RobotFramework的AppiumLibrary在Android平台自动化示例：

Appium原理介绍：

核心是一个通过JSON交互的REST API服务，有Client端和Server端。 代码运行时，Client端通过JSON协议将代码命令发给Server，Server端再翻译成App能识别的命令执行，然后App再将执行结果返回Server端，Server端再告知Client端显示结果。就是这样一个请求本地服务的交互过程。

运行环境准备：

**1）安装appiumlibrary ：**pip install robotframework-appiumlibrary。

**2）下载Appium的GUI工具：**

官网下载appium的windows安装包：

官网：<https://bitbucket.org/appium/appium.app/downloads/>

下载AppiumForWindows\_1\_4\_16\_1.zip。

**3）安装Appium：**

解压AppiumForWindows\_1\_4\_16\_1.zip，双击appium-installer.exe开始一步步安装。

**4）安装安卓开发者工具（Android SDK）：**

从网上下载，或者从别人出拷贝。

参考<http://jingyan.baidu.com/article/bea41d437a41b6b4c51be6c1.html>教程，从<http://www.androiddevtools.cn/>下载安装包，然后按照教程安装配置。

**5）准备一个beta版的安装包。**

编写脚本：

1. **配置Open Application关键字，启动App：**

先查看AppiumLibrary的Open Application关键字的使用说明。通过查看示例，能发现配置Open Application方法，该方法需要传入以下几个参数：

必填的：

remote\_url：appium启动url(本地服务、端口=4723、path=/wd/hub)，固定不变。

platformName：Android或iOS。

platformVersion：系统版本，我的是Android 4.4.2。

deviceName：设备名称，通过adb devices查看。

app：安装包的本地存放路径，如${CURDIR}/appfile/xxx.apk。

appPackage和appActivity：包名和启动activate名，可以通过`adb logcat | grep “Displayed”`查看。

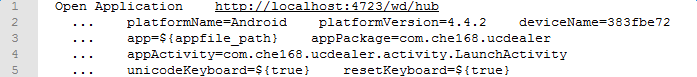
（用app参数时，可不填appPackage和appActivity参数，前者适用于未安装APK的时候，后者适用于已安装apk的时候。）

选填的：

unicodeKeyboard=true：启动unicode键盘，开启后支持中文。

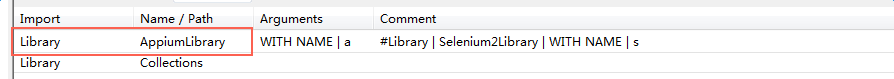
resetKeyboard=true：重置键盘，开启后支持中文。

填写后结果如图：



1. **导入AppiumLibrary：**

同RobotFramework运行Selenium2Library，导入AppiumLibrary即可使用Appium的关键字。



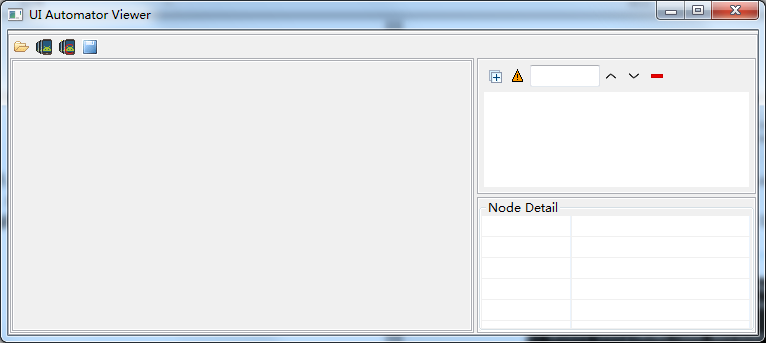
官方文档url：http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html

1. **定位App原生控件元素：**

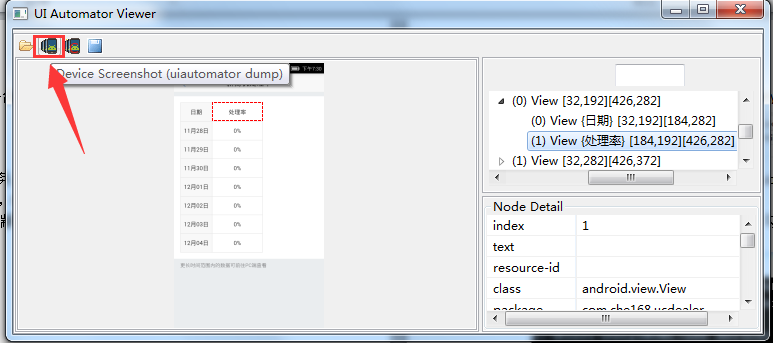
连接电脑与手机，打开”开发者选项”，勾选USB调试。然后cmd输入adb devices，若有设备显示出来，则证明连接成功。

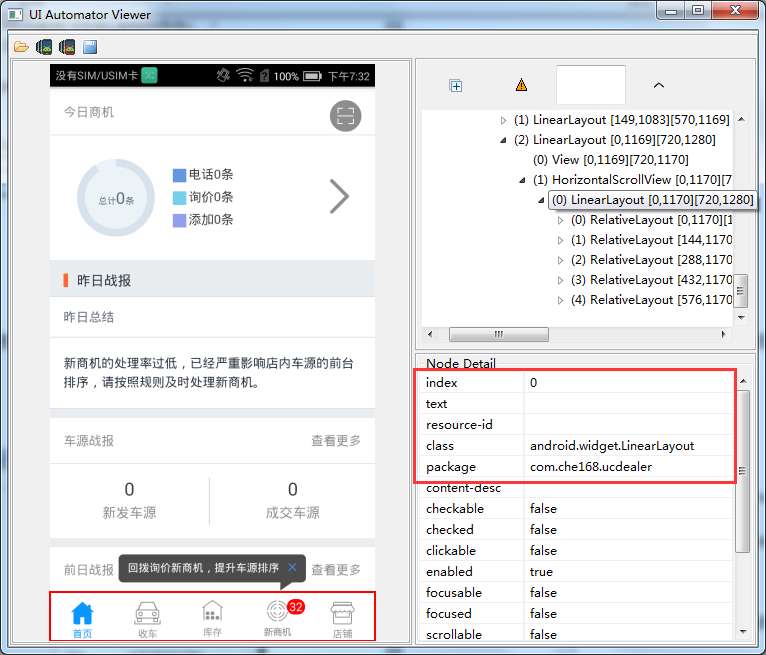


然后进入/Android SDK/tools/，双击打开uiautomatorviewer.bat工具。



左上角点击第2个按钮，会获取手机当前页面元素：





之后就可以根据index、text、id（就是resource-id）、xpath（就是class）定位元素。

1. **定位App内H5页面元素：**

确认自己安装的是beta版本，因为正式版本无权限、无法查看H5元素。然后参考<http://ask.dcloud.net.cn/article/69>教程（后半部位置），pc使用Chrome浏览器连接查看Android里H5页面元素。

（无法查看H5页面元素原因：<http://blog.csdn.net/u012263493/article/details/50347699> ）

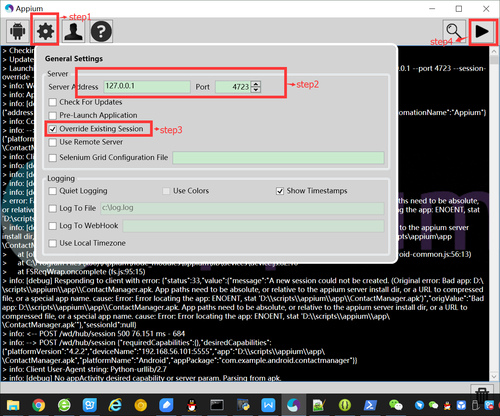
（若chrome调试打开空白页面，请复制下面url页面内所有内容，配置host后重新调试。

https://raw.githubusercontent.com/racaljk/hosts/master/hosts）

运行脚本：

1. **配置Appium并启动服务：**

打开appium，如图所示勾选"Override Existing Sessions"选项，然后点击Launch打开Appium Socket Server 。



"Override Existing Sessions"选项，可允许appium覆盖使用已存在的appium服务，防止连续运行引起appium报错。GUI工具里，在"General Settings"菜单里勾选该项，命令行启动服务，直接运行"appium --session-override"即可。

1. **运行脚本：**

同RobotFramework运行Selenium2Library，命令行输入pybot xxxx.txt即可运行。

常用命令：

点击Click Element；

输入[Input Text](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html" \l "Input%20Text" \o "Link to this keyword)；

滑动页面Swipe；

等待加载完成[Wait Until Page Contains](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html" \l "Wait%20Until%20Page%20Contains" \o "Link to this keyword)、[Wait Until Page Contains Element](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html" \l "Wait%20Until%20Page%20Contains%20Element" \o "Link to this keyword)[、Until Page Does Not Contain](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html" \l "Wait%20Until%20Page%20Does%20Not%20Contain" \o "Link to this keyword)、[Wait Until Page Does Not Contain Element](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html" \l "Wait%20Until%20Page%20Does%20Not%20Contain%20Element" \o "Link to this keyword)；

统计页面元素数量[Get Matching Xpath Count](http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html#Get%20Matching%20Xpath%20Count)（这个讲个应用，与Run KeyWord If结合，可写”如果有xxx则xxx”的自定义关键字。例如后面示例里的” If Has Advert Then CLose”关键字。）

获取操作环境Get Contexts、Get Current Context（就是获取操作焦点）；

切换操作环境Switch To Context（与上面方法结合，可在App操作和H5操作之间切换）；

放大与缩小Zoom和Pinch。

大概就这些，其他请上官网文档自己查阅：

http://serhatbolsu.github.io/robotframework-appiumlibrary/AppiumLibrary.html

示例脚本（车智赢app）：

步骤：登录 - 首页点击”前日战报”的查看更多 - 点击查看跟进率详情



目前遗留问题：

操作2个H5页面后，Appium会获取重复WebView，暂时无法解决。

发现过程：

Case里包含两个h5页面，首页和详情页，期间还有其他原生页面，我在首页第一次switch to context，能拿到首页元素，但之后流程进去详情页后，再switch to context却怎么都定位不上，xpath肯定没错，用chrome调试确认过了。之后在详情页用Get Source打印了源码，发现还是首页的代码。才发现这个会获取同一WebView的问题。

查到java版解决方法，但没想好怎么用RobotFramework实现。参考网址：https://testerhome.com/topics/4166