参考网址一：**<http://www.cnblogs.com/fnng/p/4540731.html>**

参考网址二：http://www.cnblogs.com/Nefeltari/p/5603163.html

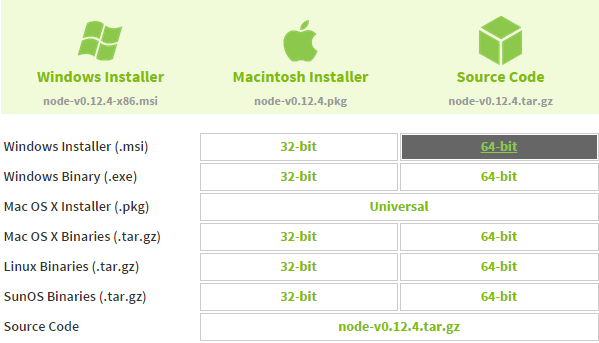
**第一节 安装Appium**

Appium官方网站：<http://appium.io/>

1. 先安装node.js

node.js官方网站：<https://nodejs.org/>

Download标签页面，左边是LTS稳定版（修复已知BUG），右边是current新版，对新技术的探索，可能有BUG。我们下载LTS。（百度的虫师的环境是win764位），我本来想用虚拟机搭，但是我觉得我还要用电脑做别的工作，同时开启虚拟机，会很卡。所以直接用本机的win10环境尝试。（后面一定要用jdk1.8）



根据你的操作系统选择相应的版本进行下载。这里我以Windows 7 (64) 为例进行安装，选择Windows installer(.msi)  64-bit 版本进行下载。

2.下载完成，双击进行安装，next即可。安装完成，打开Windows 命令提示符，敲入“npm”命令回车。



如果出现如上图信息，表示node.js安装成功。

1. 安装Appium

提供一个百度网盘的下载链接：<http://pan.baidu.com/s/1jGvAISu>

我们以Windows为例，将下载的AppiumForWindows\_1\_4\_0\_0.zip 进行解压，如下：

点击exe安装之前，要先把Microsoft dotNet4和4.5版本安装好。

百度“.net framework 4” ，进行下载安装。

百度“.net framework 4.5”，进行下载安装。

双击“appium-installer.exe”进行安装。根据提示，一步一步进行安装。最终在会桌面上生成Appium图标。



4.先配置Appium的环境变量，不然下面会不成功

直接在系统变量的Path的变量值后面加上（你的appium的安装路径）

D:\Program Files (x86)\Appium\node\_modules\.bin

**第二节 安装Android开发环境**

　　如果你的环境是MAC那么可以直接跳过这一节。就像我们在用Selenium进行web自动化测试的时候一样，我们需要一个浏览器来执行测试脚本。那么移动端自动化测试，我们同样需要一部Android手机（手机需要链接电脑）或者Android模拟器。

**1.安装java环境**

(ps：一定要下载最新的1.8版本的jdk。Android是由Java语言开发的)

官网下载jdk的地址（先选中accept…）：http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

java下载地址2：<http://www.java.com/zh_CN/download/manual.jsp>

打开下载链接选择相应的版本进行下载。

安装时可改安装路径

D:\Program Files (x86)\Java\jdk1.8.0\_25目录下。

2.下面设置环境变量：

“我的电脑”右键菜单--->属性--->高级--->环境变量--->系统变量-->新建..

|  |
| --- |
| 变量名：JAVA\_HOME  变量值：D:\Program Files (x86)\Java\jdk1.8.0\_25  变量名：CALSS\_PATH  变量值：.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar; |

找到path变量名—>“编辑”添加：

|  |
| --- |
| 变量名：PATH  变量值：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin; |

3.在Windows命令提示符下验证java是否成功：

cmd窗口中输入java –version

或者是java

然后是javac

Java命令可以运行class文件字节码。

javac命令可以将java源文件编译为class字节码文件

**2.安装android adt & SDK**

Android SDK提供了你的API库和开发工具构建，测试和调试应用程序，Android。简单来讲，Android SDK 可以看做用于开发和运行Android应用的一个软件。

|  |
| --- |
| 扩展理解：  ADT(Android Development Tools)： 目前Android开发所用的开发工具是Eclipse，在Eclipse编译IDE环境中，安装ADT，为Android开发提供开发工具的升级或者变更，简单理解为在Eclipse下开发工具的升级下载工具。adt只是一个eclipse的插件，里面可以设置sdk路径      SDK(Software Development Kit)： 是整体开发中所用到的工具包，如果你不用Eclipse作为你的开发工具，你就不需要下载ADT，只下载SDK即可开发。SDK可以自己编译，在linux环境下通过make命令进行，耗时比较长，需要有耐心哦亲。。。然后我们可以把自己编译的SDK通过ADT导入eclipse。在此基础上可以对源码包进行修改，比如修改android system/app/phone.apk中的源码， |

通过官方地址下载：<http://developer.android.com/sdk/index.html>

当然，我不能保证你可以访问到这个地址，原因你懂的！

这里提供adt-bundle 下载链接：

<http://dl.google.com/android/adt/adt-bundle-windows-x86-20140702.zip>

<https://dl.google.com/android/adt/adt-bundle-windows-x86_64-20140702.zip>

<http://dl.google.com/android/adt/adt-bundle-mac-x86_64-20140702.zip>

<http://dl.google.com/android/adt/adt-bundle-linux-x86-20140702.zip>

<http://dl.google.com/android/adt/adt-bundle-linux-x86_64-20140702.zip>

Android-sdk下载地址：

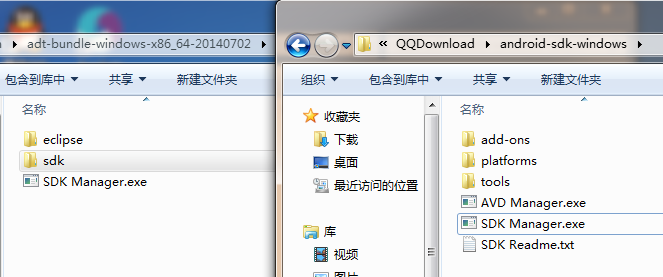
<http://dl.google.com/android/android-sdk_r23.0.2-windows.zip>

<http://dl.google.com/android/installer_r23.0.2-windows.exe(推荐用最新的24>的exe文件直接点击安装即可。)

<http://dl.google.com/android/android-sdk_r23.0.2-macosx.zip>

<http://dl.google.com/android/android-sdk_r23.0.2-linux.tgz>

根据我的验证adt和sdk 都可以用来运行android 模拟器，所以下载哪个都可以。下载链接中已经标识了系统信息，请根据自己的操作系统下载相应的版本。



Adt 和sdk 解压包对比，相对adt的内容要多一些，如果你不做android开发的话，两者都可以。

下载文件：installer\_r24.1.2-windows.exe

安装：双击进行安装，找到exe的安装目录。安装完成后需要配置两个环境变量：

“我的电脑”右键菜单--->属性--->高级--->环境变量--->系统变量-->新建..

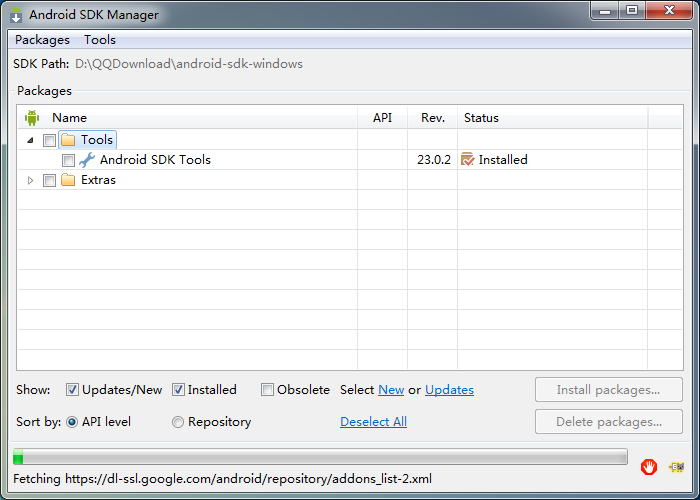
|  |
| --- |
| 变量名：ANDROID\_HOME  变量值：D:\android\android-sdk-windows |

找到path变量名—>“编辑”添加：

|  |
| --- |
| 变量名：PATH  变量值：;%ANDROID\_HOME%\platform-tools;%ANDROID\_HOME%\tools; |

**3.SDK Manager安装模拟器**

双击启动SDK Manager.exe 程序。



当然，我不能保证你SDK可以更新。所以，这里推荐一个网站：<http://www.androiddevtools.cn/>

它提供了国内的代理，以及各种开发工具的安装。（这个网站要从头开始看起，我直接跳到下载界面，刚开始没看到国内的郑州大学开源镜像站）

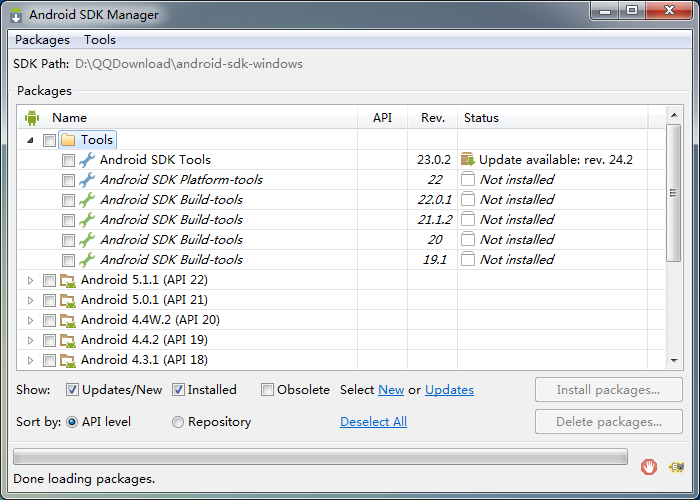
在Android SDK Manager 的菜单栏上点击“Tools”---->“Options...”设置相关代码，如下图：

*默认的不好用mirrors.opencas.cn端口80（可能是我没打开翻墙软件）*

好用的：HTTP Proxy Server：mirrors.zzu.edu.cn

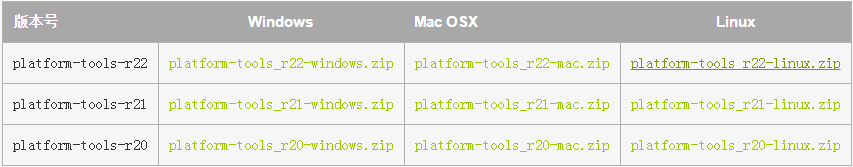
Http Proxy Port：80

然后点击关闭按钮。点击Packages下的reload(最后一个按钮)按钮重新更新镜像。

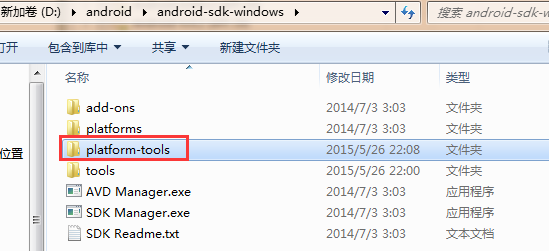


**安装SDK Platform-Tools**

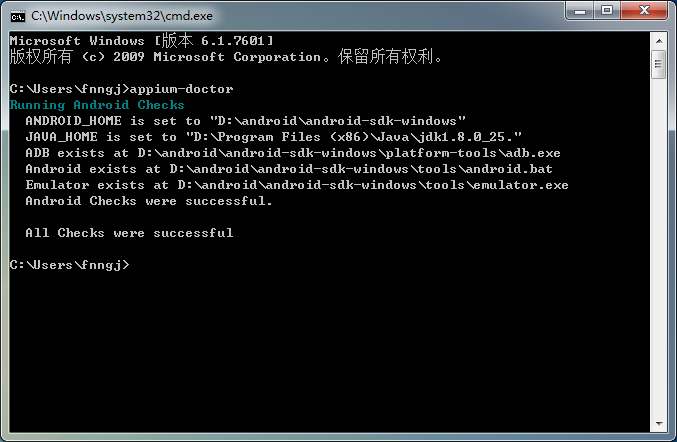
　　Android SDK Platform-tools 是版本有区别的工具文件夹，里面有adb 、aapt、fastboot 等工具包。在AndroidDevTools.cn 网站上找到Android SDK Platform-tools下载链接。



　　把解压出来的 platform-tools 文件夹放在 android sdk 根目录下，并把 adb所在的目录添加到系统 PATH 路径里(这个我们在设置android环境变量时已经添加)。



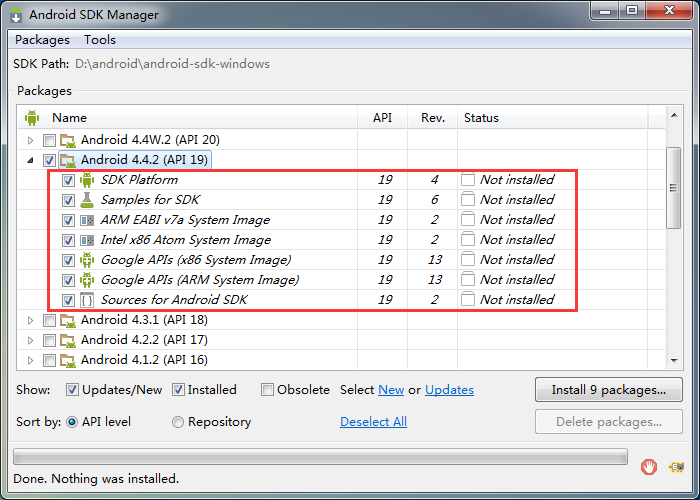
　　打开Windows命令提示符，通过“appium-doctor”命令检查appium环境。



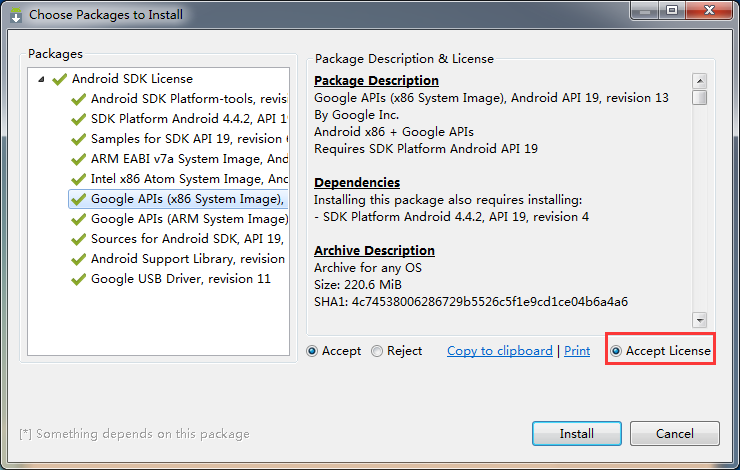
如果出现“All Checks were successful”的提示，表示成功

**第三节 安装Android模拟器或直接跳到第四节用真机**

图中以4.4.2版本为例，当然你可以自己选择版本，比如6.0。



如上图，勾选所需要安装的工具，点击右下角“Install x packages...”



 选择“Accept License”选项，点击“Install”按钮时行安装。

但是，你可能会发现这种方法会提示“Download interrupted: URL not found.”这样的错误，那么我们只能将这些工具单个的下载安装了。

如果不报错，安装下载一个android6.0有17个文件，最少两个小时下载（不过下载不完的可以下次打开接着下）。所以不想等这么久的可以单个文件下载配置安装。

好吧!再次感谢AndroidDevTools.cn网站的共享。以下下载链接均有其共享。

1. **安装SDK platform**

android 4.4.2 ：<http://pan.baidu.com/s/1eQf8ZgI>

 这是Android开发所需的sdk，下载并解压后，将解压出的整个文件夹复制或者移动到 .../android-sdk-windows/platforms/文件夹，然后重新打开SDK Manager.exe

1. **安装Samples for SDK**

android 4.4.2 ： <http://pan.baidu.com/s/1dDeSKt7>

这是Android SDK自带的示例代码，下载并解压后，将解压出的整个文件夹复制或者移动到 .../android-sdk-windows/samples文件夹下，然后重启SDK Manager.exe。

1. **安装SDK System images**

android 4.4.2 : <http://pan.baidu.com/s/1i3Jwhed>

这是在创建模拟器时需要的system image，也就是在创建模拟器时 CPU/ABI项需要选择的，下载并解压后，将解压出的整个文件夹复制或者移动到.../android-sdk-windows/system-images文件夹下即可， 如果没有 system-images目录就先创建此文件夹，然后重新打开SDK Manager.exe。

1. **GoogleMap APIs SDK**

android 4.4.2  （ARM）： <http://pan.baidu.com/s/1bno0mFt>

android 4.4.2  （x86）： <http://pan.baidu.com/s/1jGgKyZc>

这是GoogleMap APIs SDK，下载并解压后，将解压出的整个文件夹复制或者移动到 .../android-sdk-windows/add-ons文件夹下，然后打开SDK Manager

1. **Android Framework Source Code**

android 4.4.2 : <http://pan.baidu.com/s/1hqGGrVA>

这是Android Framework Source Code，下载并解压后，将解压出的整个文件夹复制或者移动到.../android-sdk-windows/sources文件夹下，然后重新打开SDK Manager.exe。

为了保险起见，以防以后用到而没有安装，所以这里全部做了安装。

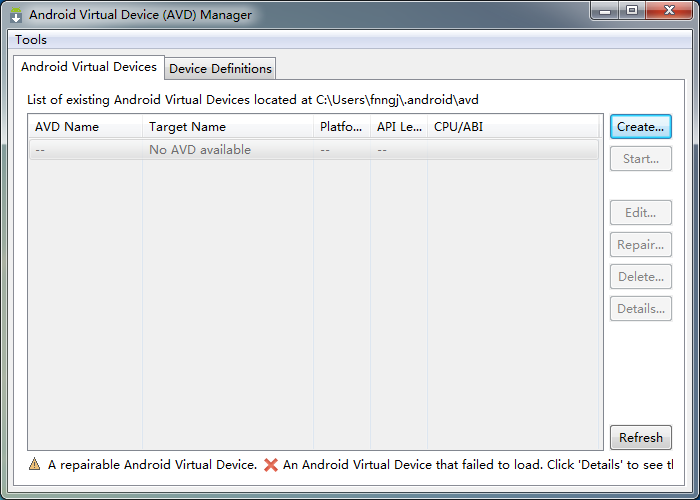
**第四节 创建android模拟器**

**CISC(复杂指令集计算机)和RISC(精简指令集计算机)是当前CPU的两种架构。**

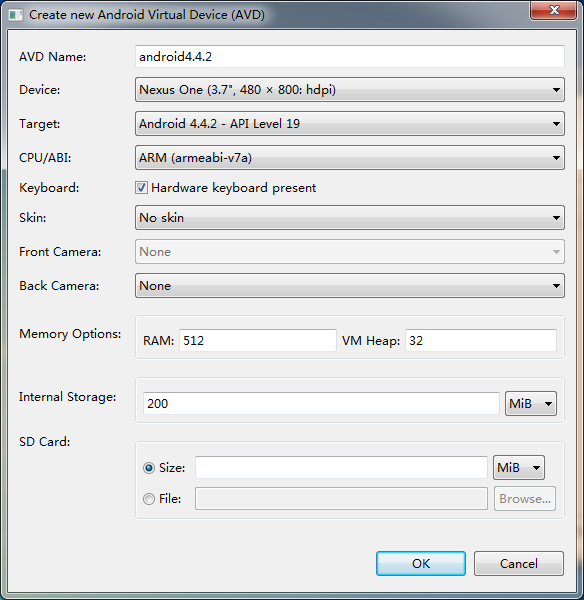
**x86架构采用CISC，而ARM采用RISC。 X86\_64就是支持64位系统。**

**1.创建一个最简单的ARM cpu的安卓模拟器**

双击“AVD Manager.exe”



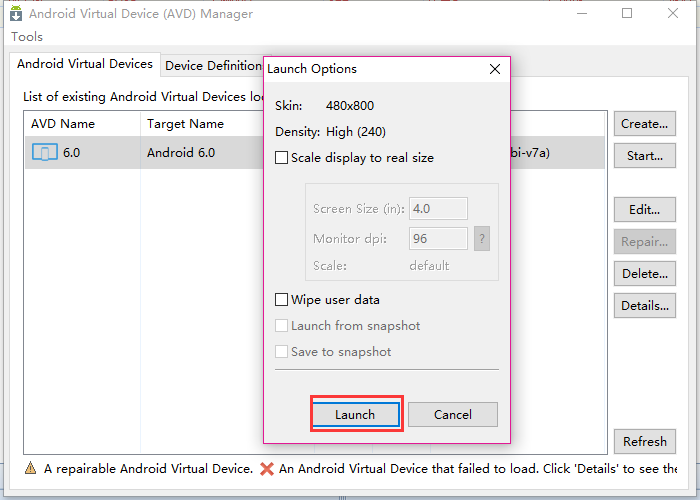
点击“Create....”



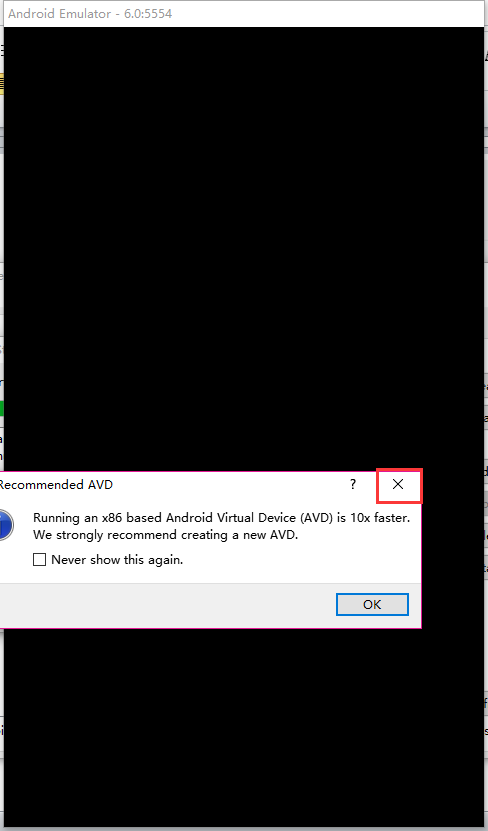
如果显示屏分辨率比较底的话，尽量选择低分辨率的“Device”。

RAM的值可手动改，最好不要超过768.

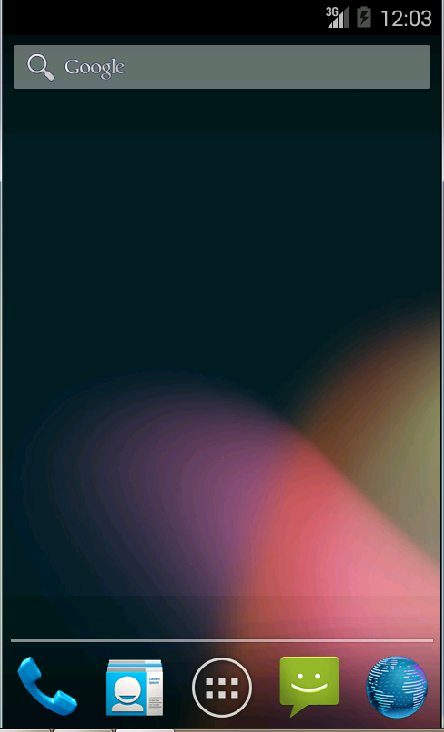
点击“OK”，在AVD  Manager 窗口，点击“Start...”按钮启动android 模拟器。



弹出提示不要点ok，要点击X（叉）因为它在建议你创建新的AVD。



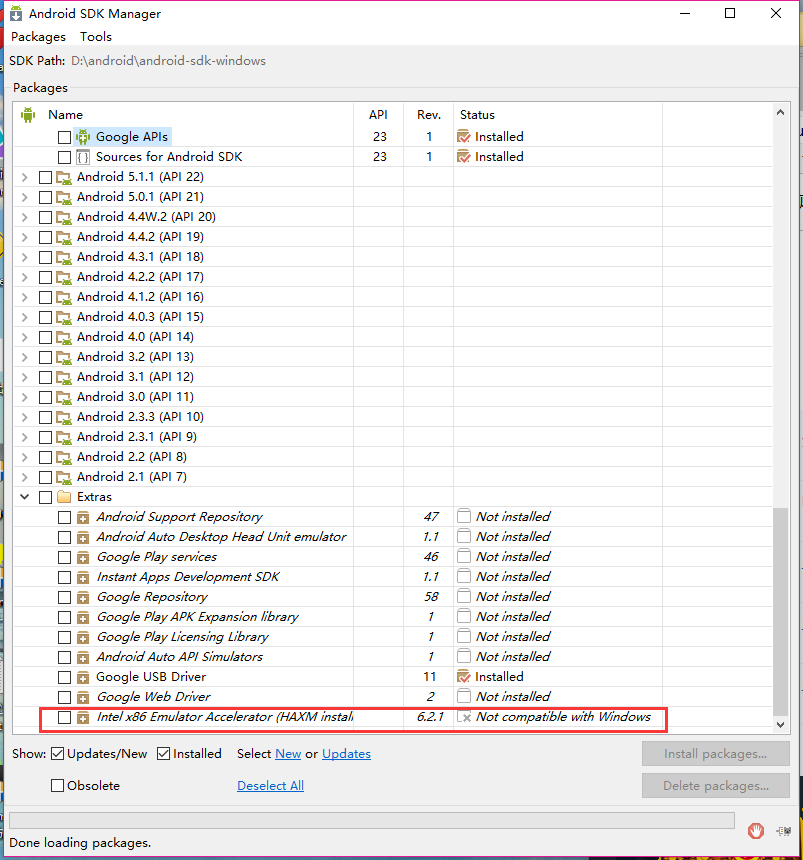
然后等2分钟就可以开机，看到首页了



**2.创建一个x86 CPU的模拟器**

X86开机和运行速度比ARM快，就是intel和奔腾的差别。

创建方法一样，就是会遇到一些坑，比如下图：



解决方法：[https://software.intel.com/sites/default/files/managed/c8/be/haxm-windows\_v6\_2\_1.zip](http://jump.bdimg.com/safecheck/index?url=rN3wPs8te/rykpBwn3Cdf18xPueseevRnX1sGp6obP1cLEoupjQ7hq5A8lOqBSzhhLh7wpMcMiicrLPK9kX8Zp4cdZitIIMHYoyaxHxI2n8lExPDX6mNuldHhilcinS/XmFHRJgjhfv9wn7wzRX+xIQ8yExvH7HOv+PIi4tFg4DKgqafaUsaK/S4GKVuMrgxtRD91eTgAUnkyTyTvh6HyYx0xWA1TaKrMDxm7iZ2BjQ=" \t "_blank)  
请下载最新的HAXM，手动安装。放在（没目录就新建）sdk\extras\intel\Hardware\_Accelerated\_Execution\_Manager目录下面

中间可能会发生错误，主要是BIOS中有个“Intel Virtualization Technology”的选项（vt-x）没开启，开启后就能正常安装，查看是否开启：右键任务栏---任务管理器---性能---cpu，虚拟化已启用）

最后我装完，图标没变，但是可以运行x86的AVD。开机都快了。

那个内存推荐768M，现在的机子内存都4个G了，是不是其他模拟器比如逍遥模拟器或是genymotion模拟器会比较大内存呢？

因为新虚拟机没了实体键，所以我们可以利用键盘按键来操作android虚拟机。

|  |  |
| --- | --- |
| 模拟器按键 | 键盘按键 |
| 后退 | ESC |
| 菜单 | F1或Page Up |
| 开始 | F2或Page Down |
| 呼叫 | F3 |
| 挂断 | F4 |
| 电源按钮 | F7 |
| 禁止/启用所有网络 | F8 |
| 开始跟踪 | F9 |
| 停止跟踪 | F10 |
| 旋转屏幕（横/竖屏切换） | Ctrl+F11 |
| 主页 | HOME |
| 方向键 左/上/右/下 | 小键盘 4/8/6/2 |
| 方向键 中心键 | 小键盘 5 |
| 调低音量 | 小键盘 负号(-) |
| 调高音量 | 小键盘 加号(+) |
|  |  |

**第五节 安卓模拟器运行程序**

1. 安装 python-client

其实，[python-client](https://github.com/appium/python-client) 的项目名称叫：Appium-Python-Client。

Java，我猜测是Appium-Java-Client

https://github.com/appium/java-client

推荐pip安装：在cmd.exe命令提示符窗口输入

pip install Appium-Python-Client

最后看到下面的英文表示成功

Successfully installed Appium-Python-Client-0.24

1. 运行第一个Appium测试

* **第一步**，启动Android模拟器。
* **第二步**，启动 Appium Server。

点击右上角 **三角** 按钮，注意Appium的启动日志。

{"address":"127.0.0.1","logNoColors":true,"platformName":"Android","platformVersion":"23","automationName":"Appium"}

> info: Console LogLevel: debug

Appium在启动时默认占用本机的4723端口，即：127.0.0.1:4723

**第三步**，编写 appnium 测试脚本。

#coding=utf-8

from appium import webdriver

desired\_caps = {}

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

desired\_caps['platformVersion'] = '6.0'

desired\_caps['deviceName'] = 'Android Emulator'

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

desired\_caps['appActivity'] = '.Calculator'#原生的就加一个点

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

driver.find\_element\_by\_name("1").click()

driver.find\_element\_by\_name("5").click()

driver.find\_element\_by\_name("delete").click()

driver.find\_element\_by\_name("5").click()

driver.find\_element\_by\_name("+").click()

driver.find\_element\_by\_name("6").click()

driver.find\_element\_by\_name("=").click()

driver.quit()

运行上面的脚本，你将会看到 Android 模拟器如下运行界面：计算器结果21

因为我自己用pycharm手敲没有复制，所以敲错了一些代码，后来修改了，运行成功。

**第六节 Appium连接真机**

1. 用数据线连接手机和电脑。连接选项共享网络就可以了。
2. 打开电脑的命令提示符cmd.exe，输入adb devices，如果没有出现UDID，就输入adb kill-server，然后重新输入adb devices。关掉其它软件多试几次可以的。下面这个错误就是由于电脑打开了酷狗音乐报的错。

error: cannot parse version string: kg01

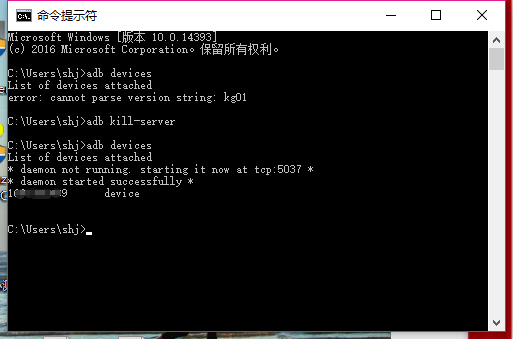
1. 重启adb服务adb start-server 重启手机adb reboot

进入shell模式adb shell 挂载adb remount

从电脑发文件到手机adb push <本地路径> <远程路径>

从手机下载文件到本地adb pull <远程路径> <本地路径>

输出日志第一种：输出到手机存储卡adb logcat > /sdcard/mylogcat.txt第二种：输出到电脑上adb logcat > D:/Temp/1.txt（1.txt必须在电脑上存在，才能写入logcat内容）



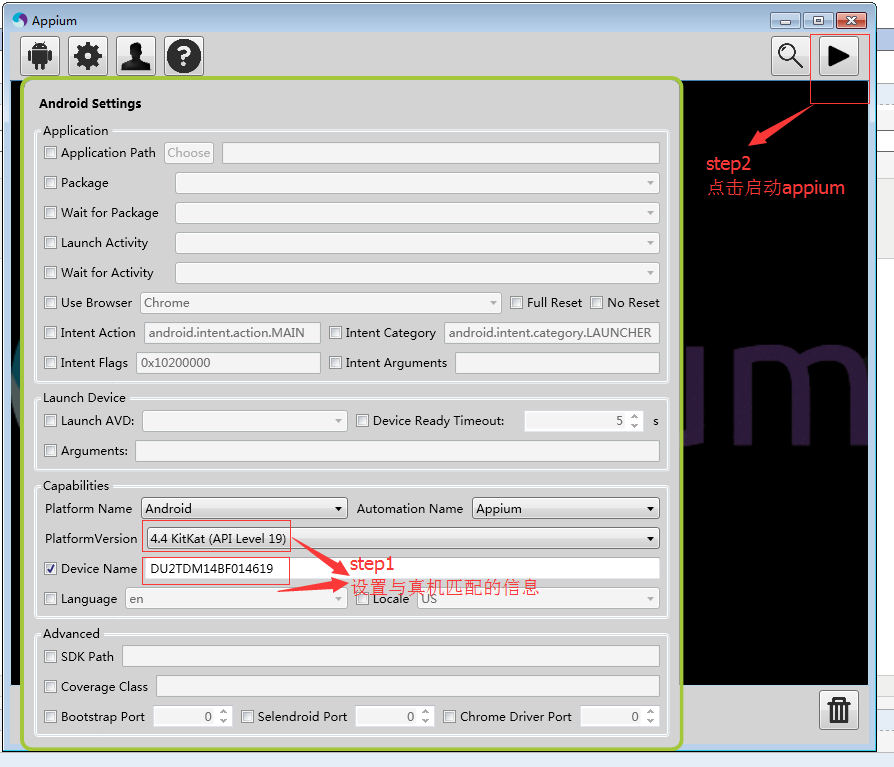
1. 然后打开Appium。

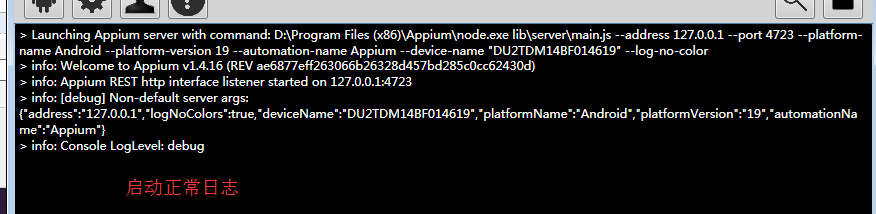
方法一：在doc命令提示符窗口输入

appium -a 127.0.0.1 -p 4723  –U  查到的UDID号 --no-reset

最后一行（共四个info）显示debug就对了。

1. 方法二（更推荐）：appium界面启动





1. 测试脚本的编写及运行

真机运行的测试脚本与模拟器上几乎是相同的，就是有些儿配置不太一样。具体脚本demo如下demo.py，实测通过，可以打开手机上的淘宝或支付宝首页。



*#启动配置环境*desired\_caps = {}  
*#设备系统名称:Android或iOS*desired\_caps[**'platformName'**]=**'Android'***#设备系统版本*desired\_caps[**'platformVersion'**]=**'6.0'***#设备名称(UUID号或设备名称比如OPPO R11s)*desired\_caps[**'deviceName'**]=**'UUDI号或设备名称都可以的，已实测'**

*#知道appPackage和appActivity的方法：手机只打开这个待测App  
#打开cmd命令提示符中输入：adb shell "dumpsys window w|grep \/|grep name=|sed 's/mSurface=Surface(name=//g'|sed 's/)//g'|sed 's/ //g'"**#app包名（有加锁的先自己去掉）*desired\_caps[**'appPackage'**]=**'com.taobao.taobao'**desired\_caps[**'appActivity'**]=**'com.taobao.tao.homepage.MainActivity3'***#app在电脑上的写法  
# desired\_caps['app'] = PATH('C:\\Users\\360\\Desktop\\taobao.apk')  
  
#启动app*driver=webdriver.Remote(**'http://localhost:4723/wd/hub'**,desired\_caps)  
  
*#启动app时，需要一定时间进入引导页，所以必须设置等待时间，不然下面会一直报错定位不到元素*time.sleep(6)  
*# driver.find\_element\_by\_id(':id/skip').click()*driver.quit()

注：在测试用例运行过程中，正常情况下手机会自动安装Appium Settings和Unlock，有些手机有兼容性问题，Appium Settings和Unlock程序打不开，所以每次运行脚本都会提示安装，按提示允许安装就可以。

Unlock绕过了我的指纹锁打开了App首页(>\_<)

Appium的Desired Capabilities是扩展了webdriver的Desired Capabilities的，下面的一些通用配置是需要指定的：

* automationName：使用哪种自动化引擎。appium（默认）还是Selendroid？
* platformName：使用哪种移动平台。iOS, Android, orFirefoxOS？
* deviceName：启动哪种设备，是真机还是模拟器？iPhone Simulator, iPad Simulator, iPhone Retina 4-inch, Android Emulator, Galaxy S4, etc...
* app：应用的绝对路径，注意一定是绝对路径。如果指定了appPackage和appActivity的话，这个属性是可以不设置的。另外这个属性和browserName属性是冲突的。
* browserName：移动浏览器的名称。比如Safari' for iOS and 'Chrome', 'Chromium', or 'Browser' for Android；与app属性互斥。
* udid：物理机的id。比如1ae203187fc012g。

**下面这些属性是android平台特定的：**

* appActivity：待测试的app的Activity名字。比如MainActivity, .Settings。注意，原生app的话要在activity前加个"."。
* appPackage：待测试的app的java package。比如com.example.android.myApp, com.android.settings。

**获取Android app的appActivity**

网上有写很多方法https://testerhome.com/topics/1030，评论区还有补充

笔者（这里指别人）用的是把apk后缀改为zip格式，打开压缩包后再打开AndroidManifest.xml文件。找到manifest和activity地方，就是包名跟activity名。

**第七节 定位App页面的元素**

1. aapt：在D:\androidsdk\android-sdk-windows\build-tools\24.0.0目录下找到aapt.exe，将这个路径设置环境变量，添加到path。打开cmd。输入aapt
2. 启动定位元素的工具：打开目录D:\androidsdk\android-sdk-windows\tools，双击启动uiautomatorviewer.bat
3. 定位App元素1.cmd打开输入adb devices,确认手机已连上2.打开手机淘宝页面，让屏幕处于点亮状态3.点左上角安卓机器人按钮Devices Screenshot按钮刷新页面

4.移动鼠标到需要定位的元素上，比如搜索输入框，右下角可以看到元素对应的属性

