Appium 自动化测试环境配置(附实例)

（python）

 本文档对appium自动化测试平台的环境搭建给出指导。自动化测试的语言选择python（python3.4）,系统环境为Win7(x64)，除了环境配置外，最后给一个简单的appium实例，讲解了详细的过程，大家可以运行一下。

1. 配置环境步骤说明
2. 说明：

整个环境搭建分为 java环境，安卓开发环境，安卓虚拟机 ，python，appium 五个主要的步骤

1. 软件准备

JDK : 官网下载（官网URL）（记得要下载X64版本的），当前使用jdk-7u80-windows-x64

云盘下载地址： http://pan.baidu.com/s/1bPdeO2

安卓开发环境：直接使用adt集成包，官网下载（官网URL）或者这个路径下载http://www.androiddevtools.cn/

，当前使用dt-bundle-windows-x86\_64-20140321

 云盘下载地址：http://pan.baidu.com/s/1bphXWsf

安卓虚拟机：Gemotion（可选，不必须）

 云盘下载地址：

Python：官网下载（官网URL）当前使用python3.4

 云盘地址：http://pan.baidu.com/s/1slwy5zR

 Appium: appium 依赖于node.js，其下载地址<http://nodejs.org/download/>）

 云盘地址：

 Appium 下载地址：

1. Java环境配置
2. 安装jdk和jre，尽量不要选择默认路径的program files路径下面（路径中带空格可能会引起不可预见的问题）。
3. 设置环境变量

计算机→属性→高级系统设置→高级→环境变量

1. 系统变量→新建 JAVA\_HOME 变量。

变量值填写jdk的安装目录（本人是 E:\Java\jdk1.7.0)

1. 系统变量→寻找 Path 变量→编辑

在变量值最后输入 %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

（注意原来Path的变量值末尾有没有;号，如果没有，先输入；号再输入上面的代码）

1. 系统变量→新建 CLASSPATH 变量

变量值填写   .;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar（注意最前面有一点）

1. 检验是否配置成功 运行cmd 输入 java -version （java 和 -version 之间有空格）

显示版本信息 则说明安装和配置成功。

1. 安卓开发环境配置
2. 解压adt集成包到合适的路径下面，比如D:\adt\
3. 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。
添加变量名：ANDROID\_HOME 值: D:\adt\sdk
编辑添加Path值： %ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools

1. 启动sdk manager 更新sdk

如何更新sdk,请参考<http://www.androiddevtools.cn/>

1. 创建安卓虚拟机

自动化测试可以使用真机，也可以使用虚拟机。安卓虚拟机可以使用安卓环境自带的虚拟机，也可以使用专门的虚拟机管理软件gemotion，相对而言，gemotion 要快一些。

1. 创建安卓环境自带虚拟机

找到sdk目录下的AVD Manager.exe 启动，点击NEW，编辑虚拟机的选项，创建完成，在虚拟机列表中找到此虚拟机，点击start 既可启动虚拟机，也可以使用此命令 emulator –avd test (test是你命名的avd name)



1. 安装下载gemotion

这个比较简单，可参考百度经验 http://jingyan.baidu.com/article/3ea51489e7d8bd52e61bba36.html

1. Python安装
2. 安装python3.4，很简单，一直下一步就安装完成，比如路径为 d:\python34
3. 设置python环境变量,添加python的安装路径到path变量中
4. 检查环境变量：打开cmd，输入python，如果出现以下界面，则说明你的python安装成功了（可以查看到你的python版本）



1. Appium安装

Appium需要首先安装依赖的程序node.js，然后安装appium，然后安装用到的扩展包seleium和 appium-python-client

1. 安装依赖 node.js

安装很简单，一直下一步就OK。安装完成以后，可以检查 Node 版本安装是否成功：

进入 CMD，输入 node -v, 可以看到版本号，不报错就说明安装成功了。

1. 安装Appium

第一种方法安装Appium。

进入 cmd 命令行，输入：

npm --registry http://registry.cnpmjs.org install -g appium

注：多等几分钟，可以看到 appium 的版本 1.1.0 及安装目录

第二种方法安装Appium。

下载Appium安装程序，一直下一步安装完成。

安装完成后配置appium环境变量

Path     D:\tools\appium\Appium\node\_modules\.bin;

测试：CMD中输入appium-doctor，能看到All Checks were successful

1. Python3以上的版本中自带seleium,以下的版本需要手动安装seleium

Cmd中输入安装命令：pip install selenium –i <http://pypi.douban.com/simple>

安装完成后，打开 python 的 shell 或者 IDEL 界面 ，输入 from selenium import webdriver 如果不报错那就说明你已经安装 selenium for python成功了。

1. 安装 appium-python-client

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

1. 一个简单的appium实例

我们选用一个安卓虚拟机上自带的一个计算器的程序，整体的步骤如下

* + - 1. 启动建立好的虚拟机，当前使用安卓自带的虚拟机，虚拟机名称为test03,安卓版本是4.4.2。可以在cmd中输入 emulator –avd test03 就可以将该虚拟机启动，或者直接找到avd manager启动虚拟机。或者启动其他虚拟机，或则链接真机到PC。链接好后打开虚拟机上的计算器程序。

虚拟机启动后，在cmd中输入adb devices可以查看到deviceName

* + - 1. 启动sdk tools 下面的hierarchyviewer.bat工具，点击刷新，可以查看到计算器程序的appPackage是com.android.calculator2,当前启动的appActivity是com.android.calculator2.Calculator，这里面也可以查看到虚拟机名称test03和deviceName。



* + - 1. 以上步骤是查看了当前的环境配置的各项参数，接下来启动appium，点击右上角的启动按钮。



* + - 1. 元素识别： 启动sdk tools下面的uiautomatorviewer.bat工具，启动后点击左上角第二个按钮，获取device的屏幕快照。在左边的界面上选取任意的元素，右边显示识别到的元素的属性，比如 resource-id



* + - 1. 建立一个test.py的文件，粘贴以下代码，编辑下步骤1中提到的各项参数。此脚本实现是1+2=3的一个过程，只使用了最简单的元素识别的方法，应该很容易看懂。

#coding=utf-8

import os

from selenium import webdriver

PATH = lambda p: os.path.abspath(

 os.path.join(os.path.dirname(\_\_file\_\_), p)

)

desired\_caps = {}

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

desired\_caps['version'] = '4.4.2'

desired\_caps['deviceName'] = 'emulator-5554'

desired\_caps['appPackage'] = 'com.android.calculator2'

desired\_caps['appActivity'] = '.Calculator'

driver = webdriver.Remote('http://127.0.0.1:4723/wd/hub', desired\_caps)

driver.find\_element\_by\_id("com.android.calculator2:id/digit1").click()

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

driver.find\_element\_by\_id("com.android.calculator2:id/digit2").click()

driver.find\_element\_by\_id("com.android.calculator2:id/equal").click()

driver.quit()

* + - 1. 运行test.py的文件 ，在cmd中输入 python ，回车运行



* + - 1. OK，等待运行完成，查看虚拟机上的运行状态，完成后查看appium输出的日志研究下，这只是appium自动化的第一步。