## 说明

1. 本次的Appium搭建环境是在MAC和Windows环境
2. 安装Appium的两种方式是，第一种官方已经编译好的安装包的方式（含GUI窗体），第二种使用在线命令的方式安装（无GUI窗体）
3. Appium使用在线命令安装的方式，优点：

* 及时使用最新发布的版本
* 利于持续集成

## Jdk，Android sdk安装

安装jdk；Android sdk问开发要压缩包，文件太大

## MAC环境搭建

### MAC环境变量配置

配置环境变量，在根目录新建一个.bash\_profile文件，如下配置

vi .bash\_profile  
JAVA\_HOME=`/usr/libexec/java\_home` *# jdk安装目录，用命令自动获取，也可收到指定绝对地址*export ANDROID\_HOME=~/Downloads/sdk *# sdk根目录*export ANDROID\_SDK\_HOME=~/Downloads/sdk  
export PATH=${PATH}:${ANDROID\_HOME}/tools   
export PATH=${PATH}:${ANDROID\_HOME}/platform-tools  
export PATH=$ANDROID\_HOME/build-tools/23.0.1:$PATH  
export JAVA\_HOME  
export PATH=$JAVA\_HOME/bin:$PATH  
CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

设置环境变量后，输入命令source .bash\_profile让环境配置生效

### 安装Homebrew

macOS上的软件使用Homebrew进行安装，使用brew安装过的软件，不再需要sudo命令。打开终端输入：

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

### 安装Nodejs

brew install node

检查是否安装成功

node -v  
*# v6.9.1*npm -v  
*# 3.10.8*

### 安装cnpm

由于npm源在国外，我们使用npm安装库时会比较慢，可以使用taobao的镜像源

npm install -g cnpm --registry=https://registry.npm.taobao.org

检查安装是否成功

cnpm -v  
*# 4.4.0*

以后均使用cnpm 命令代替npm

### 安装Carthage

Carthage 是一个管理 iOS 开发库依赖的包

brew install carthage

### 安装Python

安装Python3

brew install python3

检测安装成功

python3 -V  
*# Python 3.6.0*pip3 -V  
*# pip 9.0.1 from /usr/local/lib/python3.6/site-packages (python 3.6)*

如果系统版本之前安装过python3导致安装后输入python3没有找到命令，可以尝试brew link --overwrite python3重新链接

### 安装appium server

appium server本质上就是一个Nodejs应用，我们可以使用cnpm对其进行安装，安装完毕后就可以使用命令行启动

cnpm install -g appium

### 安装appium doctor

可以使用appium-doctor来确认安装环境是否完成

shark-Mac-mini:~ cnpm install -g appium-doctor  
shark-Mac-mini:~ shark.shi$ appium-doctor  
info AppiumDoctor Appium Doctor v.1.4.3  
info AppiumDoctor ### Diagnostic starting ###  
**info AppiumDoctor ✔ The Node.js binary was found at:** /usr/local/bin/node  
info AppiumDoctor ✔ Node version is 7.6.0  
**info AppiumDoctor ✔ Xcode is installed at:** /Library/Developer/CommandLineTools  
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/shark.shi  
**info AppiumDoctor ✔ ANDROID\_HOME is set to:** /Users/shark.shi/Downloads/sdk  
**info AppiumDoctor ✔ JAVA\_HOME is set to:** /Library/Java/JavaVirtualMachines/jdk1.8.0\_91.jdk/Contents/Home  
**info AppiumDoctor ✔ adb exists at:** /Users/shark.shi/Downloads/sdk/platform-tools/adb  
**info AppiumDoctor ✔ android exists at:** /Users/shark.shi/Downloads/sdk/tools/android  
**info AppiumDoctor ✔ emulator exists at:** /Users/shark.shi/Downloads/sdk/tools/emulator  
info AppiumDoctor ✔ Bin directory of $JAVA\_HOME is set  
info AppiumDoctor ### Diagnostic completed, no fix needed. ###  
info AppiumDoctor   
info AppiumDoctor Everything looks good, bye!  
info AppiumDoctor

## Win环境搭建

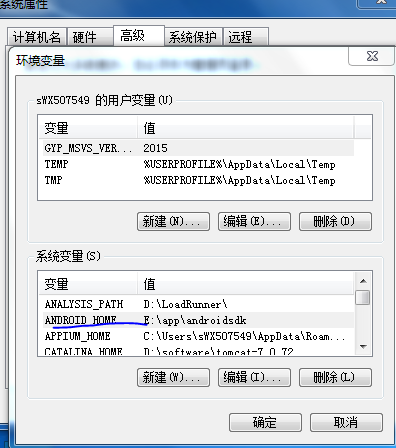
### 安装Python2.7

<https://www.python.org/downloads/>

在用npm安装appium时，必须依赖python2.7

### Win环境变量配置

鼠标右击我的电脑-属性-高级设置-环境变量新建变量并设置值：



PYTHON: e:\app\python27  
ANDROID\_HOME: e:\app\android\sdk  
JAVA\_HOME: e:\app\java\jdk1.8  
PATH: ;%PYTHON%;%PYTHON%\Scripts;%ANDROID\_HOME%\tools;%ANDROID%\platform-tools;%ANDROID\_HOME%\build-tools\23.0.3;

### 安装node.js

<https://nodejs.org/en/download/>

node -v  
*# v6.9.1*npm -v  
*# 3.10.8*

### 安装Appium Server

由于npm源在国外，我们使用npm安装库时会比较慢，可以使用taobao的镜像源

npm config set registry http://registry.npm.taobao.org  
npm install -g appium

安装过程中可能会出现vcBuild.exe错误，因为在 windows环境下，需要用到.net的一些编译环境，可以用如下命令解决：

npm install -g windows-build-tools

解决后，重新用npm install -g appium进行安装

### 安装Appium doctor

可以使用appium-doctor来确认安装环境是否完成

npm install -g appium-doctor

安装成功后，检查appium的环境是否正常

C:\Users\sWX507549>appium-doctor  
info AppiumDoctor Appium Doctor v.1.4.3  
info AppiumDoctor ### Diagnostic starting ###  
info AppiumDoctor ✔ The Node.js binary was found at: E:\app\nodejs\node.exe  
info AppiumDoctor ✔ Node version is 8.9.1  
info AppiumDoctor ✔ ANDROID\_HOME is set to: E:\app\androidsdk  
info AppiumDoctor ✔ JAVA\_HOME is set to: D:\software\Java\jdk1.8.0\_102\_x64  
info AppiumDoctor ✔ adb exists at: E:\app\androidsdk\platform-tools\adb.exe  
info AppiumDoctor ✔ android exists at: E:\app\androidsdk\tools\android.bat  
info AppiumDoctor ✔ emulator exists at: E:\app\androidsdk\tools\emulator.exe  
info AppiumDoctor ✔ Bin directory of %JAVA\_HOME% is set  
info AppiumDoctor ### Diagnostic completed, no fix needed. ###  
info AppiumDoctor  
info AppiumDoctor Everything looks good, bye!  
info AppiumDoctor

## 安装Appium Client

1. 如果是windows系统,还是建议再次安装Python3，然后设置环境变量，PYTHON: e:\app\python3.5.1
2. 安装Appium-Python-Client

pip3 install Appium-Python-Client

## 推荐IDE

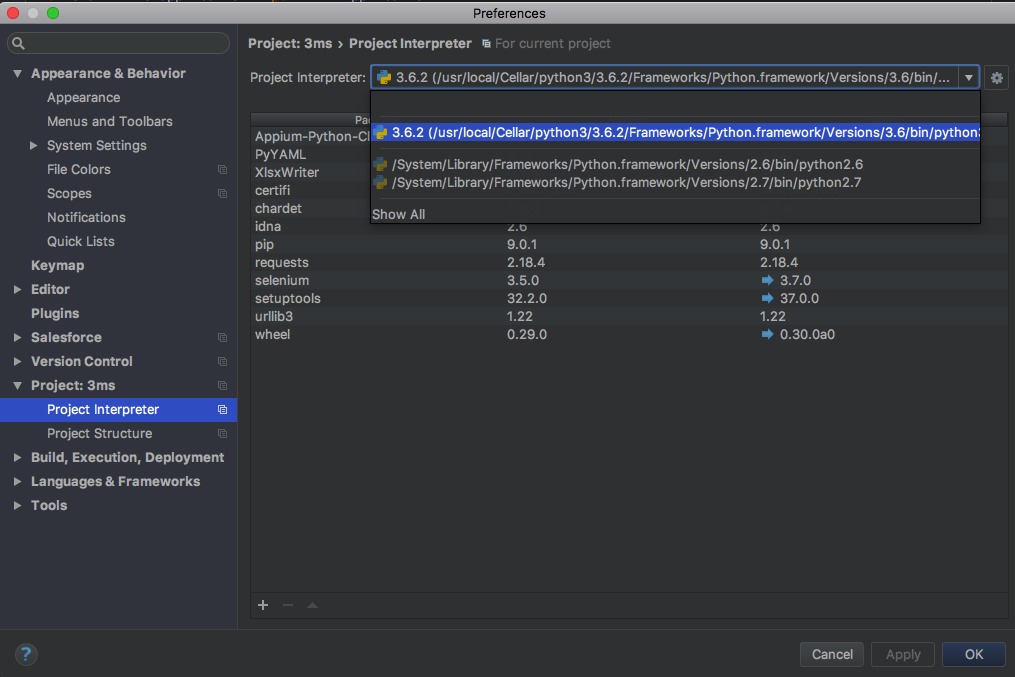
[PyCharm](http://www.jetbrains.com/pycharm/)

安装简单，调试方便

## Python版本选择

建议用 Python3 版本进行开发，所以电脑上请安装Pyhon3

1. MAC下打开Pycharm的preference-选择-project-interpreter-选择python版本
2. Windwos下打开Pycharm的file-setting-选择-project-interpreter-选择python版本



## 实例

### 启动appium server

直接输入appium启动server

shark-Mac-mini:~ shark.shi$ appium  
[Appium] Welcome to Appium v1.6.5  
[Appium] Appium REST http interface listener started on 0.0.0.0:4723

### 检查手机设备

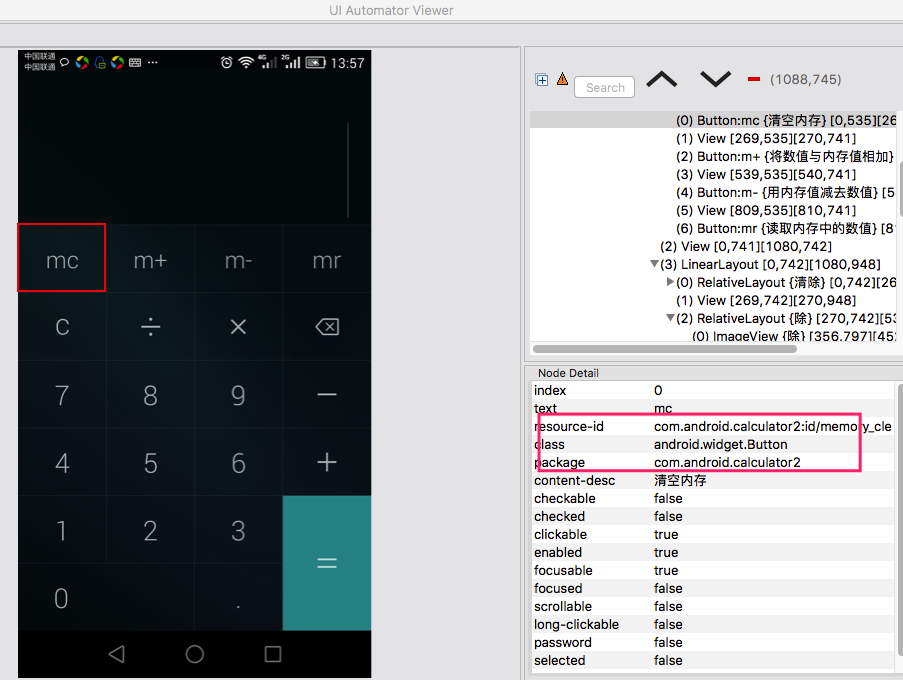
Android手机打开计算器app并连接电脑

adb devices  
List of devices attached  
DU2TAN15AJ049163 device # 设备名

### 获取页面元素

1. 进入sdk下的tools目录
2. 启动uiautomatorviewer工具，获取计算器app页面元素

cd ~/Download/sdk/tools/  
./uiautomatorviewer



### 客户端代码

运行代码看测试效果

import unittest

import time  
from appium import webdriver  
class AndroidTests(unittest.TestCase):  
 def setUp(self):  
 desired\_caps = {}  
 desired\_caps['platformName'] = 'Android' # 平台  
 desired\_caps['platformVersion'] = '4.4.2' # 手机版本  
 desired\_caps['deviceName'] = 'DU2TAN15AJ049163' # 设备名  
 desired\_caps['appPackage'] = 'com.android.calculator2' # app包名  
 desired\_caps['appActivity'] = '.Calculator' # 启动类  
 remote = "http://127.0.0.1:" + "4723" + "/wd/hub"  
 self.d = webdriver.Remote(remote, desired\_caps)  
 def tearDown(self):  
 self.d.quit()  
 def test\_plus(self):  
 self.d.find\_element\_by\_xpath("//android.widget.Button[@text='1']").

click()  
 self.d.find\_element\_by\_id("com.android.calculator2:id/plus").click()

self.d.find\_element\_by\_xpath("//android.widget.Button[@text='9']").

click()

self.d.find\_element\_by\_id("com.android.calculator2:id/equal").click()  
 time.sleep(1)

result=self.d.find\_element\_by\_xpath

("//android.widget.EditText[@text='10']")  
 if result is not None:  
 print("测试通过")

# 如果是Python2，用print “msg”,建议用python3  
 else:  
 print("测试失败")  
if \_\_name\_\_ == '\_\_main\_\_':  
 suite = unittest.TestLoader().loadTestsFromTestCase(AndroidTests)  
 unittest.TextTestRunner(verbosity=2).run(suite)