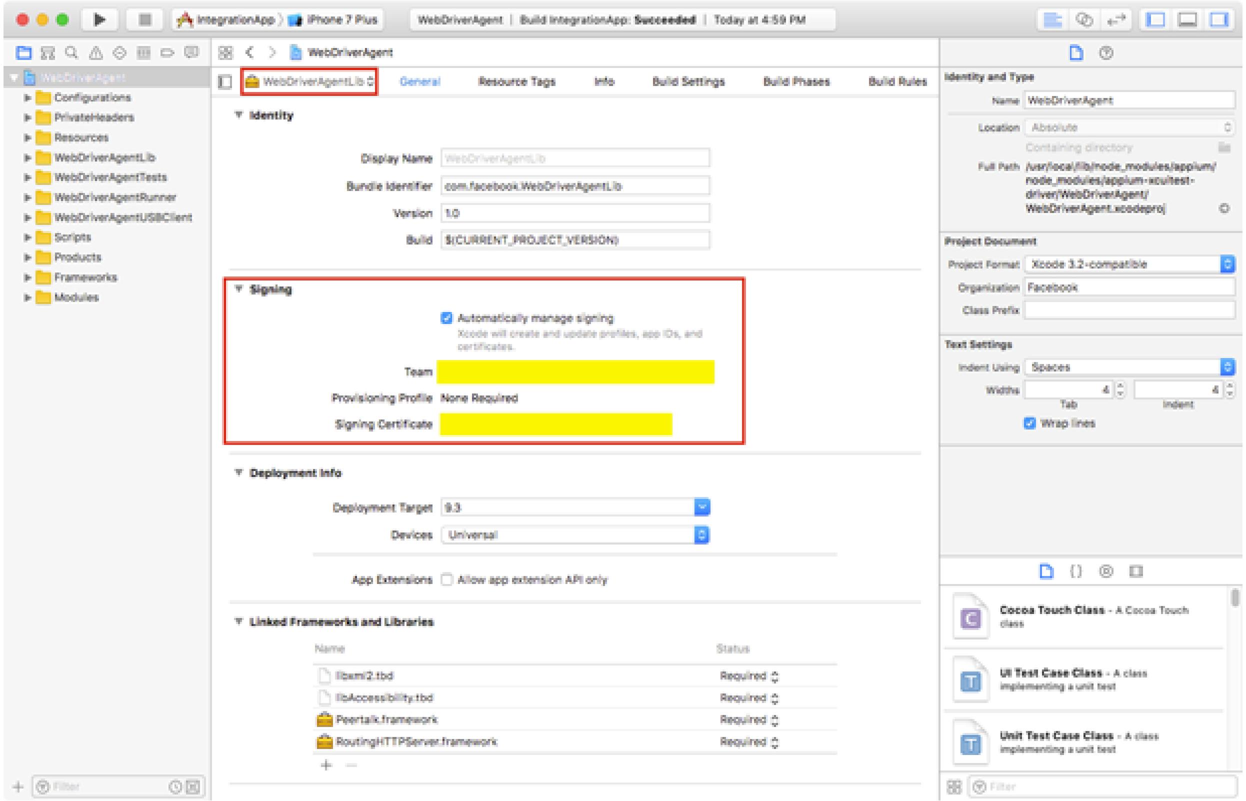
Select the type of account you would like to add:

4. Select **Apple ID** and hit on **Continue**.
5. Provide the Developers ID and Password (Make sure these are **exactly the same** credentials used for **signing the App** under test)
6. Once the developer ID is added to the list under Accounts, close the Preferences window.

Code Signing Process:

- Locate the Appium installation folder. It would usually be present in Applications,
- Navigate to below mentioned path
/Applications/Appium.app/Contents/Resources/app/node_modules/appium/node_modules/appium-xcuitest-driver/WebDriverAgent
To navigate to the above path go to Applications -> Right click on Appium -> click on "**Show Package contents**" and navigate to above path
- Double click and open xcode project: **WebDriverAgent.xcodeproj**.

- Select WebDriverAgentLib, and in the Signing section, check Automatically manage signing and select a team.



Identity

Display Name: WebDriverAgentLib

Bundle Identifier: com.facebook.WebDriverAgentLib

Version: 1.0

Build: $(CURRENT_PROJECT_VERSION)

Signing

☑ Automatically manage signing

Xcode will create and update profiles, app IDs, and certificates.

Team:

Provisioning Profile: None Required

Signing Certificate:

Deployment Info

Deployment Target: 9.3

Devices: Universal

App Extensions: □ Allow app extension API only

Linked Frameworks and Libraries

Name	Status
libxml2.tbd	Required
libAccessibility.tbd	Required
Peertalk.framework	Required
RoutingHTTPServer.framework	Required

Identity and Type

Name: WebDriverAgent

Location: Absolute

Containing directory

Full Path: /usr/local/lib/node_modules/appium/node_modules/appium-xcuitest-driver/WebDriverAgent/WebDriverAgent.xcodeproj

Project Document

Project Format: Xcode 3.2-compatible

Organization: Facebook

Class Prefix:

Text Settings

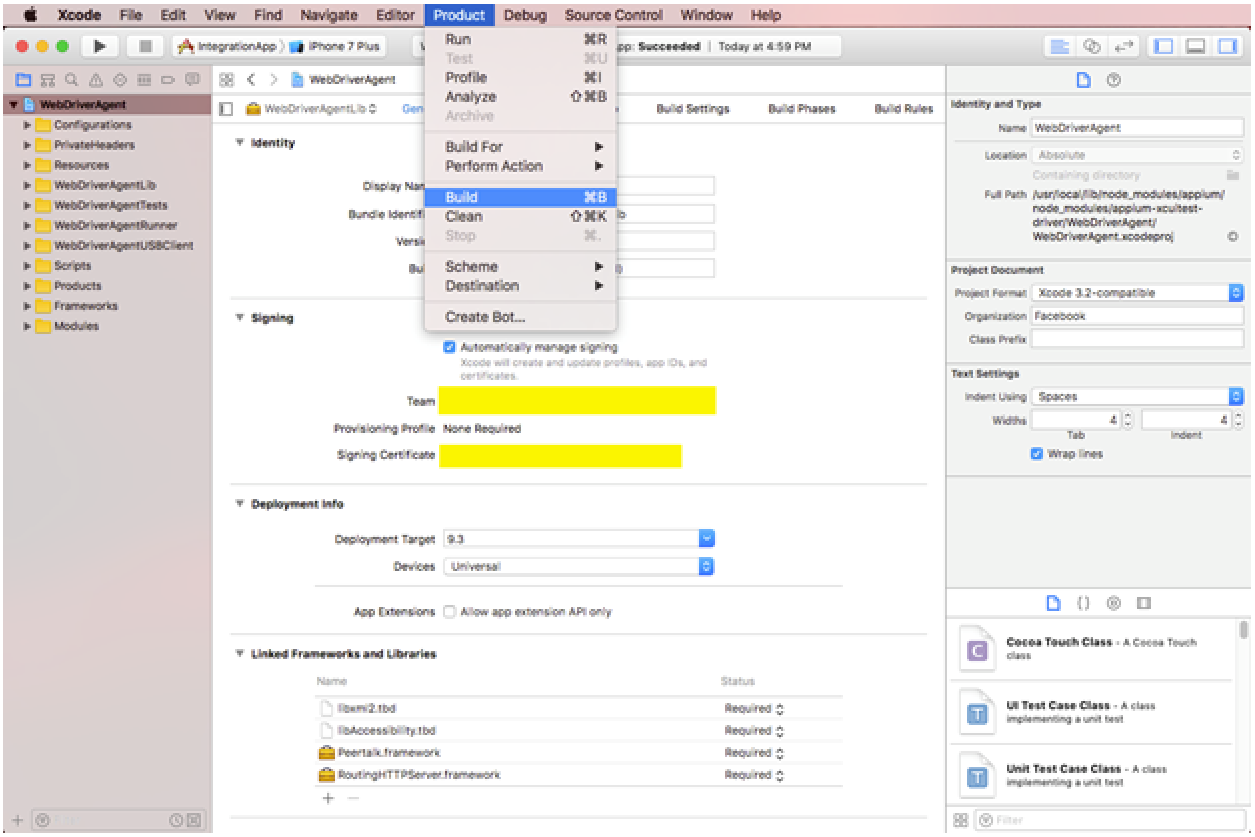
Indent Using: Spaces

Widths: 4 Tab, 4 Indent

☑ Wrap lines

Cocoa Touch Class - A Cocoa Touch class
UI Test Case Class - A class implementing a unit test
Unit Test Case Class - A class implementing a unit test
Filter

- Then on Xcode’s menu bar, select Product > Build



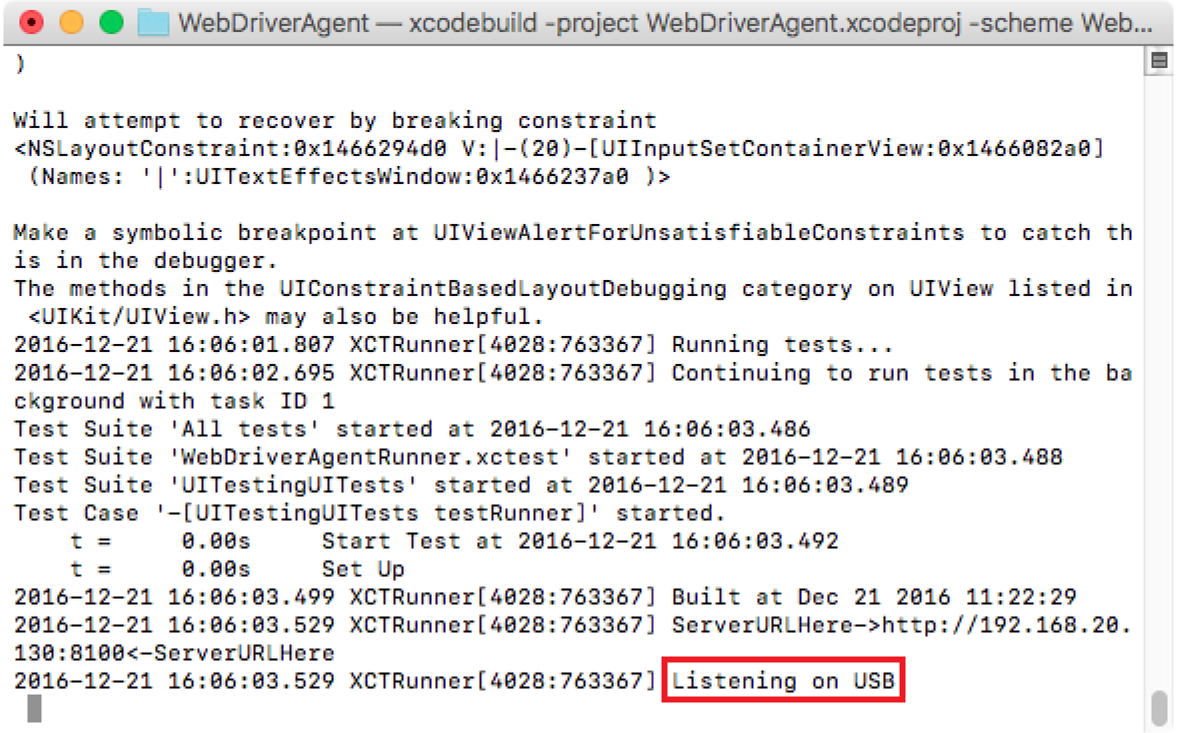
- Repeat the last two steps for WebDriverAgentRunner ALSO.

Build WebDriverAgent to verify whether the steps above work

```
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<udid>' test
```

Note: <udid> is the device UDID of your mobile device.

- You should be able to see *Listening on USB* in the build output as below

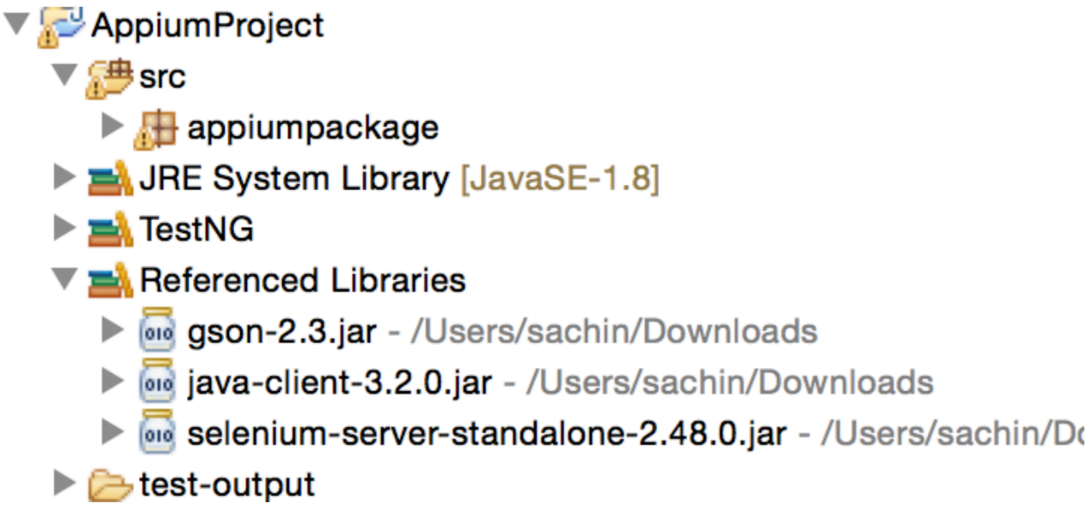

WebDriverAgent — xcodebuild -project WebDriverAgent.xcodeproj -scheme Web...

```
)

Will attempt to recover by breaking constraint
<NSLayoutConstraint:0x1466294d0 V:|-(20)-[UIInputSetContainerView:0x1466082a0]
 (Names: '|':UITextEffectsWindow:0x1466237a0 )>

Make a symbolic breakpoint at UIViewAlertForUnsatisfiableConstraints to catch th
is in the debugger.
The methods in the UIConstraintBasedLayoutDebugging category on UIView listed in
 <UIKit/UIView.h> may also be helpful.
2016-12-21 16:06:01.807 XCTRunner[4028:763367] Running tests...
2016-12-21 16:06:02.695 XCTRunner[4028:763367] Continuing to run tests in the ba
ckground with task ID 1
Test Suite 'All tests' started at 2016-12-21 16:06:03.486
Test Suite 'WebDriverAgentRunner.xctest' started at 2016-12-21 16:06:03.488
Test Suite 'UITestingUITests' started at 2016-12-21 16:06:03.489
Test Case '-[UITestingUITests testRunner]' started.
    t =     0.00s     Start Test at 2016-12-21 16:06:03.492
    t =     0.00s     Set Up
2016-12-21 16:06:03.499 XCTRunner[4028:763367] Built at Dec 21 2016 11:22:29
2016-12-21 16:06:03.529 XCTRunner[4028:763367] ServerURLHere->http://192.168.20.
130:8100<-ServerURLHere
2016-12-21 16:06:03.529 XCTRunner[4028:763367] Listening on USB
```

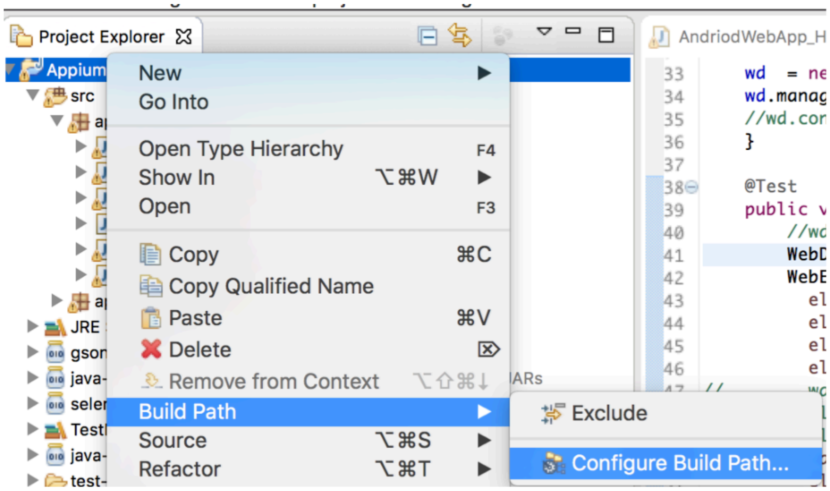
Eclipse setup

- Download (from: http://www.eclipse.org/downloads/) and extract **Eclipse** to a folder and open the Eclipse application.
- Download:
    - **Selenium**(Selenium Standalone server jar)[http://www.seleniumhq.org/download/],
    - **TestNG** jar
    - **gson** and [http://search.maven.org/#artifactdetails%7Ccom.google.code.gson%7Cgson%7C2.4%7Cjar]
    - **Java-client** jars [https://search.maven.org/#search%7Cga%7C1%7Cg%3Aio.appium%20a%3Ajava-client]
- Note: if you are using same MAC machine for Android please install **Android Sdk tools** and below mentioned android plugins

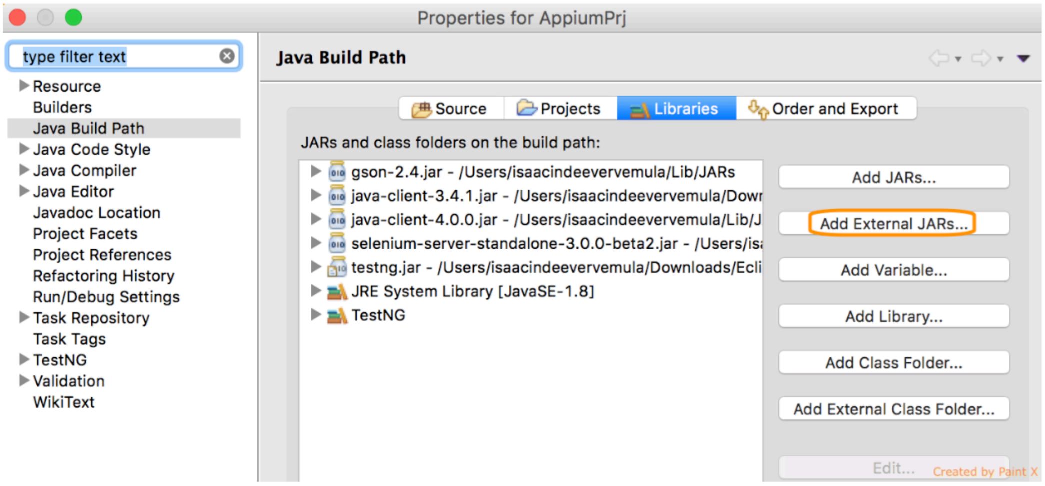
- In Eclipse, create a java project as shown below:


AppiumProject
src
appiumpackage
JRE System Library [JavaSE-1.8]
TestNG
Referenced Libraries
gson-2.3.jar - /Users/sachin/Downloads
java-client-3.2.0.jar - /Users/sachin/Downloads
selenium-server-standalone-2.48.0.jar - /Users/sachin/Dc
test-output

- Configure the downloaded Jar's in Eclipse via **Build Path by** Right-clicking on the project and configure Build Path as shown below

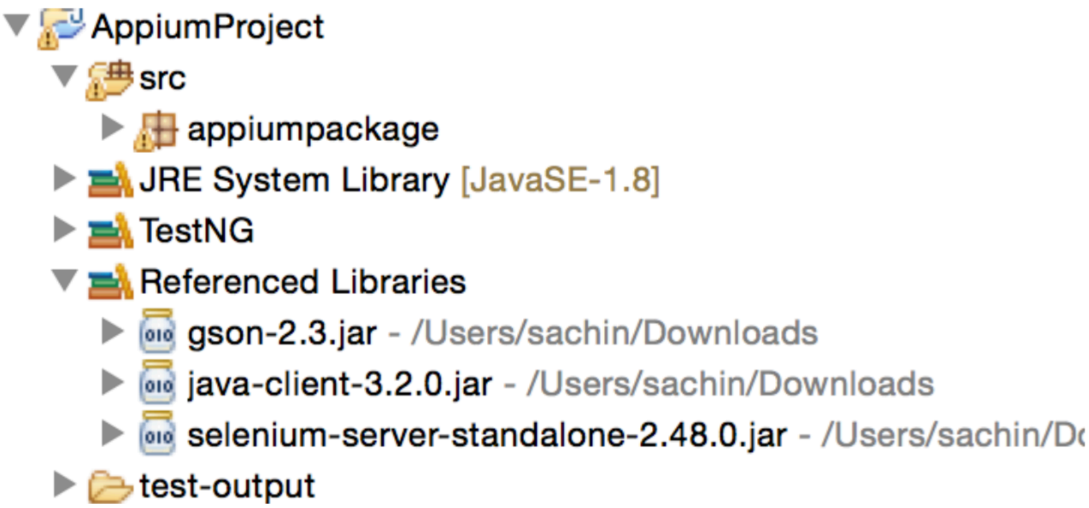


- Adder all the previously downloaded jar files



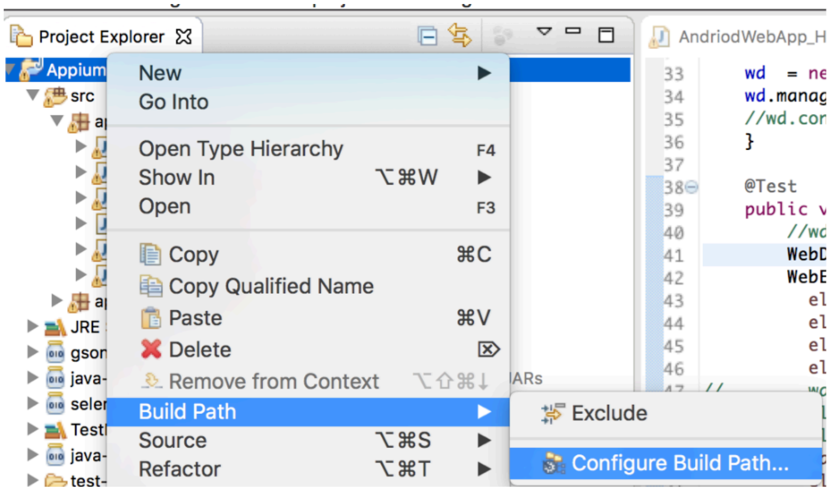
- Install ADT Plugin: can be installed from https://dl-ssl.google.com/android/eclipse(so that eclipse also recognises android-device) (in eclipse >> Help >> Install New Software >> enter the URL: https://dl-ssl.google.com/android/eclipse at work with text box)

- Go to Windows >> preferences >> Android >> SDK Location (should be properly updated)(on Mac Eclipse >> preferences >> Android



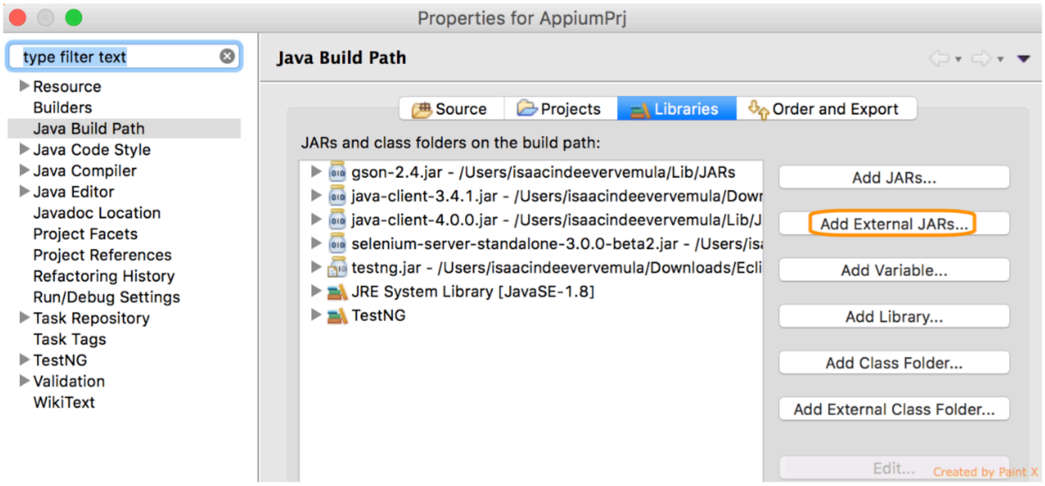
AppiumProject
- src
  - appiumpackage
- JRE System Library [JavaSE-1.8]
- TestNG
- Referenced Libraries
  - gson-2.3.jar - /Users/sachin/Downloads
  - java-client-3.2.0.jar - /Users/sachin/Downloads
  - selenium-server-standalone-2.48.0.jar - /Users/sachin/D
- test-output

- Configure the above downloaded Jar's in Eclipse via **Build Path** by Right-clicking on the project >> Build Path >> configure Build Path as shown below



```
wd = ne
wd.manag
//wd.cor
}

@Test
public v
    //wc
    WebD
    WebE
        el
        el
        el
        el
```

- Adder all the downloaded jar files



Properties for AppiumPrj

Java Build Path

JARs and class folders on the build path:

- gson-2.4.jar - /Users/isaacindeevervemula/Lib/JARs
- java-client-3.4.1.jar - /Users/isaacindeevervemula/Dowr
- java-client-4.0.0.jar - /Users/isaacindeevervemula/Lib/J
- selenium-server-standalone-3.0.0-beta2.jar - /Users/isa
- testng.jar - /Users/isaacindeevervemula/Downloads/Ecli
- JRE System Library [JavaSE-1.8]
- TestNG

- Install ADT Plugin: can be installed from https://dl-ssl.google.com/android/eclipse(so that eclipse also recognises android-device) (in eclipse >> Help >> Install New Software >> enter the URL: https://dl-ssl.google.com/android/eclipse at work with text box)

Go to Windows >> preferences >> Android >> SDK Location (should be properly updated)(on Mac Eclipse >> preferences >> Android

Desired Capabilities

Expand source

```
Setup the Desired capabilities in the above mentioned java class
import java.io.File;
import java.net.MalformedURLException;
import java.net.URL;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import io.appium.java_client.MobileElement;
import io.appium.java_client.ios.IOSDriver;

public class Realdevice {
 public static void main(String[] args) throws MalformedURLException {
        File appDir = new File("/Users/syedzubair/Desktop");
        File app = new File(appDir,"Owners.ipa");
       DesiredCapabilities capabilities = new DesiredCapabilities();
       capabilities.setCapability("deviceName", "iPhone5");
       capabilities.setCapability("platformName", "iOS");
       capabilities.setCapability("platformVersion", "11.0");
       capabilities.setCapability("udid", "fc1650f37d318a2b3c0399db133ff76209c5a6bc");
       capabilities.setCapability("app", app.getAbsolutePath());
       capabilities.setCapability("bundleId", "com.Owners.Owners.com");
       capabilities.setCapability("automationName", "XCUITest");
       WebDriver driver = new IOSDriver<MobileElement>(new
URL("http://127.0.0.1:4723/wd/hub"),
               capabilities);
       }
 }
```

Error's that you may face while running your Automation script:-

Error 1: Unknown udid

You may get this error even if you have entered correct udid as well this is because the corresponding library which talks to protocols of iphone may have not got installed,resolve this error by installing libimobiledevice

through terminal,just type below command in terminal

brew install libimobiledevice --HEAD

Error 2: could not intialize ios deploy

run below command in terminal and install ios display

npm install -g ios-deploy

incase if you face any errors while running above commands please use below mentioned command

npm install -g ios-deploy --unsafe-perm=true

Error 3:- carthage not found

run below command in terminal and install carthage

brew install carthage

Error 4:- error code 65

- First locate where appium is installed it would usually be located in Applications,
- Navigate to below mentioned path
  /Applications/Appium.app/Contents/Resources/app/node_modules/appium/node_modules/appium-xcuitest-driver/WebDriverAgent
  To navigate to the above path go to Applications -> Right click on Appium -> click on "**Show Package contents**" and navigate to above path
- Double click and open xcode project: **WebDriverAgent.xcodeproj**.
- Before you start code signing, ensure that you have added any of the ios developer account in xcode
  To add account follow these steps click on xcode option on top left corner --> click on preferences option --> click on Accounts tab -->click on + icon
  & enter developer's email & password to sign in.
- 2 folders:-
  - WebDriverAgentRunner
  - WebDriverAgentRunner

- Now first select "WebDriverAgentRunner" ,go to Tab "General" under Signing section select Team as "Realhome services & solutions,inc" ,then go to tab "Build settings" find field "Product Bundle Indentifier" by scrolling down
  and edit this field as "com.Owners.Owners.com"
- Repeat the previous steps for remaining 5 folders
- Now click on "Play" button ,once build gets Succeded you are ready for your next test run,if it displays message "Build Failed" then you have not edited above section's in "General" & "Build Settings" tab correctly.
- Now start the Appium server freshly
- And Run your java program i.e. (mobile.java),you will notice that error code 65 is no more appearing and you would able to launch owners app successfully.
  ===============
- Apart from the above instructions, if error code 65 still persist, follow the below set of instructions:
  - Use the `xcodeOrgId` and `xcodeSigningId` desired capabilities:

```
    {

"xcodeOrgId": "<Team ID>",
      "xcodeSigningId": "iPhone Developer"

   }
```

  Note: Team ID is a unique 10-character string generated by Apple that is assigned to your team
  =================
  If you still encounter the error as below follow the below steps:
  ERROR: [MJSONWP] Encountered internal error running command: Error: Unable to launch WebDriverAgent because of xcodebuild failure: "xcodebuild failed with code 65". Make sure you follow the tutorial at https://github.com/appium/appium-xcuitest-driver/blob/master/docs/real-device-config.md. Try to remove the WebDriverAgentRunner application from the device if it is installed and reboot the device.

  To confirm diagnosis of this error open Keychain Access app > Certificates "Category" and individually select each of the above certificates taking note of the status in the header pane. A Green Circle with White Checkmark indicates a healthy certificate, while a Blue Circle with White Plus sign indicates an unhealthy certificate and cause of the error.

  Use the following steps to troubleshoot the issue -

  1) Ensure that the Apple Inc. Root Certificate is installed in your keychain. If not, it can be downloaded here.

  2) Ensure the Apple Worldwide Developer Relations Certification Authority certificate is installed in your keychain. If not, it can be downloaded from the bottom link on the Provisioning Portal website > Certificates tab.

  3) First try and restore Trust settings to the proper value "Use System Defaults". To do that, double click on that certificate to expand the Trust section, and select "Use System Defaults". If after doing that the Blue Circle with White Plus sign does not automatically change to a Green Circle with White Checkmark, the problem is not yet fixed and you'll need to continue to the following step to resolve the issue.

  =================

Error 5:- Error 0xe8008018: The identity used to sign the executable is no longer valid.

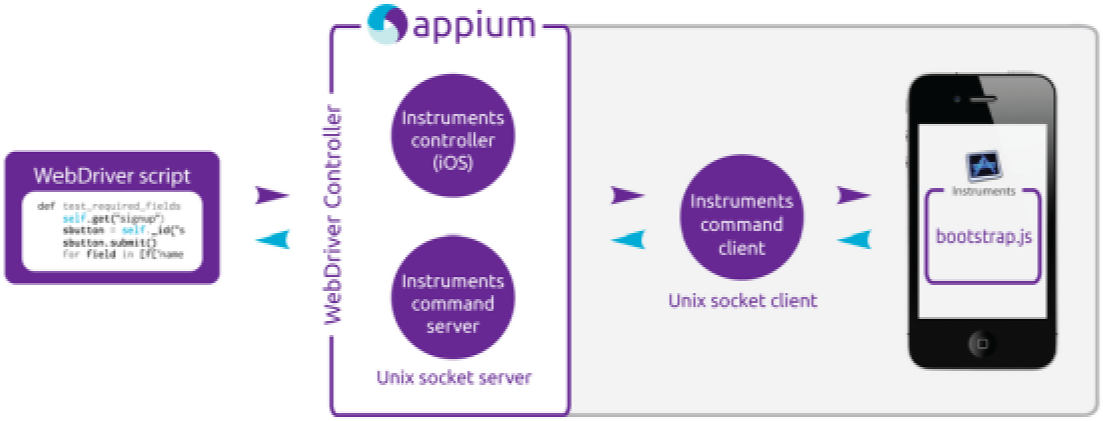
AMDeviceSecureInstallApplication(0, device, url, options, install_callback, 0)

If you face any issue saying that code signing does not match please ensure that you are using the latest version of Owners app

Other Details

Appium: Mobile Automation tool

Appium is an HTTP server which was written in node.js, it’s architecture is as same as Selenium Webdriver Server.
Receiving requests from the client libraries
Executes the commands on the target devices & emulators
Responds with an HTTP response.



```
def test_required_fields
    self.get("signup")
    sbutton = self._id("s
    sbutton.submit()
    for field in [f["name
```

Desired Capabilities:-

Below mentioned are the desired capabilities which are required to set to start an automation session on IPhone Device

```
{
     "platformName": "iOS",
    "platformVersion": "11.0",    (specify any version in which you want to execute your scripts)
    "deviceName": "iPhone 7",  (specify any device name in which you want to execute your scripts)
    "automationName": "XCUITest",
    "app": "/path/to/my.app"
    "udid": "fc1650f37d318a2b3c0399db133ff76209c5a6bc"
}
```

There are many other desired capabilities which are specific to only IOS & Android which would be used based on the requirement

Note:- Refer below mentioned link for Android appium setup

https://wiki.hubzu.com/pages/viewpage.action?pageId=16324454

References

https://appium.readthedocs.io/en/stable/en/appium-setup/real-devices-ios/

For hybrid or web tests, you will also need to follow the ios-webkit-debug-proxy setup instructions