**环境搭建**

创建日期 星期一 30 四月 2018

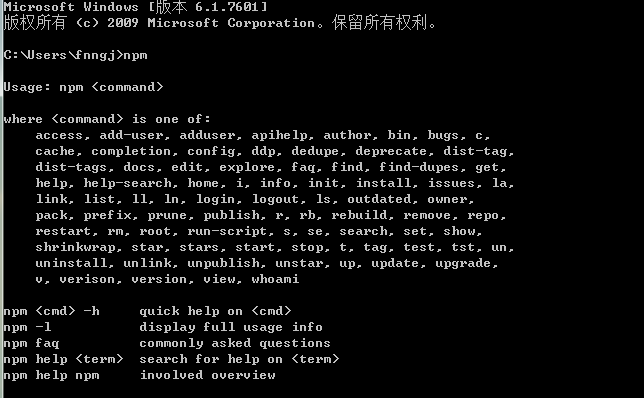
1 : 安装nodejs

Appium官方网站：[http://appium.io/](http://appium.io/" \o "http://appium.io/)  
Easy setup process, run a test now.  
> brew install node # get node.js  
> npm install -g appium # get appium  
> npm install wd # get appium client  
> appium & # start appium  
> node your-appium-test.js

官方首页给出了appium的安装步骤。

我们需要先安装node.js 。node.js官方网站：[https://nodejs.org/{{~\Documents\GitHub\JavaEE\MyWiKi\Pic\Appium\nodejs.png](https://nodejs.org/%7b%7b~/Documents/GitHub/JavaEE/MyWiKi/Pic/Appium/nodejs.png" \o "https://nodejs.org/{{~\\Documents\\GitHub\\JavaEE\\MyWiKi\\Pic\\Appium\\nodejs.png)}}  
根据你的操作系统选择相应的版本进行下载

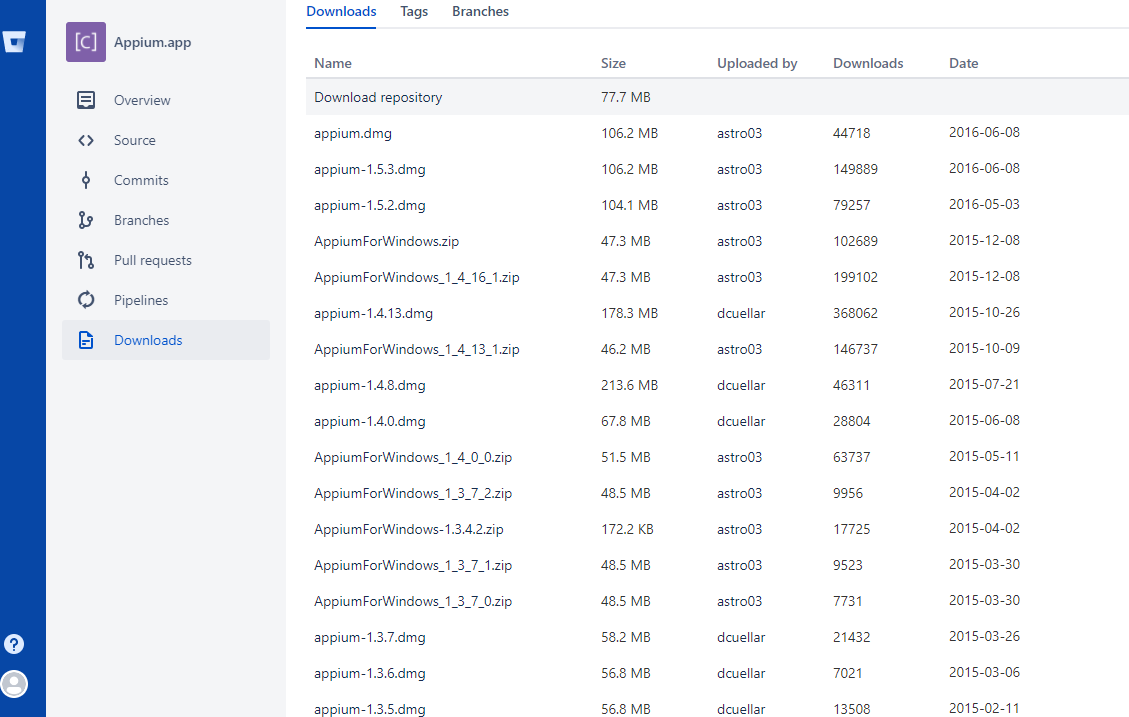
安装完成，打开Windows 命令提示符，敲入“npm”命令回车。



2 : 安装appium

npm安装完成 可以使用npm install -g appium自动化获取appium 但是由于伟大的墙，可能会失败或缺少组件

可以绕过这个直接在官网下载：<https://bitbucket.org/appium/appium.app/downloads/>



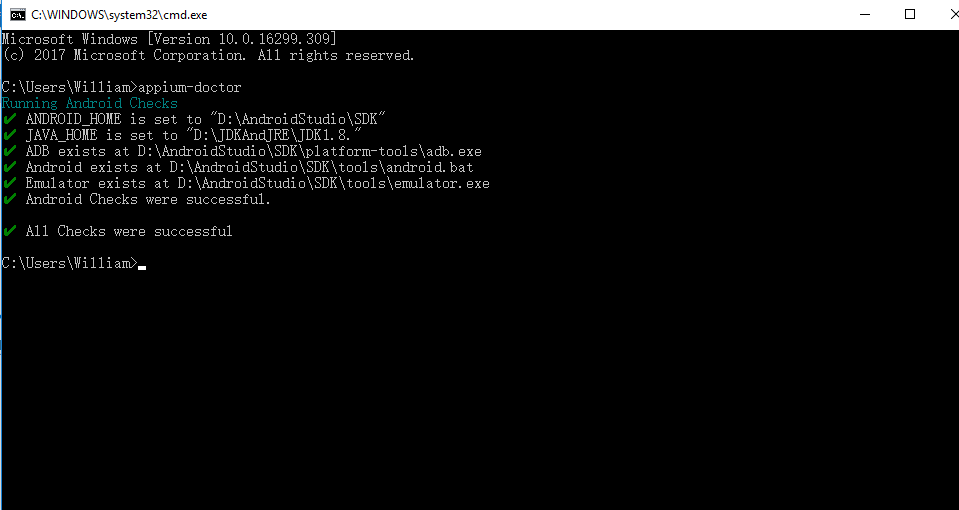
选择对应的版本

启动界面



3 ： android 环境

jdk环境 lue  
设置环境变量：ANDROID\_HOME ---> androidSDK目录  
将sdk目录下的platform-tools 和 tools加入path环境变量

最后运行appium-doctor检查  


4：安装Appium Client

Appium Client是对webdriver原生api的一些扩展和封装。它可以帮助我们更容易的写出用例，写出更好懂的用例。

　　Appium Client是配合原生的webdriver来使用的，因此二者必须配合使用缺一不可。

　　在本文档的开始已经介绍Appium支持多种编程语言编写自动化测试脚本，由于我个人熟悉Python语言，所以选择Python语言来编写自动化测试脚本。

　　访问Python官方网站：[https://www.Python.org/](https://www.Python.org/" \o "https://www.Python.org/)

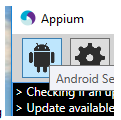
　　下载最新版本的Python2，截止作者发稿，最新版本为Python2.7.10版本。读者根据自己的平台选择相应的版本进行下载；对于Windows用户来说，如果你的系统是32位的请选择x86版本，如果是64位系统请选择64版本进行下载。下载完成会得到一个以.msi为后缀名的文件，双击进行安装。

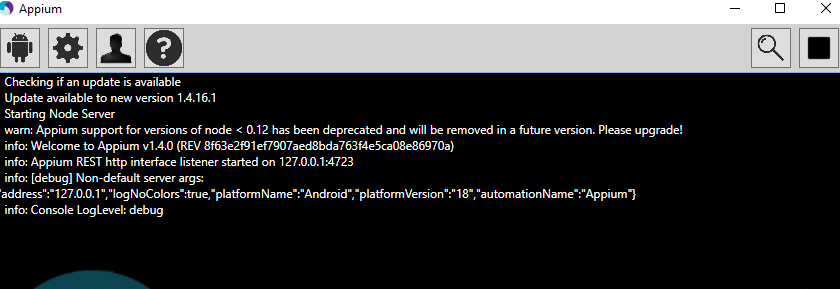
安装完毕后会获得pip自动化管理工具：使用 pip install Appium-Python-Client获得Python包

**自动化脚本(Python)**

创建日期 星期一 30 四月 2018

启动android虚拟机  
启动appium

点击android setting   
点击右边的三角启动



测试脚本

#coding=utf-8  
from appium import webdriver

desired\_caps = {}  
desired\_caps['platformName'] = 'Android'  
desired\_caps['platformVersion'] = '4.3'  
desired\_caps['deviceName'] = 'Android Emulator'  
desired\_caps['appPackage'] = 'com.android.calculator2'  
desired\_caps['appActivity'] = '.Calculator'

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

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

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

driver.find\_element\_by\_name("7").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("1").click()  
driver.find\_element\_by\_name("1").click()  
driver.find\_element\_by\_name("7").click()

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

driver.quit()

测试脚本的api和selenium非常相似

这是python的脚本测试，较为方便

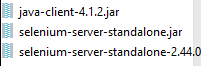
**自动化脚本（android studio）**

创建日期 星期一 30 四月 2018

添加module

选择 java library

导入jar包



启动虚拟机

编写脚本

package com.example;

import org.junit.\*;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.remote.DesiredCapabilities;

import java.io.File;  
import java.net.URL;  
import java.util.List;

import io.appium.java\_client.\*;  
import io.appium.java\_client.android.\*;

/\*\*  
\* Created by William on 2018/4/30.  
\*/

public class AndroidContactsTest {

//Driver  
private AppiumDriver<AndroidElement> driver;

/\*\*  
\*/  
@Before  
public void setUp() throws Exception {

// D:\WorkSpace\AutoTest\DemoTest\src\main\java\apps\ContactManager.apk

File classpathRoot = new File(System.getProperty("user.dir"));  
File appDir = new File(classpathRoot, "/DemoTest/src/main/java/apps/");  
File app = new File(appDir, "ContactManager.apk");  
System.out.print(app.getAbsolutePath());  
DesiredCapabilities capabilities = new DesiredCapabilities();  
capabilities.setCapability("deviceName","Nexus\_5X\_API\_18");  
capabilities.setCapability("platformVersion", "4.3");  
capabilities.setCapability("app", app.getAbsolutePath());  
capabilities.setCapability("appPackage", "com.example.android.contactmanager");  
capabilities.setCapability("appActivity", ".ContactManager");  
driver = new AndroidDriver<>(new URL("<http://127.0.0.1:4723/wd/hub>"), capabilities);

}

@After  
public void tearDown() throws Exception {

driver.quit();

}

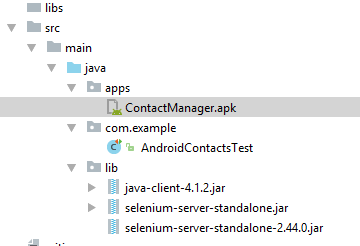
/\*\*  
\*/  
@Test  
public void addContact(){

WebElement el = driver.findElement(By.name("Add Contact"));  
el.click();  
List<AndroidElement> textFieldsList = driver.findElementsByClassName("android.widget.EditText");  
textFieldsList.get(0).sendKeys("Some Name");  
textFieldsList.get(2).sendKeys("[Some@example.com](mailto:Some@example.com)");  
driver.swipe(100, 500, 100, 100, 2);  
driver.findElementByXPath(".//\*[@text='Save']").click();

}

}

将测试的app放入目录下



启动测试