# Appium环境搭建

需要用到的软件如下：

1. jdk-8u121-window(32位的就下载32位的，64位的就下载64位的)。

2.Android-sdk\_r24.3.4-windows（下载这个adt-bundle-windows-x86-20140624）

3.Python:（V3.6也可以）

4.appium：1.4.13.1

5.Node.js：node-v4.4.7-x64

6.Appium-python-Client

7.pycharm

其中1-5是可以先下载的，6则需要在完成前面的五个步骤后才进行，7为可选Ide工具。

2.1 JDK

下载好jdk直接点下一步就可以了，然后开始配置变量，参考如下路径：http://jingyan.baidu.com/article/d45ad148ba5ab169552b80d3.html

 注意：

classpath值：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

path值：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin

.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;是有“.”这个符号的不能漏掉

按快捷键win+r，在运行内输入cmd，并输入Java，回车，如果出来的是些用法中文，那说明这是正确的，在接下来输入javac，回车，如果出现些用法中文，说明是环境变量配好了。

2.2 下载SDK

http://www.androiddevtools.cn/

直接下载adt/Android studio完美集成，下载好后并进行解压，然后开始配置三个变量：

1.解压adt集成包到合适的路径下面，比如D:\adt\

2.设置安卓环境变量

配置环境变量，

设置ANDROID\_HOME系统变量为你的android SDK路径，并把tools和platform-tools两个目录加入到系统的Path路径里。

eg:添加变量名：ANDROID\_HOME 变量值: D:\adt\sdk

并把这句添加到 %ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools记得要加上";"

配置完环境后，可以在运行cmd后输入 adb(能看到当前版本号)，再输入android-h（tools目录下的，如果出来一堆文字则是可以的）

2.3 安装python

进入官网或百度下载python下载好后安装到所需安装路径，如在D盘中新建一个python文件夹，如d:\python34

安装完成后将D:\python和D:\python\Scripts，添加到环境变量path下

打开cmd输入python,出现版本号，然后输入print("helloworld!")这样python就是安装好了。

2.4 安装node.js

进入官网地址并下载：https://nodejs.org/en/download/

下载后一路傻瓜式安装，安装完成后，运行cmd，输入node –v查看版本号，然后输入npm

2.5 安装appium服务端

方法一：进入官网地址并下载

https://bitbucket.org/appium/appium.app/downloads/

完整安装带UI的appium，可以从官网直接下载dmg（mac）或者zip（Windows），运行里面的app即可。

appium安装好后：

找到这个文件安装目录D:\appium\node\_modules\.bin

将上面的地址添加到环境变量path下；

方法二：node.js包管理安装

npm install -g appium

appium –v

npm install -g appium-doctor

appium-doctor

命令安装的appium是控制台程序，没有UI界面。可到github搜索appium的appiumDesktop项目中下载最新版本

但是由于GFW关系，此方法会在安装过程中卡住或是直接报错，解决办法：挂VPN代理FQ；为了满足国内开发人员的需要，淘宝npm镜像cnpmjs.org可替代官方版本。打开输入:

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

方法三：百度pan

直接下载appium的GUI界面安装包，下载地址：https://pan.baidu.com/s/1jGvAISu#list/path=%2F，Windows最新版本是AppiumForWindows\_1.4.16.1.zip，MacOS最新版本是appium－1.5.3.dmg。笔者以windows为例，下载的AppiumForWindows\_1.4.16.1.zip 进行解压安装，启动appium，弹出错误提示框

很容易在网上查到错误原因，因为node.js需要.NETFramework框架的支持，所以在这里需要安装net framework4.5。当.net framework 4.5安装完成，再次启动Appium，

打开appium

appium安装好后：

找到这个文件安装目录D:\appium\node\_modules\.bin

将上面的地址添加到环境变量path下；

Tips:检查appium的所有需求环境

npm install -g appium-doctor

appium-doctor

打开cmd，输入appium-doctor，检查环境是否OK，出现allchecks were successful，说明环境OK；

2.6 安装Appium-Python-Client

进入cmd 输入：pip install Appium-Python-Client。通过Python安装等待安装完成。

如果从官方的资源库下载Appium－Python－Client失败，请自行更换下载源重新下载。在命令行输入"pip install Appium-Python-Client -ihttp://pypi.douban.com/simple"重试

Tips:检查selenium版本3.3.3

安装 Appium-Python-Client 的同时会安装一个selenium模块.试着进入python3交互命令行,然后执行下面命令:

import selenium

selenium.\_\_version\_\_

如果你显示的 selenium 版本是当前最新版本3.3.3的话,那么最好使用下面命令将版本装成3.0.2，网传某些Api的话会报错。

pip3 install -I selenium==3.0.2

2.7 安装pycharm

安装流程参考：http://www.jianshu.com/p/042324342bf4

需要激活可以购买正版或网上搜索激活码。

三，简单自动化测试用例编写

3.1 新建Python unit test文件

新建Python unit test类型文件，对test类补充setup及teardown函数

对类MyTestCase添加setUp函数（这是测试用例执行前的准备动作，负责告诉appium监听那个端口、与那个手机连接、安装那个应用等信息）

对类MyTestCase添加TearDown函数（这是测试用例执行结束后的动作，可以执行保存执行结果等操作）

3.2 编写测试用例

简单实例点位两个控件，执行点击事件

# MyTestCase.py

#!/usr/bin/env python

# \_\*\_ coding:utf-8 \_\*\_

import unittest

import selenium

import time

from appium import webdriver

class MyTestCase(unittest.TestCase):

@classmethod

def setUp(self):

# super().setUp()

print('selenium version = ', selenium.\_\_version\_\_)

desired\_caps = {}

desired\_caps['platformName'] = 'Android'

desired\_caps['platformVersion'] = '4.4'

desired\_caps['deviceName'] = '192.168.1.54:5555'

desired\_caps['appPackage'] = 'com. xx'

#desired\_caps['app'] = 'F:// debug.apk'

desired\_caps['appActivity'] = 'com.xx.MainActivity'

self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)

def test\_something(self):

print('test\_something click ------ ')

# xpath：

time.sleep(2)

self.driver.find\_element\_by\_xpath("//android.widget.ListView/android.widget.TextView[contains(@text,'测试')]").click()

# uiautomator -UiSelector：

# name方式在1.5版本后已废除，能找到接口，不可使用，使用new UiSelector().text替代

# self.driver.find\_element\_by\_android\_uiautomator("newUiSelector().text(\"测试\")").click()

# class\_name - child：

# items =self.driver.find\_elements\_by\_class\_name('android.widget.TextView')

# items[1].click()

# id:

time.sleep(2)

self.driver.find\_element\_by\_id('com.hisense.vod:id/test\_video\_resize').click()

@classmethod

def tearDown(self):

time.sleep(5)

print('tearDown ------ ')

self.driver.quit()

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

四，踩坑

1 selenium版本3.3.3兼容性

安装 Appium-Python-Client 的同时会安装一个selenium模块.试着进入python3交互命令行,然后执行下面命令:

import selenium

selenium.\_\_version\_\_

结果

如果你显示的 selenium 版本是当前最新版本3.3.3的话，那么最好使用下面命令将版本装成3.0.2

pip3 install -I selenium==3.0.2

否则的话，在使用Appium-Python-Client的某些Api的话会报错.应该是版本兼容性的问题，3.11已修复的。

2 findElementByName无效.

Searching by name was deprecated over ayear ago and removed from 1.5. In general, searching by accessibility id isbetter for a variety of reasons.

如上findElementByName这个方法从Appium 1.5之后删除了,但是API不经能找到并且也没提示过时，这不坑爹嘛。后来使用下面的代码才解决用name，即text查找元素的方法。

java：

1

2

String query = "new UiSelector().textContains" + "(\"" + locator.value + "\")";

webElements = mDriver.findElementsByAndroidUIAutomator (query);

Python：

self.driver.find\_element\_by\_android\_uiautomator("newUiSelector().text(\"测试\")").click()

3 据说Appium 1.6.3可以查找 Toast 的信息了.

试了下网上的例子发现不好使，一度以为是Client版本的问题。搞了半天才发现需要加下面的代码：

java：

1

capabilities.setCapability (MobileCapabilityType.AUTOMATION\_NAME, AutomationName.ANDROID\_UIAUTOMATOR2);

4 安装的时候重新签名

无意中发现测试的时候弹Toast报错，然后直接编译安装却不保存，猜测是不是在安装过程中Appium改了啥，看了下Service日志，竟然在安装的时候重新签名…

App not signed with debug cert.

2017-02-13 18:17:19:848 - info: [debug] [ADB] Resigning apk.

2017-02-13 18:17:23:938 - info: [debug] [ADB] Zip-aligning 'app-debug.apk'

2017-02-13 18:17:24:104 - info: [AndroidDriver] Remote apk path is /data/local/tmp/463eb03788048b4a1dacfe28545ee76e.apk

解决方法：

java：

capabilities.setCapability(AndroidMobileCapabilityType.NO\_SIGN, true);

————————————————

版权声明：本文为CSDN博主「起个昵称那么难」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_16206535/article/details/79586818