#### Appium简介

Appium是开源的、跨平台的测试自动化测试框架；

主要用于测试移动端应用；

Appium支持模拟器和真机上的原生应用、混合应用、Web应用；

Appium使用WebDriver的JSON wire协议来驱动Apple系统的UI Automator库、Android系统的UI Automator框架；

因Appium集成了Selendroid框架，所以Appium支持比较老版本的Android；

Appium支持Selenium WebDriver支持的所有语言（Java、Python、JavaScript、Ruby、C#、Rust、php等）。

（国内教程基本上全是Python）

官网：<https://appium.io/>

###### Appium原理

**Android端过程**

客户端client，指咱们的测试脚本；

Appium在服务端启动一个Server（4723端口），用来接口WebDriver client标准的rest请求，调用对应框架响应操作；

Appium Server把请求转发给中间件Bootstrap.jar（手机上），Bootstrap.jar监听4724端口并接收Appium命令；

Bootstrap将执行结果返回给Appium Server；

Appium Server再将结果返回给Appium Client。

**iOS端过程**

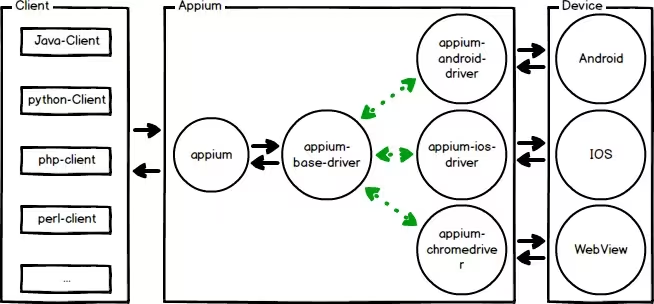
客户端client，指咱们的测试脚本；

Appium在服务端启动一个Server（4723端口），用来接口WebDriver client标准的rest请求，调用对应框架响应操作；

Appium Server调用instruments.js，启动socket server，分出子进程instruments.app，将bootstrap.js注入device；

bootstrap.js将执行结果返回给Appium Server；

Appium Server再将结果返回给Appium Client。



UI Automator测试框架是Android SDK自带的APP UI自动化测试Java库；

而UI Automator对H5支持有限；

Appium 引入了chromedriver等来实现基于H5的自动化；

Android和iOS启动过程的区别在Appium将请求转发给了Bootstrap.jar或bootstrap.js,再由Bootstrap驱动UI Automator或UI Automation。

#### 环境搭建

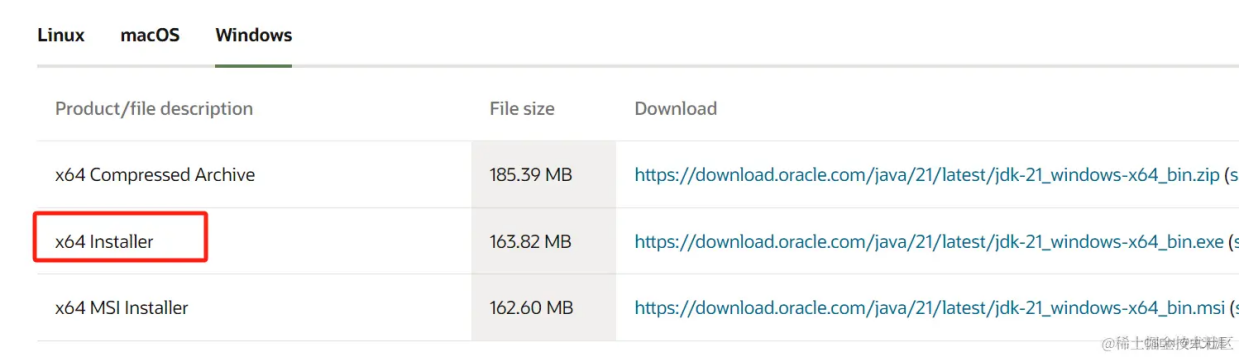
参考：

<https://www.cnblogs.com/jiachangwei/p/12143687.html>

所需的安装包已全部放到 \\192.168.1.2\程序共享\appium

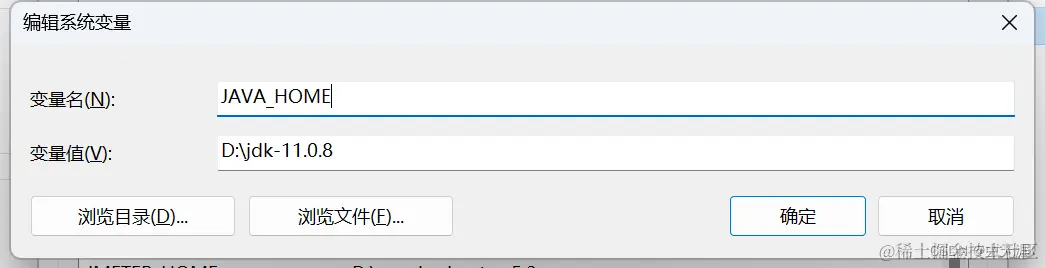
**1、安装JDK，及配置环境变量**

下载地址：<https://www.oracle.com/java/technologies/downloads/>



按照提示，可以一步步安装

新建系统环境变量JAVA\_HOME：



找到名为Path的变量名，点击编辑

新建以下三个环境变量，可以直接粘贴

%JAVA\_HOME%\bin

%JAVA\_HOME%\lib\tools.jar

%JAVA\_HOME%\jre\bin

**2、安装SDK，及配置环境变量**

下载sdk tools 地址 <https://www.androiddevtools.cn/>

建议直接去共享文件加下下载 ,所需的包已都下载。

[\\192.168.1.2\程序共享\appium\sdk.zip](\\\\192.168.1.2\\程序共享\\appium\\sdk.zip)

解压后配置环境变量

新增环境变量ANDROID\_HOME

变量值：你解压后的sdk 文件夹路径

找到名为Path的变量名，点击编辑

添加环境变量，可以直接复制

%ANDROID\_HOME%\tools

%ANDROID\_HOME%\platform-tools

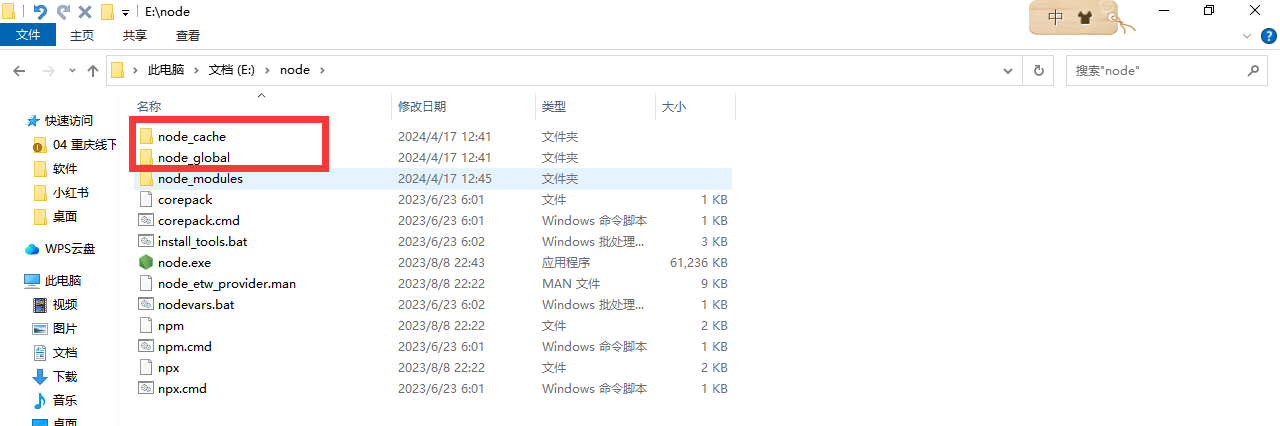
完成后点击确定，返回上层菜单，再点确定，保存

在终端输入 adb version 查看是否安装成功

**3、安装Node.js**

下载地址： [https://nodejs.org/en/download/](http://testingpai.com/forward?goto=https://nodejs.org/en/download/" \o "https://nodejs.org/en/download/)

在nodejs的安装路径下，新建两个空的文件夹，并分别按照下图命名



在cmd命令行执行下面两句指令：

npm config set prefix "nodejs安装路径\node\_global"  
 npm config set cache "nodejs安装路径\node\_cache"

(上面两个命令去掉引号执行)

配置nodejs环境变量：

在系统变量中新建NODE\_PATH变量，变量值：

Nodejs安装路径\node\_global\node\_modules

在系统变量Path增加值：%NODE\_PATH%

将用户变量Path中自带的C:\Users\用户名\AppData\Roaming\npm修改为

Nodejs安装路径\node\_global

**4、安装appium**

配置npm源

npm config set registry <https://registry.npmmirror.com>

安装appium

npm install -g appium

下载驱动

appium driver install uiautomator2

安装客户端库 我这里使用的是node.js 需要选择一个与 Appium 兼容的客户端库。

维护得最好的库和 Appium 团队推荐使用的库是 WebdriverIO。

通过npm安装

npm i --save-dev webdriverio

#### 运行appium 测试

###### 启动appium

打开终端 执行appium 命令

###### 链接安卓设备

以雷电模拟器作为安卓设备进行测试；

打开雷电模拟器

在终端执行 adb devices 命令就可以看得到我们已连接到的安卓设备



如何连接真机 自行搜索。

###### 运行代码

一下为官方给出的demo 自动找到设置中的电池选项

const {remote} = require('webdriverio');

const capabilities = {

  platformName: 'Android',

  'appium:automationName': 'UiAutomator2',

  'appium:deviceName': 'Android',

  'appium:appPackage': 'com.android.settings',

  'appium:appActivity': '.Settings',

};

const wdOpts = {

  hostname: process.env.APPIUM\_HOST || 'localhost',

  port: parseInt(process.env.APPIUM\_PORT, 10) || 4723,

  logLevel: 'info',

  capabilities,

};

async function runTest() {

  const driver = await remote(wdOpts);

  try {

    const batteryItem = await driver.$('//\*[@text="Battery"]');

    await batteryItem.click();

  } finally {

    await driver.pause(1000);

    await driver.deleteSession();

  }

}

runTest().catch(console.error);

## 附件：

Appium 开启自动下载驱动模式

appium server --allow-insecure chromedriver\_autodownload

此模式可以自动下载我们需要的驱动。

查看chromedriver 版本：

adb shell am start -a android.intent.action.VIEW -d <https://liulanmi.com/labs/core.htm>

如上面方法查不到

通过手机设置查看

路径：设置 → 应用管理 → Android System WebView (找不到搜索)

驱动下载地址：

<https://chromedriver.chromium.org/downloads>

###### adb 常用命令

查看当前连接设备

adb devices

链接设备

adb connect ip:sport

安装apk文件

adb install xxx.apk

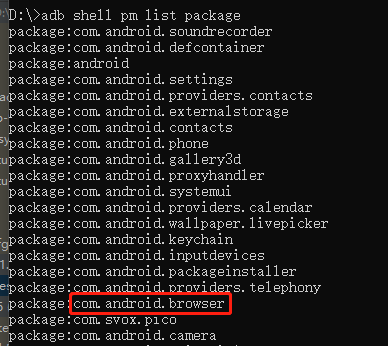
覆盖安装

adb install -r -t xxx.apk

###### 如何获取应用 appPackage & appActivity

1. 查看 appPackage

adb中用adb shell pm list package命令



2、查看appActivity

在adb中，用adb logcat -c 清除日志

然后用adb logcat ActivityManager:I \*:s

然后在手机或者模拟器中打开你要查看的laucherActivity的软件

打开后就会显示该软件的laucherActivity名称

com.android.browser.BrowserActivity(脚本中需要去掉截图里面的斜杆)

