iOS Testing with Appium.

Estefanía Fdez Muñoz

@Opensouthcode

HELLO!

I am Estefanía Fdez

Head of QA at you can book·me

Previously… Head of QA at SDOS

Una QA en Apuros Blog. SevillaQA. Qa

You can find me on Twitter as @estefafdez

Or Linkedin: https://es.linkedin.com/in/estefafdez

Roadmap:

- Prerequisites.
- Why Appium?
- TestApp
- Install Appium on MacOS.
- Live Example using Maven and Java.

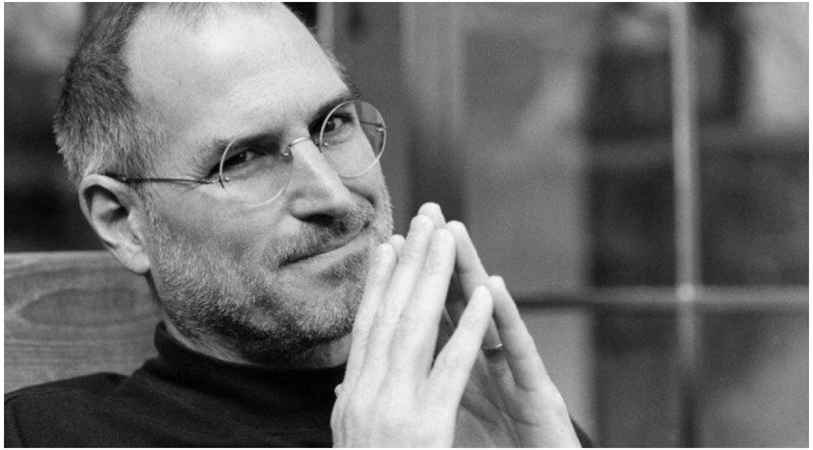
1.

PREREQUISITES

How do we start?

I ❤

To be able to test an iOS App you must have a Mac.

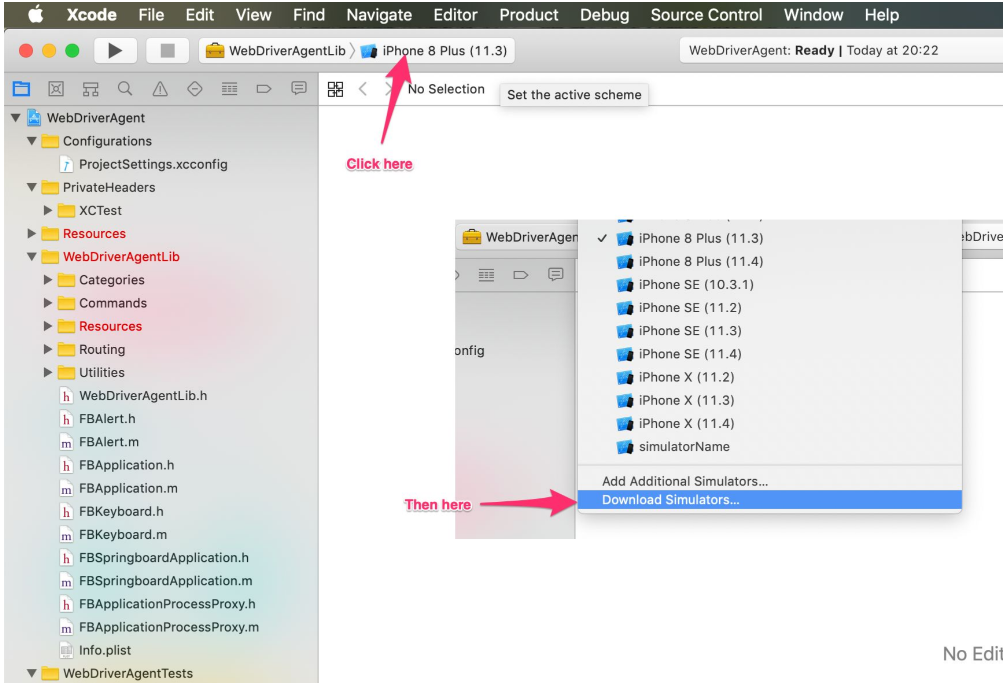


Xcode 9.4 (9F1027a) already installed.

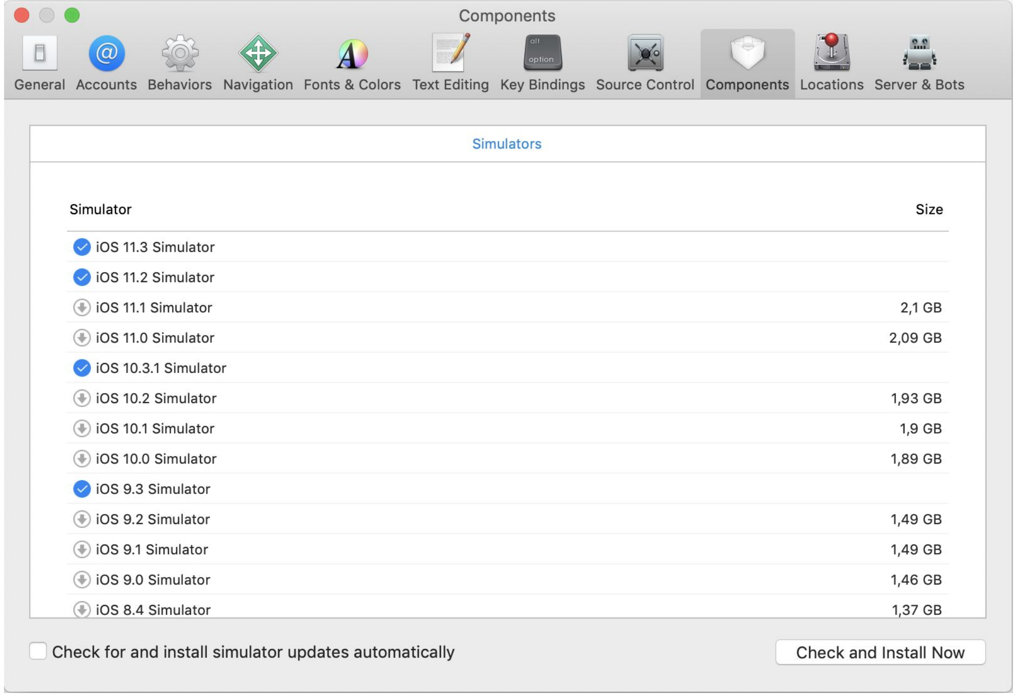
You can download it from: https://developer.apple.com/xcode/

Or without a developer account: https://xcodereleases.com/

Download iOS simulator with iOS 11.3 version.



Download iOS simulator with iOS 11.3 version (II).



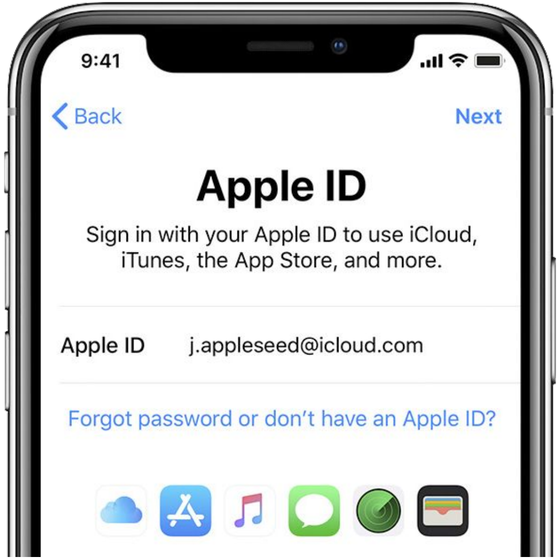
Simulator	Size
✓ iOS 11.3 Simulator	
✓ iOS 11.2 Simulator	
↓ iOS 11.1 Simulator	2,1 GB
↓ iOS 11.0 Simulator	2,09 GB
✓ iOS 10.3.1 Simulator	
↓ iOS 10.2 Simulator	1,93 GB
↓ iOS 10.1 Simulator	1,9 GB
↓ iOS 10.0 Simulator	1,89 GB
✓ iOS 9.3 Simulator	
↓ iOS 9.2 Simulator	1,49 GB
↓ iOS 9.1 Simulator	1,49 GB
↓ iOS 9.0 Simulator	1,46 GB
↓ iOS 8.4 Simulator	1,37 GB

☐ Check for and install simulator updates automatically

Eclipse IDE already installed.

You can download it from: https://www.eclipse.org/downloads/

Have an Apple ID already created.



9:41

Back

Next

Apple ID

Sign in with your Apple ID to use iCloud, iTunes, the App Store, and more.

Apple ID j.appleseed@icloud.com

Forgot password or don't have an Apple ID?

Patience….

.

2.

APPIUM

Why Appium?

Appium is an Open Source tool.

You can download it from: https://github.com/appium

Available for both platform: Android and iOS.

appium

- Available for **hybrid** and **native** apps.
- **10 Programming Languages:** Java, Objective-C, JavaScript con Node.js, PHP, Python, Ruby, C#, Clojure, or Perl.
- Mobile Automation Standard.
- Appium Community: https://discuss.appium.io/

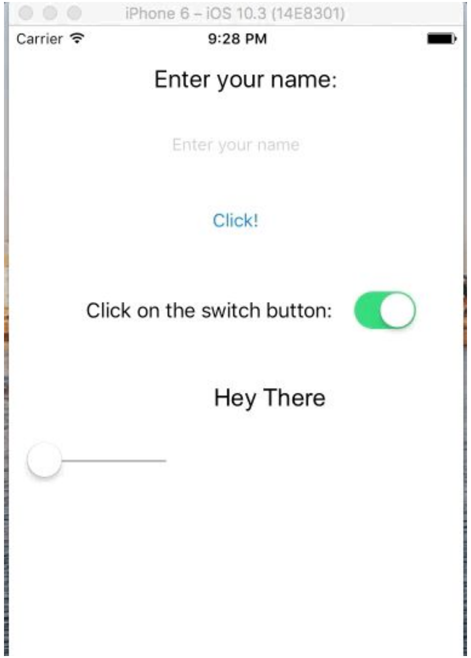
3.

TestApp

Show me some code!

How can I do an iOS App to test?

Hello TestApp!


iPhone 6 – iOS 10.3 (14E8301)

Carrier 9:28 PM

Enter your name:

Enter your name

Click!

Click on the switch button:

Hey There

TestApp

- Single View Application developed in **Swift 3** for testing purposes.
- Download the code: https://github.com/estefafdez/iOSTestApp
- iOS App extension: **.ipa**, **.app**.

4.

INSTALL APPIUM

Ready? Let’s go!

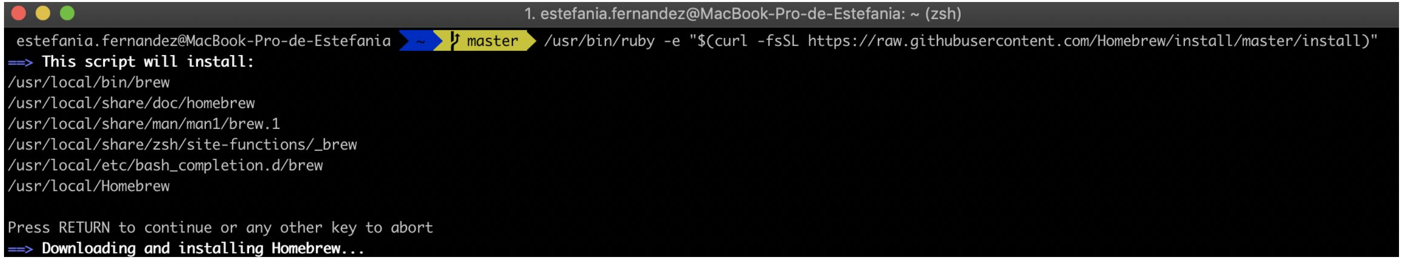
Things to install

- Homebrew
- Maven
- Java
- NodeJS
- Carthage
- Appium Server
- Appium Doctor
- WebDriverAgent
- Libimobiledevice
- ideviceinstaller
- ios-deploy
- Xcpretty
- Appium-ios-driver
- Appium Desktop

INSTALL HOMEBREW

Homebrew installation.

- Go to: https://brew.sh/index_es.html
- Follow the steps!



```
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
==> Downloading and installing Homebrew...
```

Add Homebrew path to the ~/.bash_profile.

Type on your terminal:

```
echo 'export PATH="/usr/local/bin:$PATH"' >> ~/.bash_profile
echo 'export PATH="/usr/local/sbin:$PATH"' >> ~/.bash_profile
source ~/.bash_profile
```

INSTALL MAVEN

Install Maven using Homebrew.

Type on your terminal:

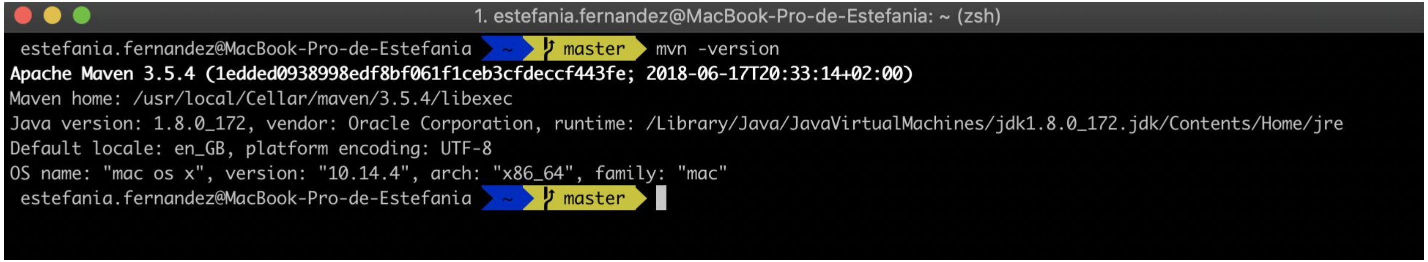
- brew install maven

NOTE: This will install the latest version of maven available.

Install Maven using Homebrew (II).

Check your maven version:

» mvn -version



```
 estefania.fernandez@MacBook-Pro-de-Estefania  ~   master  mvn -version
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T20:33:14+02:00)
Maven home: /usr/local/Cellar/maven/3.5.4/libexec
Java version: 1.8.0_172, vendor: Oracle Corporation, runtime: /Library/Java/JavaVirtualMachines/jdk1.8.0_172.jdk/Contents/Home/jre
Default locale: en_GB, platform encoding: UTF-8
OS name: "mac os x", version: "10.14.4", arch: "x86_64", family: "mac"
 estefania.fernandez@MacBook-Pro-de-Estefania  ~   master 
```

INSTALL JAVA

Install Java using Homebrew.

Type on your terminal:

```
brew cask install homebrew/cask-versions/java8
```

NOTE: We suggest install version: 8.

Add Java path to the ~/.bash_profile.

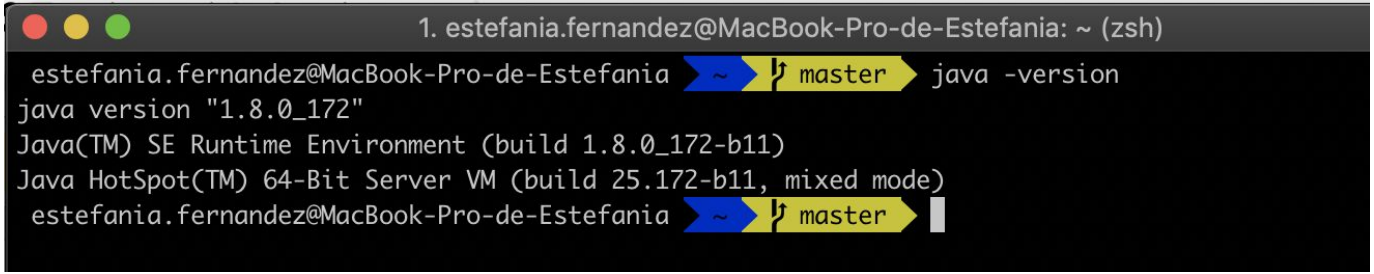
Type on your terminal:

```
echo 'export JAVA_HOME="$(/usr/libexec/java_home)"' >> ~/.bash_profile
echo 'export PATH="$JAVA_HOME/bin:$PATH"' >> ~/.bash_profile
source ~/.bash_profile
```

Install Java using Homebrew (II).

Check your java version:

» java -version



1. estefania.fernandez@MacBook-Pro-de-Estefania: ~ (zsh)

```
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  java -version
java version "1.8.0_172"
Java(TM) SE Runtime Environment (build 1.8.0_172-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.172-b11, mixed mode)
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master 
```

INSTALL NODE

Install Node using Homebrew.

Type on your terminal:

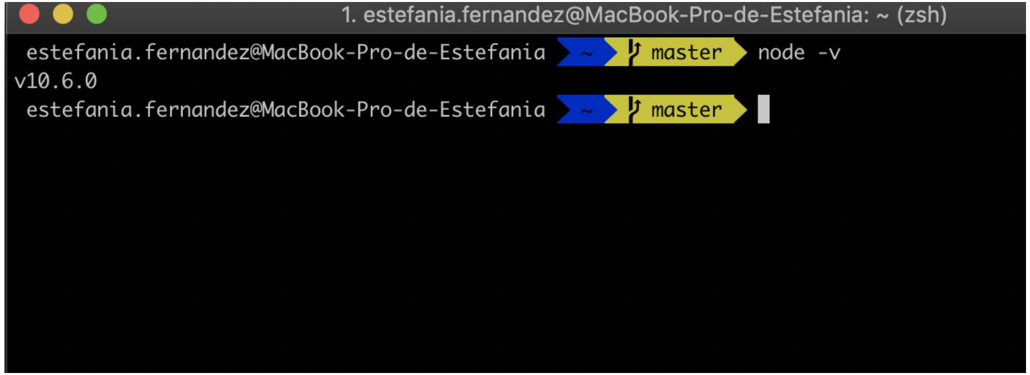
» brew install node

NOTE: We suggest install version: v10.6.0.

Install Node using Homebrew (II).

Check your node version:

- node -v



```
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  node -v
v10.6.0
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  
```

INSTALL CARTHAGE

Install Carthage using Homebrew.

Type on your terminal:

» brew install carthage

INSTALL APPIUM SERVER

appium

Mobile Apps Automation Tool

Install Appium Server using Homebrew.

Type on your terminal:

```
npm install -g appium@1.12.1 --unsafe-perm=true --allow-root
```

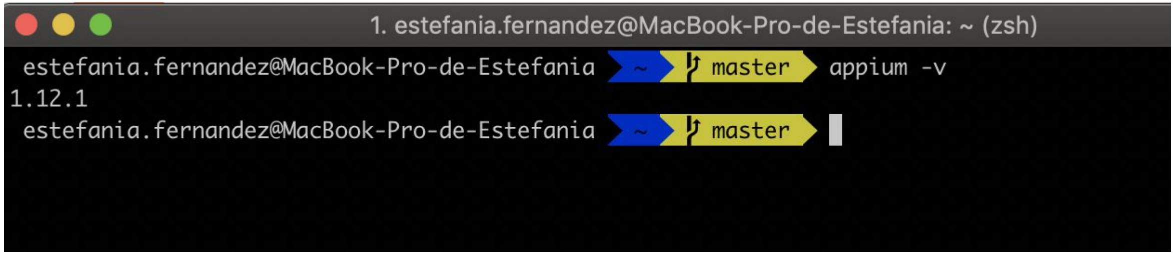
NOTES:

- We suggest install version: 1.12.1.
- The --unsafe-perm=true and --allow-root are not mandatory but they are necessary sometimes.

Install Appium Server using Homebrew (II).

Check your Appium version:

» appium -v

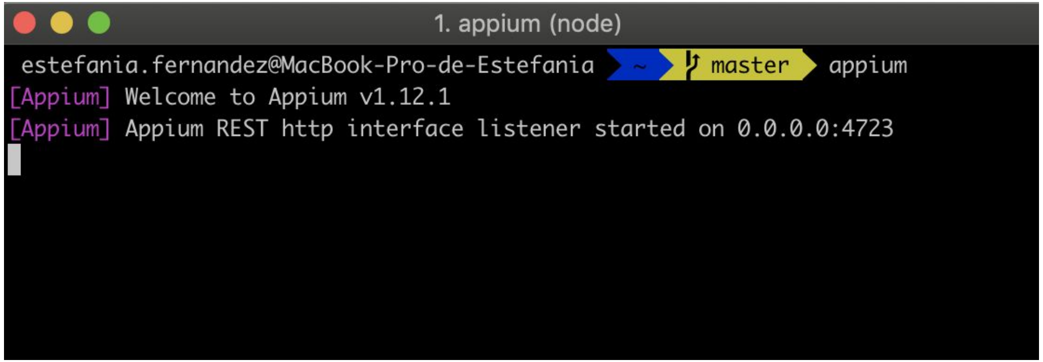


```
estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  appium -v
1.12.1
estefania.fernandez@MacBook-Pro-de-Estefania  ~  master 
```

Execute Appium Server by command line.

Type on your terminal:

» appium &

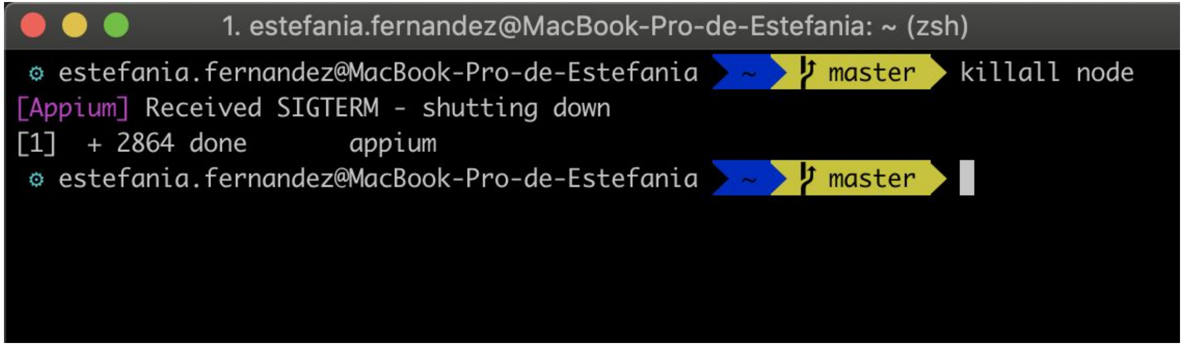


```
estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  appium
[Appium] Welcome to Appium v1.12.1
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

Stop Appium Server by command line.

Type on your terminal:

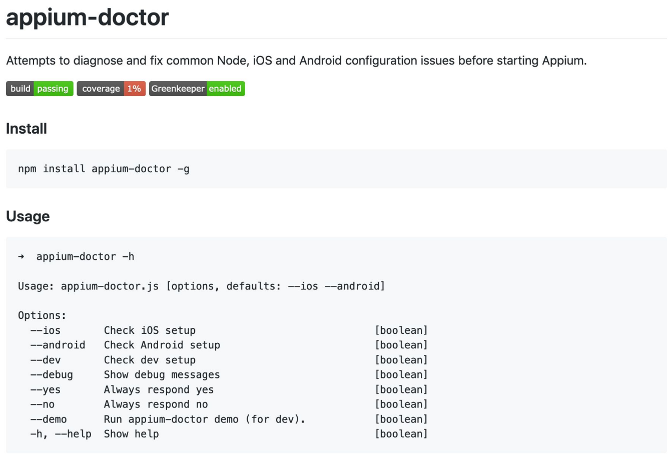
» killall node



1. estefania.fernandez@MacBook-Pro-de-Estefania: ~ (zsh)

```
⚙ estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  killall node
[Appium] Received SIGTERM - shutting down
[1]  + 2864 done       appium
⚙ estefania.fernandez@MacBook-Pro-de-Estefania  ~  master
```

INSTALL APPIUM DOCTOR



appium-doctor

Attempts to diagnose and fix common Node, iOS and Android configuration issues before starting Appium.

Install

```
npm install appium-doctor -g
```

Usage

```
➜ appium-doctor -h

Usage: appium-doctor.js [options, defaults: --ios --android]

Options:
  --ios        Check iOS setup                                [boolean]
  --android    Check Android setup                            [boolean]
  --dev        Check dev setup                                [boolean]
  --debug      Show debug messages                            [boolean]
  --yes        Always respond yes                             [boolean]
  --no         Always respond no                              [boolean]
  --demo       Run appium-doctor demo (for dev).              [boolean]
  -h, --help   Show help                                      [boolean]
```

Install Appium Doctor using Homebrew.

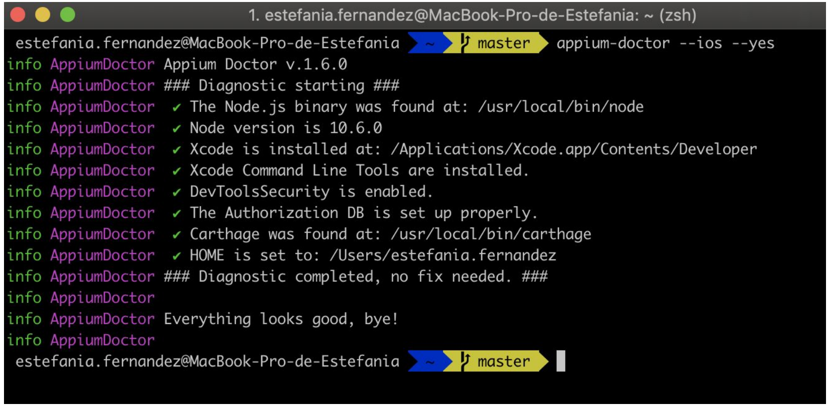
Type on your terminal:

» npm install -g appium-doctor

Check iOS installation is ok with Appium Doctor

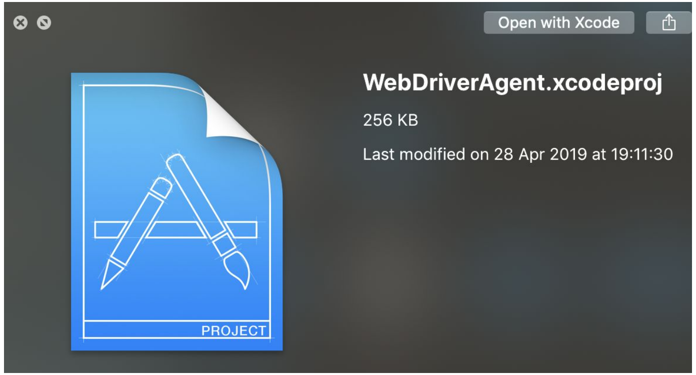
To check if everything you need to start with Appium and iOS is correct, type on your terminal:

» appium-doctor --ios --yes



```
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master  appium-doctor --ios --yes
info AppiumDoctor Appium Doctor v.1.6.0
info AppiumDoctor ### Diagnostic starting ###
info AppiumDoctor  ✔ The Node.js binary was found at: /usr/local/bin/node
info AppiumDoctor  ✔ Node version is 10.6.0
info AppiumDoctor  ✔ Xcode is installed at: /Applications/Xcode.app/Contents/Developer
info AppiumDoctor  ✔ Xcode Command Line Tools are installed.
info AppiumDoctor  ✔ DevToolsSecurity is enabled.
info AppiumDoctor  ✔ The Authorization DB is set up properly.
info AppiumDoctor  ✔ Carthage was found at: /usr/local/bin/carthage
info AppiumDoctor  ✔ HOME is set to: /Users/estefania.fernandez
info AppiumDoctor ### Diagnostic completed, no fix needed. ###
info AppiumDoctor
info AppiumDoctor Everything looks good, bye!
info AppiumDoctor
 estefania.fernandez@MacBook-Pro-de-Estefania  ~  master 
```

CONFIGURE WEBDRIVER AGENT



WebDriverAgent.xcodeproj

256 KB

Last modified on 28 Apr 2019 at 19:11:30

Configure WebDriverAgent

Go to Appium-XCUITest and WebDriverAgent folder:

- cd "/usr/local/lib/node_modules/appium/node_modules/appium-xcuitest-driver/WebDriverAgent/"

Create the following folder:

- mkdir -p Resources/WebDriverAgent.bundle

Configure WebDriverAgent (II)

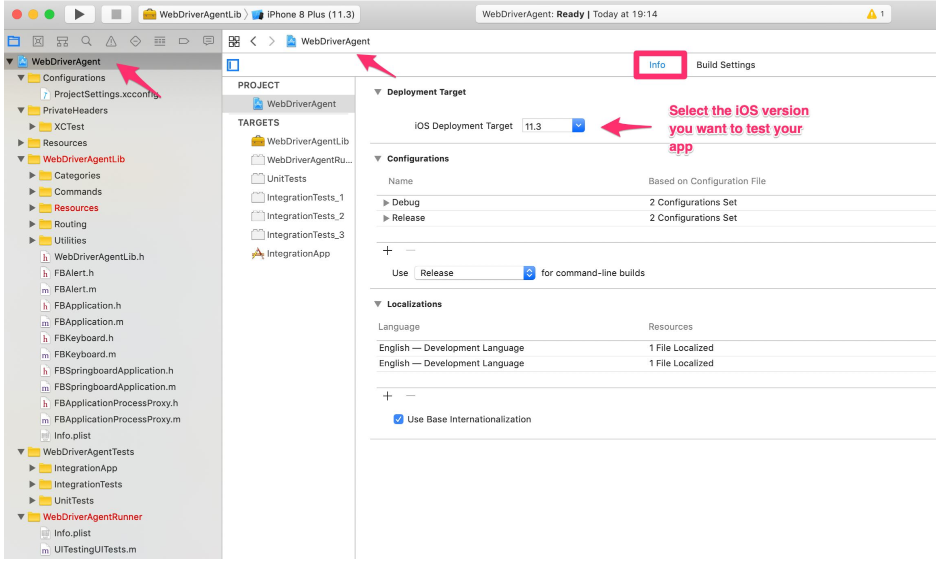
Execute the bootstrap script:

```
./Scripts/bootstrap.sh
```

Now let’s go to the Xcode!

Configure WebDriverAgent on Xcode.

» Now, select the WebDriverAgent project and click on the Info tab to change on the Deployment Target , the iOS Deployment Target. Select the iOS version you want to test your application.



Deployment Target

iOS Deployment Target 11.3

Select the iOS version you want to test your app

Configurations

Name	Based on Configuration File
Debug	2 Configurations Set
Release	2 Configurations Set

Use Release for command-line builds

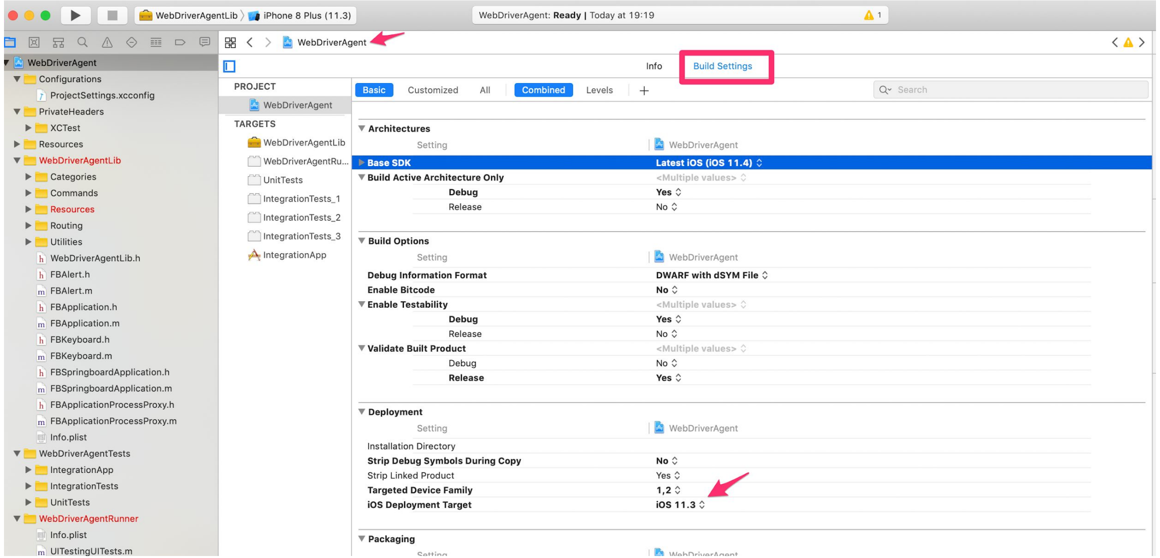
Localizations

Language	Resources
English — Development Language	1 File Localized
English — Development Language	1 File Localized

☑ Use Base Internationalization

Configure WebDriverAgent on Xcode.

» On the Build Settings tab, on the Deployment settings, change the iOS Deployment Target as well, to match the same version as the Info tab.



Architectures

Setting	WebDriverAgent
Base SDK	**Latest iOS (iOS 11.4)**
Build Active Architecture Only	<Multiple values>
Debug	**Yes**
Release	No

Build Options

Setting	WebDriverAgent
Debug Information Format	**DWARF with dSYM File**
Enable Bitcode	**No**
Enable Testability	<Multiple values>
Debug	**Yes**
Release	No
Validate Built Product	<Multiple values>
Debug	No
Release	**Yes**

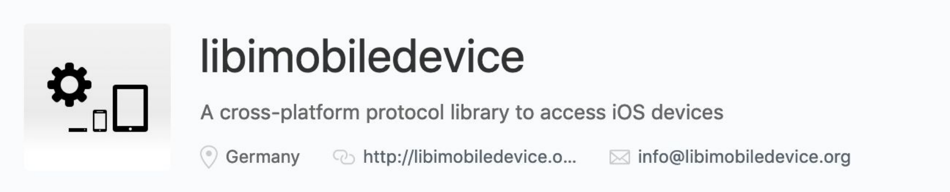
Deployment

Setting	WebDriverAgent
Installation Directory	
Strip Debug Symbols During Copy	**No**
Strip Linked Product	Yes
Targeted Device Family	**1,2**
iOS Deployment Target	**iOS 11.3**

Packaging

Setting	WebDriverAgent

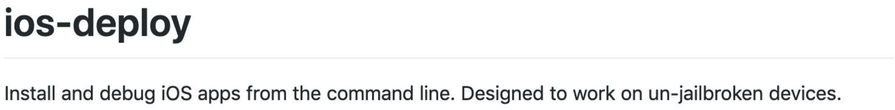
INSTALL LAST LIBRARIES



libimobiledevice

A cross-platform protocol library to access iOS devices

Germany http://libimobiledevice.o... info@libimobiledevice.org



ios-deploy

Install and debug iOS apps from the command line. Designed to work on un-jailbroken devices.

Install more libraries using Homebrew.

Type on your terminal:

» brew install libimobiledevice --HEAD

» npm install -g ios-deploy --unsafe-perm=true --allow-root

» sudo gem install xcpretty

Install more libraries using Homebrew (II).

Type on your terminal:

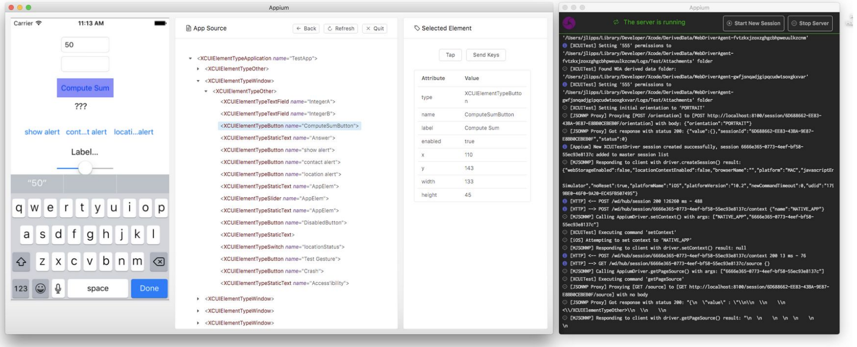
```
brew install ideviceinstaller
```

```
npm install appium-ios-driver
```

INSTALL APPIUM DESKTOP

Appium Desktop

Azure Pipelines succeeded | localized 44% | Greenkeeper disabled



```
▾ <XCUIElementTypeApplication name="TestApp">
  ▸ <XCUIElementTypeOther>
  ▾ <XCUIElementTypeWindow>
    ▾ <XCUIElementTypeOther>
        <XCUIElementTypeTextField name="IntegerA">
        <XCUIElementTypeTextField name="IntegerB">
        <XCUIElementTypeButton name="ComputeSumButton">
        <XCUIElementTypeStaticText name="Answer">
        <XCUIElementTypeButton name="show alert">
        <XCUIElementTypeButton name="contact alert">
        <XCUIElementTypeButton name="location alert">
        <XCUIElementTypeStaticText name="AppElem">
        <XCUIElementTypeSlider name="AppElem">
        <XCUIElementTypeStaticText name="AppElem">
        <XCUIElementTypeButton name="DisabledButton">
        <XCUIElementTypeStaticText>
        <XCUIElementTypeSwitch name="locationStatus">
        <XCUIElementTypeButton name="Test Gesture">
        <XCUIElementTypeButton name="Crash">
        <XCUIElementTypeStaticText name="Access'ibility">
  ▸ <XCUIElementTypeWindow>
  ▸ <XCUIElementTypeWindow>
  ▸ <XCUIElementTypeWindow>
```

Attribute	Value
type	XCUIElementTypeButto n
name	ComputeSumButton
label	Compute Sum
enabled	true
x	110
y	143
width	133
height	45

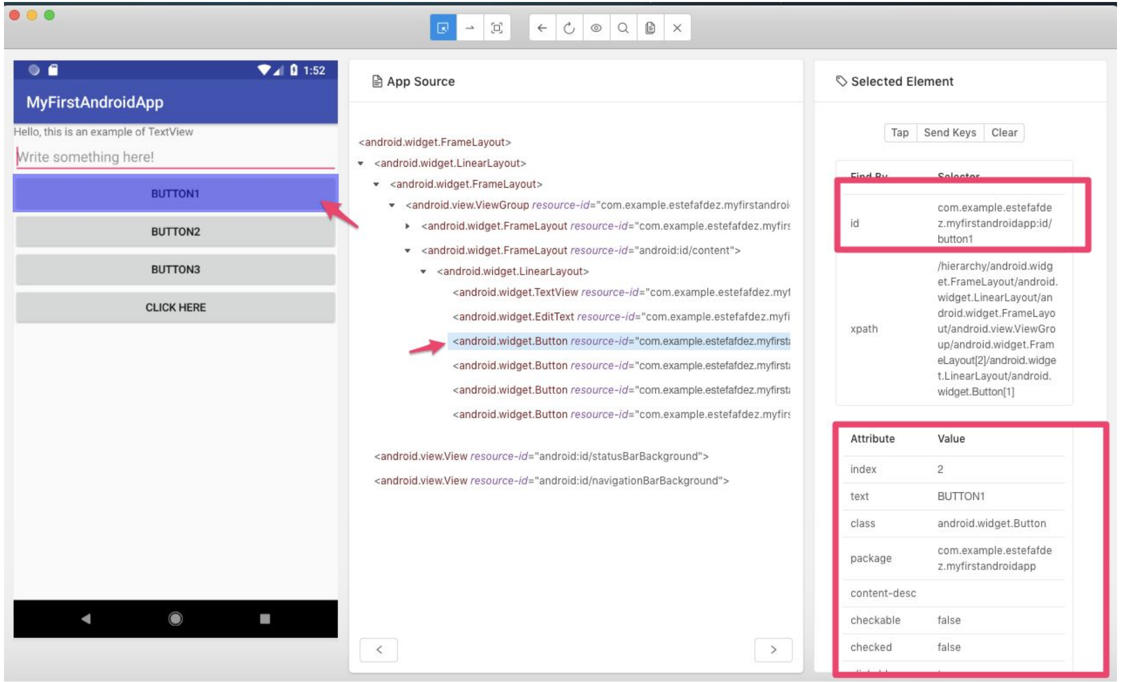
```
'/Users/jlipps/Library/Developer/Xcode/DerivedData/WebDriverAgent-fvtzkxjzoxzghgcbhpweuulkzcnm'
[XCUITest] Setting '555' permissions to
'/Users/jlipps/Library/Developer/Xcode/DerivedData/WebDriverAgent-
fvtzkxjzoxzghgcbhpweuulkzcnm/Logs/Test/Attachments' folder
[XCUITest] Found WDA derived data folder:
'/Users/jlipps/Library/Developer/Xcode/DerivedData/WebDriverAgent-gwfjsnqadjgipqcudwtsoxgkxvar'
[XCUITest] Setting '555' permissions to
'/Users/jlipps/Library/Developer/Xcode/DerivedData/WebDriverAgent-
gwfjsnqadjgipqcudwtsoxgkxvar/Logs/Test/Attachments' folder
[XCUITest] Setting initial orientation to 'PORTRAIT'
[JSONWP Proxy] Proxying [POST /orientation] to [POST http://localhost:8100/session/6D688662-EE83-
43BA-9E87-E8BB0CEBEB0F/orientation] with body: {"orientation":"PORTRAIT"}
[JSONWP Proxy] Got response with status 200: {"value":{},"sessionId":"6D688662-EE83-43BA-9E87-
E8BB0CEBEB0F","status":0}
[Appium] New XCUITestDriver session created successfully, session 6666e365-0773-4eef-bf58-
55ec93e8137c added to master session list
[MJSONWP] Responding to client with driver.createSession() result:
{"webStorageEnabled":false,"locationContextEnabled":false,"browserName":"","platform":"MAC","javascriptEr

Simulator","noReset":true,"platformName":"iOS","platformVersion":"10.2","newCommandTimeout":0,"udid":"179
9BE0-46F0-9A20-EC45FB507495"}
[HTTP] <-- POST /wd/hub/session 200 126260 ms - 488
[HTTP] --> POST /wd/hub/session/6666e365-0773-4eef-bf58-55ec93e8137c/context {"name":"NATIVE_APP"}
[MJSONWP] Calling AppiumDriver.setContext() with args: ["NATIVE_APP","6666e365-0773-4eef-bf58-
55ec93e8137c"]
[XCUITest] Executing command 'setContext'
[iOS] Attempting to set context to 'NATIVE_APP'
[MJSONWP] Responding to client with driver.setContext() result: null
[HTTP] <-- POST /wd/hub/session/6666e365-0773-4eef-bf58-55ec93e8137c/context 200 13 ms - 76
[HTTP] --> GET /wd/hub/session/6666e365-0773-4eef-bf58-55ec93e8137c/source {}
[MJSONWP] Calling AppiumDriver.getPageSource() with args: ["6666e365-0773-4eef-bf58-55ec93e8137c"]
[XCUITest] Executing command 'getPageSource'
[JSONWP Proxy] Proxying [GET /source] to [GET http://localhost:8100/session/6D688662-EE83-43BA-9E87-
E8BB0CEBEB0F/source] with no body
[JSONWP Proxy] Got response with status 200: "{\n  \"value\" : \"\\n\\n  \\n    \\n
<\\/XCUIElementTypeOther>\\n  \\n    \\n
[MJSONWP] Responding to client with driver.getPageSource() result: "\n  \n    \n  \n  \n    \n
\n
```

Install Appium Desktop.

- Appium Desktop is an app for Mac, Windows, and Linux which gives you the power of the Appium automation server in a beautiful and flexible UI. You can inspect element on an App.

- You can download it from: https://github.com/appium/appium-desktop

Install Appium Desktop (II).

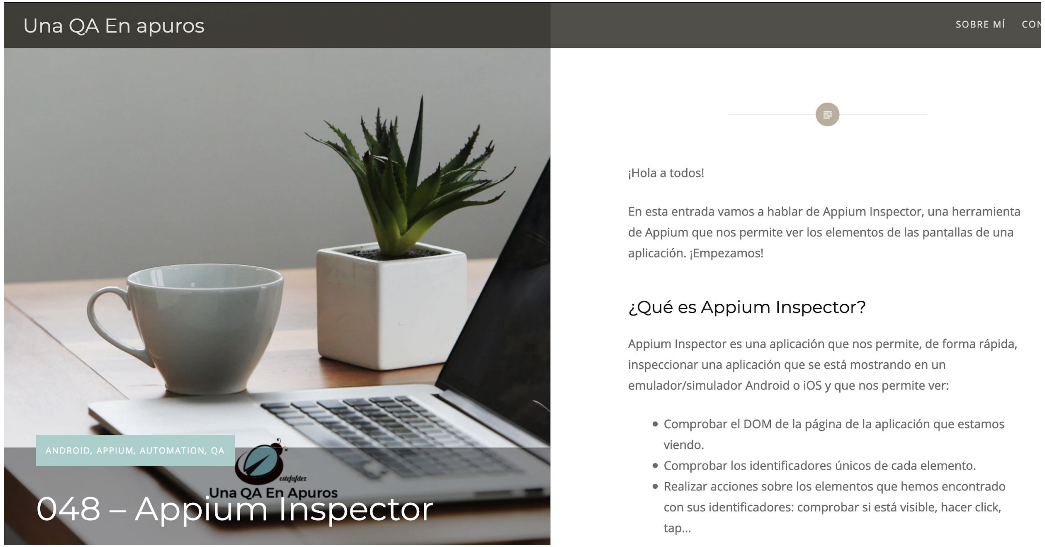


```
<android.widget.FrameLayout>
 ▾ <android.widget.LinearLayout>
   ▾ <android.widget.FrameLayout>
     ▾ <android.view.ViewGroup resource-id="com.example.estefafdez.myfirstandroi
       ▸ <android.widget.FrameLayout resource-id="com.example.estefafdez.myfirs
       ▾ <android.widget.FrameLayout resource-id="android:id/content">
         ▾ <android.widget.LinearLayout>
             <android.widget.TextView resource-id="com.example.estefafdez.myf
             <android.widget.EditText resource-id="com.example.estefafdez.myfi
             <android.widget.Button resource-id="com.example.estefafdez.myfirst
             <android.widget.Button resource-id="com.example.estefafdez.myfirst
             <android.widget.Button resource-id="com.example.estefafdez.myfirst
             <android.widget.Button resource-id="com.example.estefafdez.myfirs

   <android.view.View resource-id="android:id/statusBarBackground">
   <android.view.View resource-id="android:id/navigationBarBackground">
```

Find By	Selector
id	com.example.estefafdez.myfirstandroidapp:id/button1
xpath	/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[2]/android.widget.LinearLayout/android.widget.Button[1]

Attribute	Value
index	2
text	BUTTON1
class	android.widget.Button
package	com.example.estefafdez.myfirstandroidapp
content-desc	
checkable	false
checked	false

Tutorial about Appium Desktop available here:



Una QA En apuros

048 – Appium Inspector

¡Hola a todos!

En esta entrada vamos a hablar de Appium Inspector, una herramienta de Appium que nos permite ver los elementos de las pantallas de una aplicación. ¡Empezamos!

¿Qué es Appium Inspector?

Appium Inspector es una aplicación que nos permite, de forma rápida, inspeccionar una aplicación que se está mostrando en un emulador/simulador Android o iOS y que nos permite ver:

- Comprobar el DOM de la página de la aplicación que estamos viendo.
- Comprobar los identificadores únicos de cada elemento.
- Realizar acciones sobre los elementos que hemos encontrado con sus identificadores: comprobar si está visible, hacer click, tap...

» https://unaqaenapuros.wordpress.com/2019/02/27/048-appium-inspector/

5.

LIVE EXAMPLE USING MAVEN AND JAVA.

Ready? Let’s go!

Download the repository:

- https://github.com/estefafdez/appium-native-java-example

LIVE EXAMPLE....
START NOW!

6.

SUCCESS! ✓

Enjoy testing your iOS App!

THANKS!

Any questions?

You can find me at @estefafdez & estefafdez@gmail.com

- http://estefafdez.com
- https://unaqaenapuros.wordpress.com/