前言http://www.cnblogs.com/yoyoketang

mac上搭建appium+python的环境还是有点复杂的，需要准备的软件

1.nodejs

2.npm

3.cnpm

4.appium

5.pip

6.Appium-Python-Client

一、nodejs安装

1.nodejs下载地址：<https://nodejs.org/>

2.下载完成后傻瓜式下一步安装

3.打开终端，查看版本号是否安装成功，输入：node -v

二、npm安装

1.打开终端，输入：sudo bash

2.继续输入：sudo curl -L https://npmjs.org/install.sh | sh

3.看到上图所示，就是安装完成了。检查npm是否安装成功，先关掉终端，再重新打开，输入：npm -v

三、cnpm安装

1.用npm安装appium时候，中途会报错(由于某种原因,直接用npm下载安装会有好多网络问题，安装淘宝的cnpm要比npm好用),

所以这里用cnpm代替了npm

2.打开终端，输入sudo npm install -g cnpm --registry=https://registry.npm.taobao.org



四、appium安装

1.appium的安装有两种：一种是有GUI图形界面的，下载dmg包，手动安装；另外一种是通过cnpm指令安装的，无图形界面

方式一：下载dmg包手动安装（新手推荐）

官方下载地址：http://appium.io/

百度云下载地址：链接: https://pan.baidu.com/s/1pKAj1qj 密码: q3qu

下载后傻瓜式下一步安装就行了

方法二：cnpm安装（无图形界面）

2.这里下载1.6.3的版本，@后面可以指定版本号。终端输入：cnpm install -g appium@1.6.3

（要是报错，就用sudo去安装：sudo cnpm install -g appium@1.6.3）

3.安装完成后，想启动appium，就在终端输入:appium

看到如下界面：Welcome to Appium v1.6.3说明安装成功了

----------------------以上是安装appium相关包，接下来是准备python相关的包---------------

五、pip安装

1.pip是安装python第三方包的最主要的一个小工具，打开终端输入：sudo easy\_install pip

六、Appium-Python-Client

1.安装appium的python包，打开终端输入：sudo pip install Appium-Python-Client

2.中途要是弹出Password的提示，就输入当前用户的密码就行

还需要安装的软件：

1.jdk

2.xcode

3.pycharm

appiumQQ群：512200893

在mac上搭建appium踩了不少坑，先是版本低了，启动后无限重启模拟器。后来全部升级最新版本，就稳稳的了。

环境准备：

1.OS版本号10.12

2.xcode版本号8.3.2

3.appium版本号1.6.4（appium-desktop1.1.0）

4.ios模拟器版本号10.3

一、OS10.12

1.这里Mac上的OS系统一定要升级到10.12，低于10.12是无法安装8.3.2的xcode版本的



2.下载地址：https://www.apple.com/macos/sierra/

二、xcode

1.Xcode版本号下载8.3以上，小编这里是8.3.2的版本：



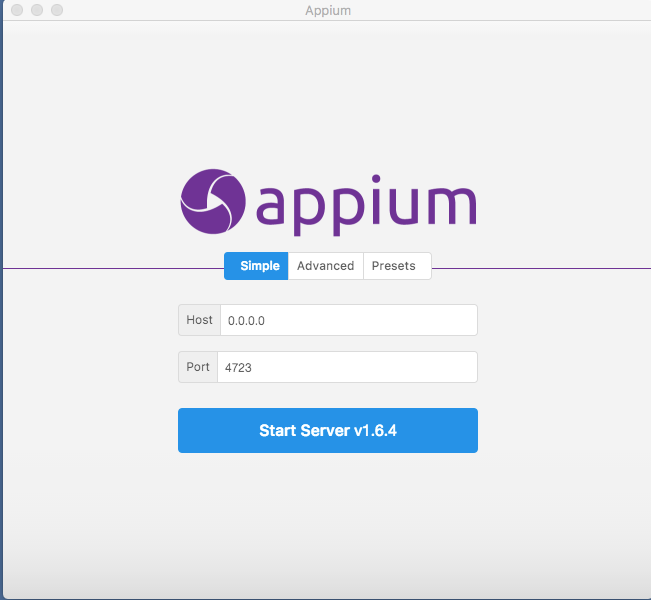
2.xocde在sppstore就能搜索下载，但是太慢了，这里直接去苹果开发者官网下载：https://developer.apple.com/support/xcode/

三、appium

1.appium的依赖环境，参考这一篇：[appium+python在Mac上环境搭建](http://www.cnblogs.com/yoyoketang/p/7368246.html)

2.这里主要是下载一个客户端的包，下载地址：http://pan.baidu.com/s/1pKEoHBH，如下图，下载那个dmg文件就能傻瓜式安装了。

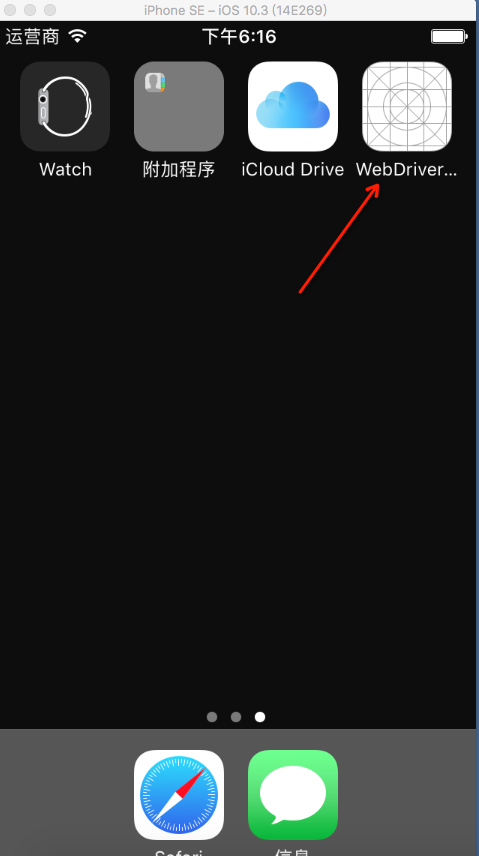
3.安装好后，启动界面如下，不用做任何的修改，直接点Start Server 1.6.4



四、ios模拟器Simulator

1.启动xcode，新建一个项目，随便设置下，在屏幕左上角找到Simulator,参考教程：https://jingyan.baidu.com/article/7c6fb4280f3c4580642c90de.html

2.我这里选的是iOS10.3的版本，iPhone SE



五、pycharm安装

1.先去pycharm官网下载一个dmg包，地址：https://www.jetbrains.com/pycharm/download/

2.选中mac版的Professional专业版就行，下载完之后傻瓜式安装

3.打开软件，在License server address中填入http://xidea.online，然后点击Activate即可顺利破解！

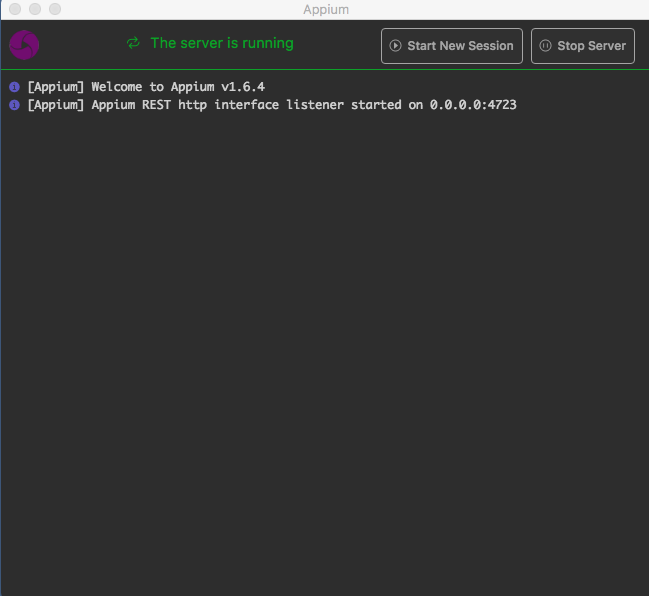
接下来开始你的hello world!之旅吧

六、启动Safari

1.打开pycharm，写如下代码



2.appium要是开着的



七、**WebDriverAgent**

1.第一次启动脚本，会在手机上自动安装一个**WebDriverAgent**，如下图所示，一定要看到这么个玩意，要不然肯定不成功

2.WebDriverAgent 在 iOS 端实现了一个 [WebDriver server](https://w3c.github.io/webdriver/webdriver-spec.html) ，借助这个 server 我们可以远程控制 iOS 设备。

上一篇已经可以启动iOS模拟器上的safari浏览器了，启动app比启动浏览器要复杂一点，本篇以github上的源码为案例详细介绍如何启动iOS模拟器的app

一、clone源码

1.github上Appium的案例源码地址：[https://github.com/appium/sample-code.git](http://www.cnblogs.com/yoyoketang/p/appium%E6%BA%90%E7%A0%81%E4%B8%8B%E8%BD%BD%E7%82%B9%E8%BF%99%E9%87%8C)

2.确保mac上已经安装了git  (不会安装的自己想办法解决吧)

3.在mac本地新建一个文件夹,如：git\_test\_app，把代码clone到本地

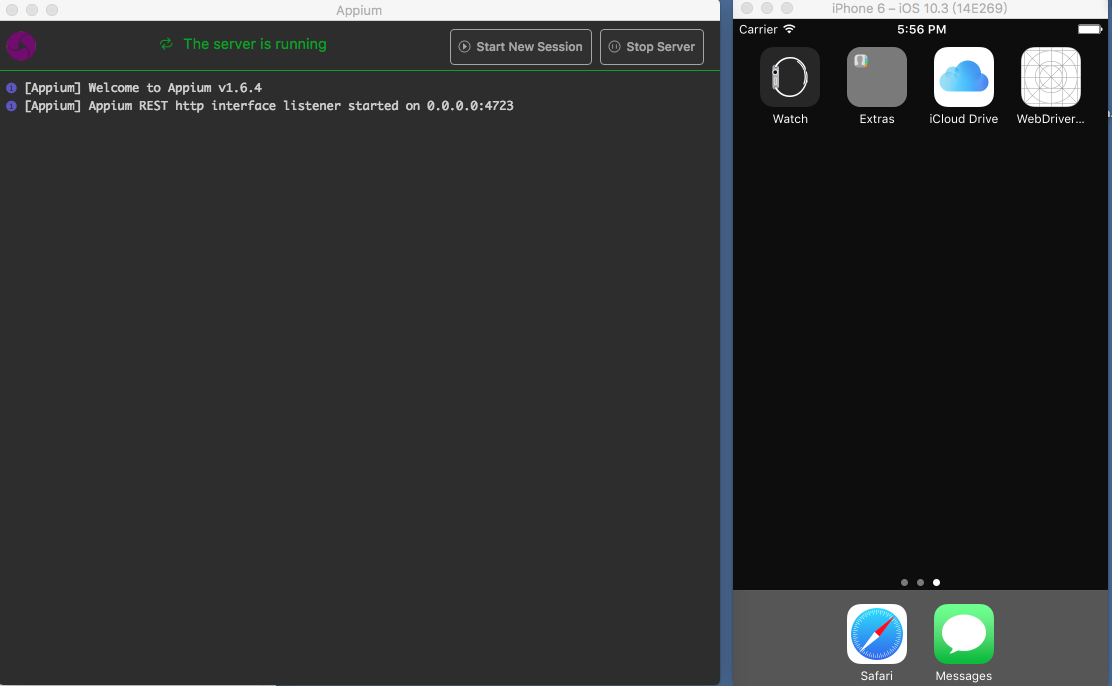
$ cd /git\_test\_app

$ git clone https://github.com/appium/sample-code.git

二、启动appium

1.确保appium是启动状态，如下左图

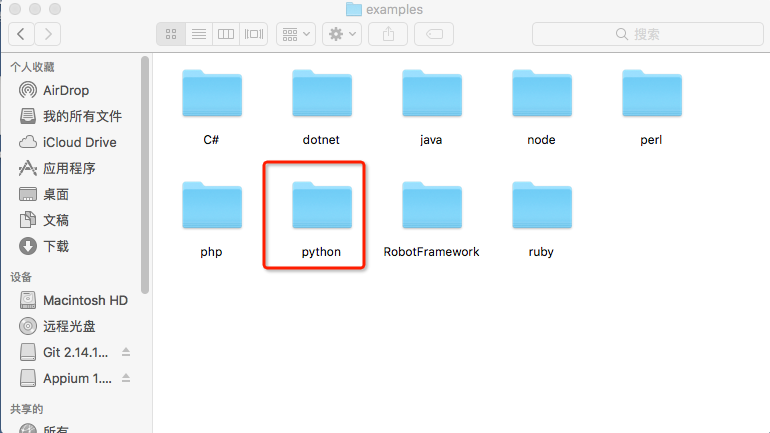
2.确保iOS模拟器是启动状态，如下右图



三、pycharm执行代码

1.在第一步clone下来的代码里，找到路径：/sample-code/examples/python，就是appium案例的python代码。

当然这里也有其它语言的，如：java、php、perl、ruby等，语言种类还是很全的。

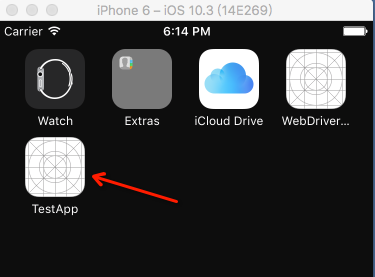


2.把找到的python这个项目包导入到pycharm里（自己想办法吧），找到ios\_simple.py这个脚本打开如下：

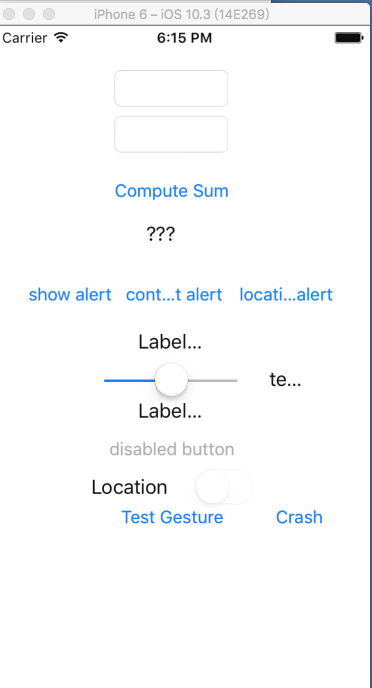
3.这里主要改2个地方（其它地方不要动），一个是platformVersion，保持跟右边模拟器iOS系统版本号一致，

第二个是deviceName，如右图所示，我这里是iPhone 6

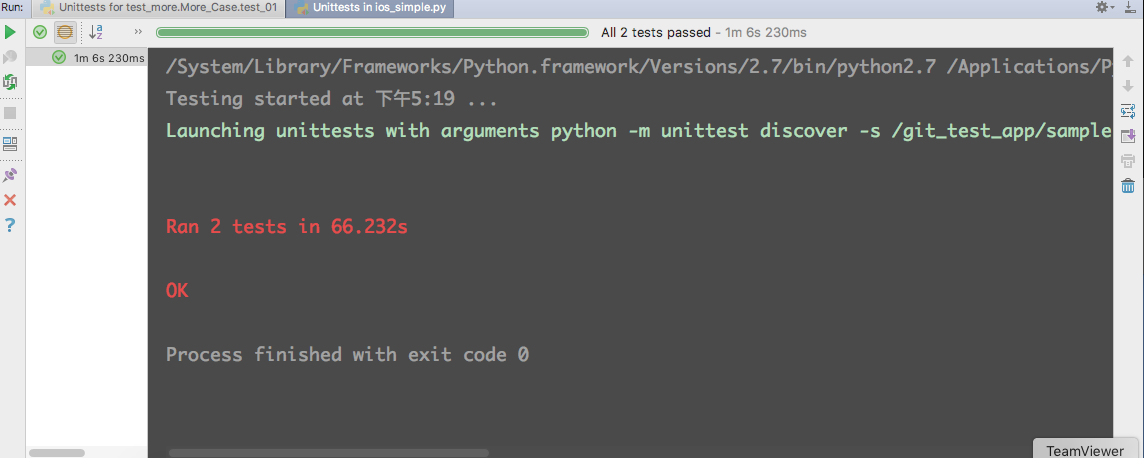
4.运行代码后在模拟器上会安装一个TestApp



5.运行时，手机上界面如图



6.pycharm运行结果：



四、终端执行

1.除了在pycharm执行，也可以直接在终端执行

$ cd /git\_test\_app/sample-code/sample-code/examples/python

$ python ios\_simple.py

 在学习过程中有遇到疑问的，可以加appium（python+java） QQ群交流:512200893

本篇安装brew、carthage,解决启动appium时的报错问题，另外安装appium-doctor检查appium的环境

1.brew

2.carthage

3.appium-doctor

一、brew安装

1.brew是一个软件包管理工具,类似于centos下的yum或者ubuntu下的apt-get,类似于python之pip或java之maven，就是这么回事。

2.brew的安装是需要用ruby安装的，ruby这玩意mac自带了，终端输入：ruby -v 可查看版本号

3.继续输入这个这个指令安装brew：ruby -e "$(curl --insecure -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

$ ruby -v

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

4.安装完成后查看版本号

$ brew -v

4.卸载指令：

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

二、carthage

1.直接用上一步安装的brew去安装carthage

$ brew install carthage

三、appium-doctor

1.用之前安装cnpm去安装appium-doctor,先确定版本号

$ cnpm -v

$ cnpm install appium-doctor

2.用appium-doctor去检查appium的环境，这个之前玩过windows上的appium环境应该都熟悉

$ appium-doctor -v

$ appium-doctor

3.上图除了红色框框区域，上面的显示√才是正常的，显示X就得想办法解决了。

上图红色框框区域都是一些android的环境，显示X也没关系，要是不打算再mac上安装android环境可以暂时不用管。

做过iOS上app测试的小伙伴应该都知道，普通用户安装app都是从appstore下载安装，安装测试版本的app，一般就是开发给的二维码扫码安装，

或者开发给个.ipa的安装包文件，通过itools安装。对于没有iPhone真机的小伙伴，想在iOS模拟器上安装app测试可不是那么容易的事情。

一、启动iOS Simulator

1.Xcode>Open Developer Tool>iOS Simulator，打开后在hardware里面选对应的iPhone版本就能启动模拟器了

2.第二种方法，通过指令启动模拟器：

$ xcrun instruments -w 'device name'

注意：这里的device name一定要是这种格式：iPhone 6 (10.3)

二、拿到.app包

1.这里科普下了（敲黑板，记重点），在真机上是安装.ipa的包，在模拟器上是安装.app的包。

这是两种不同的打包方法，也就是说从appstore这种渠道下载的.ipa是无法安装到模拟器上的，必须要找你们开发专门针对iOS模拟器打个模拟器的包，后缀是.app 。

（别问我是怎么知道的，都是泪~~~）

2.以[https://github.com/appium/sample-code.git](http://www.cnblogs.com/yoyoketang/p/appium%E6%BA%90%E7%A0%81%E4%B8%8B%E8%BD%BD%E7%82%B9%E8%BF%99%E9%87%8C) 这里的一个.app包为例，在路径/sample-code/apps/TestApp/build/release-iphonesimulator下有个TestApp.app

这种才是可以直接安装到iOS模拟器上的

3.通过指令安装到模拟器上：

$ xcrun simctl install booted /path/to/xxx.app

备注：/path/to/xxx.app 这个是.app包的绝对路径，可以输入前面的指令，然后找到这个包，直接拽到终端就行

4.卸载app的指令：

$ xcrun simctl uninstall booted xxx.app

 xxx.app就是包名，这里不需要路径，如：TestApp.app

三、xcode-select

1.要是遇到xcrun报错:xcrun: error: active developer path ("/Applications/Xcode.app/Contents/Developer") does not exist

就根据报错提示，输入提示的指令解决：

$ sudo xcode-select --switch path/to/Xcode.app

同样，这里path/to/Xcode.app也是Xcode在application里的绝对路径，直接找到图标拽到终端就行

2.或者输入这个也可以：

$ xcode-select --install

四、总结

1.在iOS模拟器上必须是通过xcode打出来的.app的安装包，真机的包（.ipa）选择Debug-iphone os,模拟器的包（.app）选择Debug-iphone Simulator

2.有以下几个方法都能安装到模拟器上：

--运行脚本，会自动安装上去，参考：[Appium+python自动化17-启动iOS模拟器APP源码案例](http://www.cnblogs.com/yoyoketang/p/7545824.html)

--通过指令安装，也就是这篇讲的

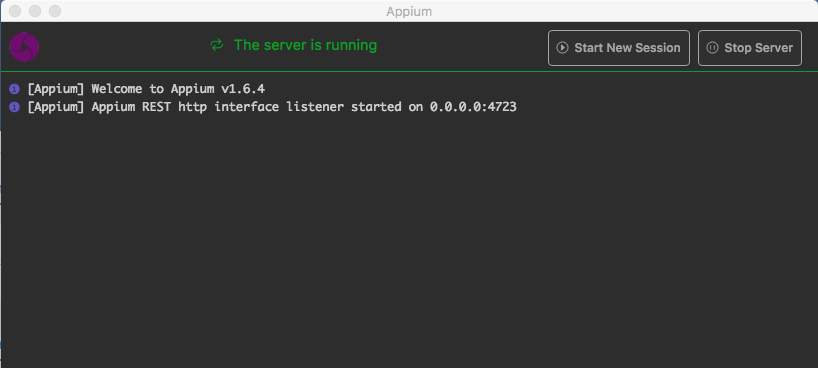
--通过appium上的配置也能安装，下篇讲appium调试app时候定位元素

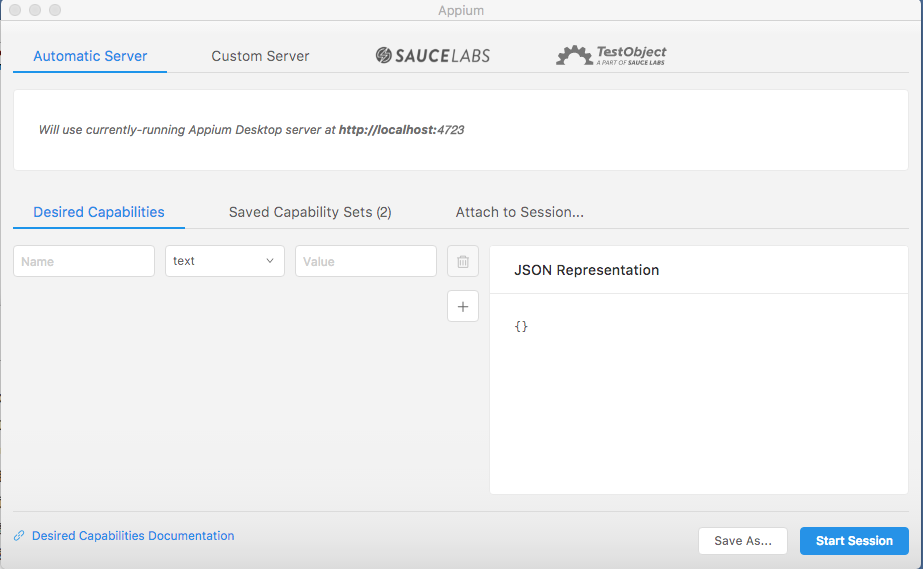
--要是会xcode，可以直接拿到拿到源码，用xcode直接运行上去

学UI自动化首先就是定位页面元素，玩过android版的appium小伙伴应该都知道，appium的[windows版自带的Inspector](http://www.cnblogs.com/yoyoketang/p/6128803.html)可以定位app上的元素  
Mac版的appium1.6的版本在UI上有了很大的改变，本篇详细解决如何使用appium1.6定位iOS的app页面元素。

**一、Start New Session**

1.启动appium后点Start New Session



2.打开后看到如下界面，刚开始看到这个界面，我也比较懵。  


**二、Desired Capabilities**

1.Desired Capabilities这里的配置，参考之前[Appium+python自动化17-启动iOS模拟器APP源码案例](http://www.cnblogs.com/yoyoketang/p/7545824.html)这篇里面启动app的代码  
```python  
class SimpleIOSTests(unittest.TestCase):

def setUp(self):

# set up appium

app = os.path.abspath('../../apps/TestApp/build/release-iphonesimulator/TestApp.app')

self.driver = webdriver.Remote(

command\_executor='http://127.0.0.1:4723/wd/hub',

desired\_capabilities={

'app': app,

'platformName': 'iOS',

'platformVersion': '10.1',

'deviceName': 'iPhone 6'

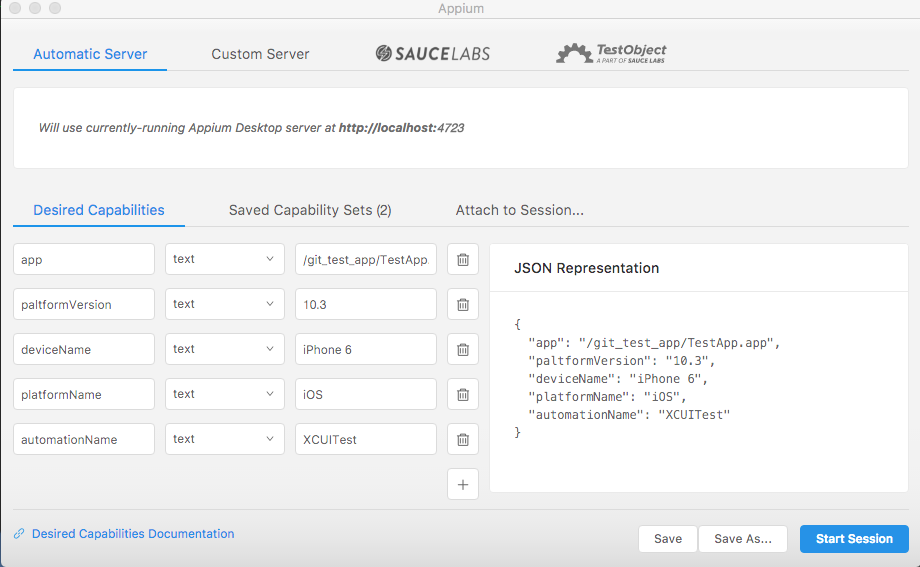
})

def tearDown(self):

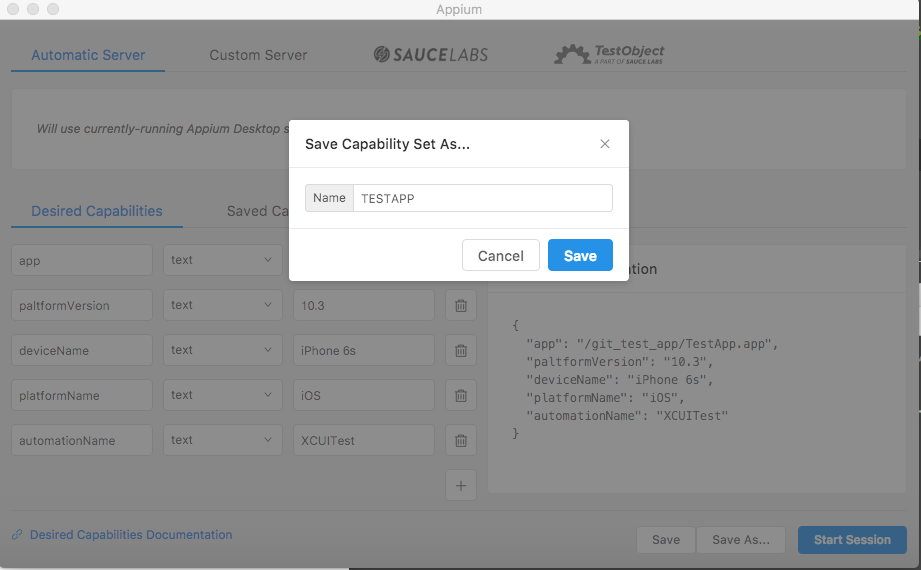
self.driver.quit()

```  
这里主要有四个参数：app、platforrmName、platforrmVersion、deviceName，在appium上需多加一个参数automationName  
> **appium1.6以前的版本是基于uiautomation框架的，1.6以后的版本是基于facebook的XCUITest，所以这个参数必须添加**  
- app: 这里的地址是本地.app包的绝对路径  
- platforrmName: iOS系统版本号  
- platforrmVersion: iPhone设备名称  
- platforrmVersion: iOS系统  
- automationName: XCUITest

2.在Desired Capabilities先填入对应的参数后，右侧会自动生成JSON Representation

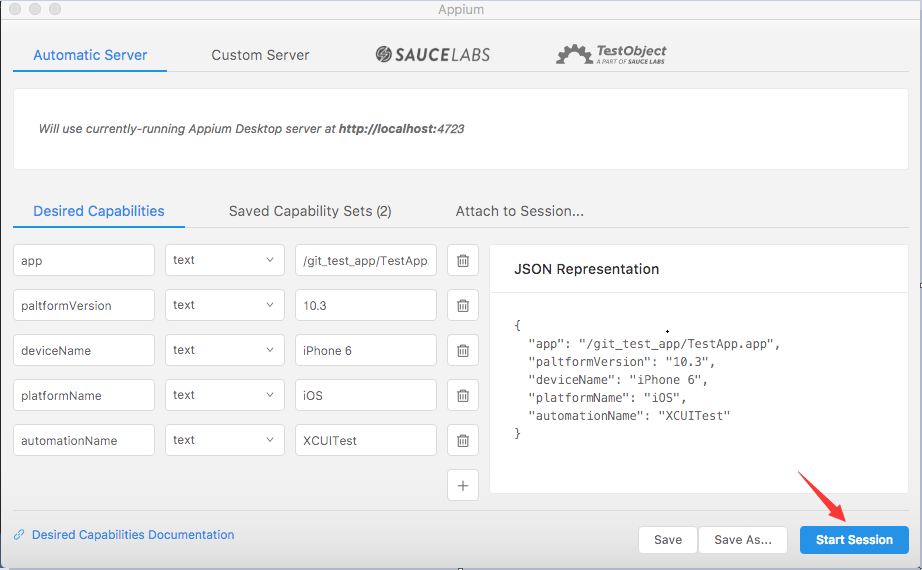


3.填好配置后，为了下次方便使用，可以点Sav As ..按钮保存设置



**三、Inspector**

1.Start Session就能启动Inspector界面了



2.接下来就可以定位app上的元素了

