Step 1: Install Java JDK

    Set environment variables

    JAVA\_HOME

    PATH

    java -version

Step 2: Install Homebrew

https://brew.sh/

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

Step 3: Install node js and npm

   brew install node

Step 4: Install Appium : npm install -g appium

       Install Appium Desktop (optional)

Step 5: Get XCode

Xcode is the development and debug environment on Mac.

Provides required tools, files for dev/automation for Mac and iOS apps

Step 6: Install XCode command line tools

    Open terminal and run

    xcode-select --install

Step 7: Create Apple ID

   Goto Xcode - preferences - account

   Add Apple ID

Step 8: Install Carthage     brew install Carthage

In order to launch WebDriverAgent, your macOS will need to have Carthage installed

(Not needed for automation on android)

Step 9: Initialize WebDriverAgent project

On terminal cd to WebDriverAgent folder

WebDriverAgent driver will be found in /path/where/installed/lib/node\_modules/appium/node\_modules/appium-xcuitest-driver/WebDriverAgent

For Appium Desktop

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

For Appium Installed through node

/usr/local/lib/node\_modules/appium/node\_modules/appium-xcuitest-driver/WedDriverAgent

mkdir -p Resources/WebDriverAgent.bundle

./Scripts/bootstrap.sh -d

Note: this command needs Carthage to be installed

Step 10: Open WebDriverAgent.xcodeproj in Xcode

Project name : WebDriverAgent.xcodeproj

Location :

For Appium Desktop

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

For Appium Installed through node

/usr/local/lib/node\_modules/appium/node\_modules/appium-xcuitest-driver/WedDriverAgent

https://github.com/appium/appium/issues/12586

Step 11: Under the project in Xcode

For WebDriverAgentLib and WebDriverAgentRunner targets, go to general tab and select "Automatically manage signing", and then select your Development Team

Xcode may fail to create a provisioning profile for the WebDriverAgentRunner target:

Failed to create a provisioning profile

No profile for ‘…..’ were found

manually change the bundle id for the target by going into the "Build Settings" tab, and changing the "Product Bundle Identifier" from com.Facebook.WebDriverAgentRunner to something that Xcode will accept:

Going back to the "General" tab for the WebDriverAgentRunner target, you should now see that it has created a provisioning profile and all is well:

Clean - Build - Run

Finally, you can verify that everything works. Build the project:

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

udid = device UDID

To get the udid of the real device connected

A9F18306-6B22-442B-99F4-F217A5EE8D38

xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=A9F18306-6B22-442B-99F4-F217A5EE8D38' test

npm install -g ios-deploy

ios-deploy -c

OR

instruments -s devices

Step 12: Create a project and add desired capabilities for automation on iOS

deviceName

platformName

platformVersion

app

Open Eclipse

Create a new maven project or use any existing maven project

Add maven dependencies

 appium

 selenium

  DesiredCapabilities cap = new DesiredCapabilities();

  cap.setCapability("platformName", "iOS");

  cap.setCapability("platformVersion", "11.4");

  cap.setCapability("deviceName", "iPhone 8");

  cap.setCapability(CapabilityType.BROWSER\_NAME, "safari");

  cap.setCapability("app", “location of .app or .ipa file“);

  URL url = new URL("http://127.0.0.1:4723/wd/hub");

  IOSDriver driver = new IOSDriver(url, cap);

Extent Reports

https://www.lambdatest.com/blog/extent-reports-in-selenium/