## 问题思考

在混合开发的App中，经常会有内嵌的H5页面。那么这些H5页面元素该如何进行定位操作呢？

## 解决思路

针对这种场景直接使用前面所讲的方法来进行定位是行不通的，因为前面的都是基于Andriod原生控件进行元素定位，而Web网页是单独的B/S架构，两者的运行环境不同因此需要进行上下文（context）切换，然后对H5页面元素进行定位操作。

## context

#### 简介

Context的中文翻译为：语境; 上下文; 背景; 环境，在开发中我们经常说“上下文”，那么这个“上下文”到底是指什么意思呢？

Android源码中的注释是这么来解释Context的：

Interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.

翻译如下：

关于应用程序环境的全局信息的接口。 这是一个抽象类，其实现由Android系统提供。 它允许访问特定于应用程序的资源和类，以及对应用程序级操作的调用，如启动活动、广播和接收意图等。

#### 通俗理解

在程序中context我们可以理解为**当前对象在程序中所处的一个环境**。 比如前面提到的App一个界面是属于Activity类型，也就是Android界面环境，但是当访问内嵌的网页是属于另外一个环境（网页环境），两者处于不同的一个环境。

#### 案例讲解

dr.fone app 内嵌网页地址：<https://drfone.wondershare.com/backup.html>

## WebView

WebView是Android系统提供能显示网页的系统控件，它是一个特殊的View，同时它也是一个ViewGroup，可以有很多其他子View。

在Android 4.4以下(不包含4.4)系统WebView底层实现是采用WebKit(<http://www.webkit.org/>)内核，而在Android 4.4及其以上Google 采用了chromium(<http://www.chromium.org/>)作为系统WebView的底层内核支持。

这里简单介绍下基于Chromium 的Webview和基于Webkit webview的差异，基于Chromium Webview提供更广的HTML5,CSS3,Javascript支持，在目前最新Android 系统版本5.0上基于chromium 37，Webview提供绝大多数的HTML5特性支持。Webkit JavaScript引起采用WebCore Javascript 在Android 4.4上换成了V8能直接提升JavaScript性能。另外Chromium 支持远程调试(Chrome DevTools)。

## H5元素定位环境搭建

#### 资源下载

* Chrome PC浏览器：

1. [官网下载地址](https://www.google.cn/chrome/)
2. [国内站点下载](https://www.chromedownloads.net)

* 手机版 Chrome

手机上安装Chrome最好到Google play去安装，手机上没有Google play可以先安装一个“GO谷歌安装器” 安装后注册Google play账号（由于众所周知的网络原因，Googleplay大陆地区无法使用）

* Chrome driver [下载地址](http://npm.taobao.org/mirrors/chromedriver/)

chrome driver要与Chrome的版本对应：

[ChromeDriver版本与Chrome版本对应表](http://npm.taobao.org/mirrors/chromedriver/2.37/notes.txt)

[Appium Chromedriver 官方说明文档](https://github.com/appium/appium/blob/71fac8d2759b18bb2a506a5c8359bfdf41a6443b/docs/cn/advanced-concepts/chromedriver.md)

chromediver路径 一般位于appium路径中的...\node\_modules\appium-chromedriver\chromedriver\win里面，如下所示：运行chromedriver.exe可以查看到当前的版本信息。

C:\Users\Shuqing\AppData\Roaming\npm\node\_modules\appium\node\_modules\appium-chromedriver\chromedriver\win

* 逍遥模拟器（夜神模拟器由于操作系统兼容问题，无法获取Webview context）
* dr.fone app 3.2.0

Tips：

以上所需资源可以在视频下方【获取素材】中第四章软件素材——《H5元素定位软件全家桶》文件夹中获取！

#### 操作步骤

1. 手机与电脑连接，开启USB调试模式，通过adb devices可查看到此设备。（设备系统Android 5.0以上）
2. 电脑端、移动端必须安装chrome浏览器。（尽量保证移动端chrome版本与PC端一致）根据对应的Chrome浏览器版本安装对应的Chrome driver。
3. App Webview开启debug模式
4. 在电脑端Chrome浏览器地址栏输入chrome://inspect/#devices，进入调试模式
5. 执行测试脚本

### Webview 调试模式检查与开启

#### 基础检查方式

1. 打开app对应的h5页面，在 chrome://inspect/#devices 地址中，检查是否显示对应的webview，如没有，则当前未开启调试模式。
2. 在自动化脚本中，进入到对应的H5页面，打印输出当前context,如果一直显示为Natvie App，则webview未开启。

#### 开启方式

在app中配置如下代码（在WebView类中调用静态方法setWebContentsDebuggingEnabled）：

**if** (Build.VERSION.SDK\_INT >=Build.VERSION\_CODES.KITKAT) {

WebView.setWebContentsDebuggingEnabled(**true**);

注：此步骤，一般需要App开发人员开启。

## H5定位实践案例

#### 测试场景

启动dr.fone app 进入backup H5页面中的输入邮箱并点击提交，然后返回

#### 测试环境

* 测试设备：逍遥模拟器 Android 5.1.1 /Chrome 66.0
* PC系统环境: Win10 64bit /Chrome 66.0
* 测试app: dr.fone3.2.0.apk
* H5页面地址：<https://drfone.wondershare.com/backup.html>

#### 需求分析

1. 先进入到H5页面，然后切换到context,再进行相关元素定位操作。
2. conetext切换：可以通过contexts()方法来获取到页面的所有context,然后切换到H5页面的context
3. 在H5页面进行元素定位操作

获取方法实践

contexts=driver.contexts

**print**(contexts)

#打印结果

['NATIVE\_APP', 'WEBVIEW\_com.android.launcher', 'WEBVIEW\_com.wondershare.drfone', 'WEBVIEW\_com.psiphon3']

#### 代码实现

by\_h5.py

from appium import webdriver

from selenium.webdriver.support.ui import WebDriverWait

desired\_caps={}

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

desired\_caps['platformVersion']='5.1.1'

desired\_caps['deviceName']='127.0.0.1:21503'

desired\_caps['app']=r'C:\Users\Shuqing\Desktop\dr.fone3.2.0.apk'

desired\_caps['appPackage']='com.wondershare.drfone'

desired\_caps['appActivity']='com.wondershare.drfone.ui.activity.WelcomeActivity'

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

driver.implicitly\_wait(5)

**print**('click BackupBtn')

driver.find\_element\_by\_id('com.wondershare.drfone:id/btnBackup').click()

**WebDriverWait**(driver,8).until(lambda x:x.find\_element\_by\_id('com.wondershare.drfone:id/btnRecoverData'))

**print**('click NextBtn')

driver.find\_element\_by\_id('com.wondershare.drfone:id/btnRecoverData').click()

**WebDriverWait**(driver,8).until(lambda x:x.find\_element\_by\_class\_name('android.webkit.WebView'))

contexts=driver.contexts

**print**(contexts)

#需android4.4及以上版本的系统中才会输出更多的webview

**print**('switch conetext')

driver.switch\_to.context('WEBVIEW\_com.wondershare.drfone')

**print**('edit email')

driver.find\_element\_by\_id('email').send\_keys('shuqing@wondershare.cn')

**print**('click sendBtn')

driver.find\_element\_by\_class\_name('btn\_send').click()

#切换context 点击返回

driver.switch\_to.context('NATIVE\_APP')

driver.find\_element\_by\_class\_name('android.widget.ImageButton').click()

#### 报错&解决方案

报错1

raise exception\_class(message, screen, stacktrace)

selenium.common.exceptions.WebDriverException:Message: An unknown server-side error occurred **while** processing **the** **command**.

Original error: Failed to start Chromedriver session: A new session could **not** be created.

(Original error: session **not** created exception: Chrome version must be >= 60.0.3112.0

报错2

**error:** Chromedriver: Chromedriver exited unexpectedly with code null, signal SIGTERM

【解决方案】

下载对应版本的chromedriver驱动，放置在： {Appium path}\node\_modules\appium\node\_modules\appium-chromedriver\chromedriver\win 替换即可

#### 参考资料

* <https://www.jianshu.com/p/94e0f9ab3f1d>
* <https://testerhome.com/topics/7866>
* <https://blog.csdn.net/typename/article/details/40425275>
* <https://www.cnblogs.com/niansi/p/6754766.html>