# WebView的自动化

## 准备工作：修改应用源代码，添加webview调用

**protected** **void** onCreate(**Bundle** savedInstanceState) {

super.onCreate(savedInstanceState);

**WebView** myWebView = (**WebView**) findViewById(**R**.id.jcywebview);

**myWebView.setWebContentsDebuggingEnabled(true);**

};

每个webview加入黄色部分代码，通常一个app里只有一个webview，添加到一处即可

## 查看webview中的网页内容

### 第一种情况：被测试应用webview不依赖app

方法：直接用chrome浏览器F12里面的手机模式打开对应的网页。

### 第二种情况：被测应用webview与app原生有交互，依赖app

方法：通过chrome浏览器的远程调试功能（前提是本电脑可以科学上网）

详细步骤：

确保被测试app在手机上打开

打开chrome浏览器，地址栏输入chrome://inspect

等待界面显示如下



点击 inspect即可。

注意：有的系统比较老的，比如android 4.4 对应的webview的版本比较老 ，比如33.0 的，可能不能inspect， 所以可以使用新一些的android 版本。

弹出的界面里面，开始没有手机界面， 可以点击右上角

## 自动化webview

### 通过应用自动化webview网页内容

**查看app context:**

IN: driver.contexts

OUT:['NATIVE\_APP', 'WEBVIEW\_com.example.jcy.wvtest']

本地context一般名为'NATIVE\_APP'

WebView context一般名为'WEBVIEW\_XXXXX' (xxxx指的是程序的包名，或者开发会直接告诉你这个webview的名称)

**查看app当前所在context**

IN: driver.current\_context

OUT: NATIVE\_APP #这个时候还不能操作webview

**切换到WebView context**

IN: driver.switch\_to.context(**'WEBVIEW\_com.example.jcy.wvtest'**)

IN: driver.current\_context

OUT: WEBVIEW\_com.example.jcy.wvtest #这个时候可以操作webview了

切换到webview之后就可以像selenium操作web网页自动化那样操作了

在app中自动化webview代码：

desired\_caps = {}  
desired\_caps[**'platformName'**] = **'Android'** *#测试平台*desired\_caps[**'platformVersion'**] = **'6'** *#平台版本,不能写错*desired\_caps[**'deviceName'**] = **'test'** *#设备名称，多设备时需区分  
# desired\_caps['app'] = r'd:\apk\wv.apk' #app package名*desired\_caps[**'appPackage'**] = **'com.example.jcy.wvtest'** *#app package名*desired\_caps[**'appActivity'**] = **'.MainActivity'** *#app默认Activity*desired\_caps[**'unicodeKeyboard'**] = **True**desired\_caps[**'resetKeyboard'**] = **True***# desired\_caps['skipUnlock'] = True*desired\_caps[**'noReset'**] = **True**desired\_caps[**'newCommandTimeout'**] = 6000  
driver = webdriver.Remote(**'http://localhost:4723/wd/hub'**, desired\_caps) *#启动Remote RPC*driver.implicitly\_wait(10)  
**try**:  
 *# -----------------------* time.sleep(3)  
 print(driver.contexts)  
 print(driver.current\_context)  
  
 driver.switch\_to.context(**'WEBVIEW\_com.example.jcy.wvtest'**)  
  
 driver.find\_element\_by\_id(**'index-kw'**).send\_keys(**'松勤'**)  
  
 driver.find\_element\_by\_id(**'index-bn'**).click()  
  
 driver.switch\_to.context(**'NATIVE\_APP'**)  
 driver.find\_element\_by\_id(**'com.example.jcy.wvtest:id/navigation\_dashboard'**).click()  
 time.sleep(2)  
 driver.find\_element\_by\_id(**'com.example.jcy.wvtest:id/navigation\_notifications'**).click()  
  
  
 *# -----------------------***except**:  
 print(traceback.format\_exc())  
  
input(**'\*\*\*\* Press to quit..'**)  
driver.quit()

### 修改appium默认chromedriver:

http://note.youdao.com/noteshare?id=f68a98815f64216ea5d2cc433a2b303f

# 手机操作

## 查看通知

一、可以通过屏幕顶端向下滑动

driver.swipe(start\_x, start\_y, end\_x, end\_y, duration=300)

二、通过appium方法直接打开通知栏

driver.open\_notifications()

## 按键操作

调用press\_keycode方法

如按下安卓返回键：

driver.press\_keycode(4)

4代表安卓系统对于的返回键实际值

如果想要模拟其他键盘事件就要输入相应的按键码

具体的按键码对照表可以从网上查询到：

<https://developer.android.com/reference/android/view/KeyEvent>

（谷歌地址需要翻墙）