安卓Appium自动化系统架构分析

目录

[**一、** **Webdriver的使用** 1](#_Toc34167630)

[**二、 Appium的XPath定位** 1](#_Toc34167631)

[**1.** **如果元素text是唯一的，可以通过text文本定位** 1](#_Toc34167632)

[**2.** **如果元素id是唯一的，也可以id属性定位** 1](#_Toc34167633)

[**3.** **class属性唯一的话，同样可以通过class属性定位，有两种方法** 2](#_Toc34167634)

[**4.** **通过content-desc属性定位** 2](#_Toc34167635)

[**5. contains模糊定位** 3](#_Toc34167636)

[**6. 组合定位** 3](#_Toc34167637)

[**7. 层级定位** 4](#_Toc34167638)

[三、 Util->desired\_setting.py[启动Appium,初始化界面] 7](#_Toc34167639)

1. **安卓物理按键定义**

**二、 Appium的XPath定位**

**# 定位Text**

1. **如果元素text是唯一的，可以通过text文本定位**

   //\*[@text=’text文本属性’]

driver.find\_element\_by\_xpath("//\*[@text='扫一扫']").click()

**# 定位 resource-id**

1. **如果元素id是唯一的，也可以id属性定位**

   //\*[@resource-id=’id属性’]

driver.find\_element\_by\_xpath

("//\*[@resource-id='com.taobao.taobao:id/tv\_scan\_text']").click()

**# 联合@resource-id属性和@text文本属性来下定位**

同样可以联合上面两种方式定位，如下

driver.find\_element\_by\_xpth("//\*[@resource-id='com.taobao.taobao:id/tv\_scan\_text'][@text='扫一扫']").click()

**# 定位搜索框 //class属性**

1. **class属性唯一的话，同样可以通过class属性定位，有两种方法**

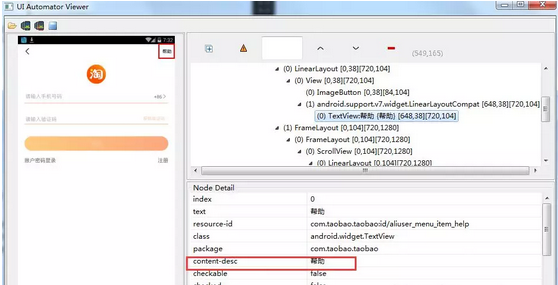
   第一种： //class属性

driver.find\_element\_by\_xpath("//android.widget.EditText").click()

   第二种： //\*[@class=’class属性’]

driver.find\_element\_by\_xpath("//\*[@class='android.widget.EditText']").click()

1. **通过content-desc属性定位**



   //\*[@content-desc=’desc的文本’]

#点登录/注册

driver.find\_element\_by\_xpath("//\*[@text='注册/登录']").click()

time.sleep(3)

**#content-desc定位**

driver.find\_element\_by\_xpath("//\*[@content-desc='帮助']").click()

**5. contains模糊定位**

1）、contains是模糊匹配的定位方法，对于一个元素的id或者text不是固定的，但有一部分是固定的，这种就可以模糊匹配。

  //[contains(@content-desc, ‘帮助’)]

**# contains匹配text**

driver.find\_element\_by\_xpath('//\*[contains(@text, "注册/登录")]').click()

time.sleep(3)

**# contains匹配textcontent-desc**

driver.find\_element\_by\_xpath("//\*[contains(@content-desc, '帮助')]").click()

2）、contains也能模糊匹配id和class属性

  //[contains(@resource-id, ‘id属性’)]

  //[contains(@clsss, ‘class属性’)]

[复制代码](javascript:void(0);)

**#** **contains定位搜索框class**

driver.find\_element\_by\_xpath("//\*[contains(@class, 'EditText')]").click()

time.sleep(3)

driver.back()

**#** **contains定位id**

driver.find\_element\_by\_xpath("//\*[contains(@resource-id, 'id/home\_searchedit')]").click()

**6. 组合定位**

1）、如果一个元素有2个属性，通过xpath也可以同时匹配2个属性,text, resource-id,class ,index,content-desc这些属性都能任意组合定位

[复制代码](javascript:void(0);)

**# id和class属性 定位搜索框**

id\_class = '//android.widget.EditText[@resource-id="com.taobao.taobao:id/home\_searchedit"]'

driver.find\_element\_by\_xpath(id\_class).click()

time.sleep(3)

driver.back()

**# text和index属性 定位登录/注册**

desc\_class = '//\*[@text="注册/登录" and @index="1"]'

driver.find\_element\_by\_xpath(desc\_class).click()

time.sleep(3)

**# class和text属性 定位输入手机号**

class\_text = '//android.widget.EditText[@text="请输入手机号码"]'

driver.find\_element\_by\_xpath(class\_text).send\_keys("512200893")

time.sleep(3)

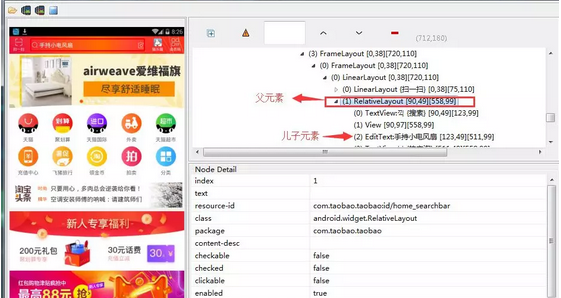
**# class和desc 定位帮助**

id\_desc = '//\*[contains(@resource-id, "aliuser\_menu\_item\_help") and @content-desc="帮助"]'

driver.find\_element\_by\_xpath(id\_desc).click()

**7. 层级定位**

1）、如果一个元素,它除了class属性（class属性肯定会有），其它属性啥都没有，这种情况用上面方法就不适用了，这个时候可以找他父元素，**通过父亲定位儿子**



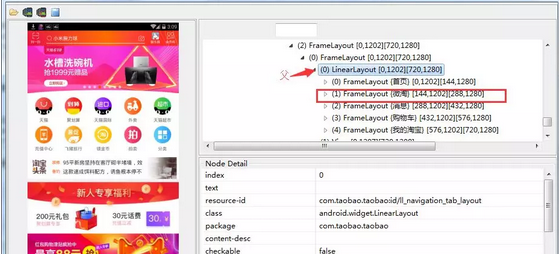
**#通过父亲定位儿子 搜索输入框**

fa\_sun = '//\*[@resoure-id="com.taobao.taobao:id/home\_searchbar"]/android.widget.EditText'

t = driver.find\_element\_by\_xpath(fa\_sun).text

print(t)

如果一个父元素下，有多个相同class的儿子时候，可以通过xpath的索引去取对应第几个，xpath是从1开始数的

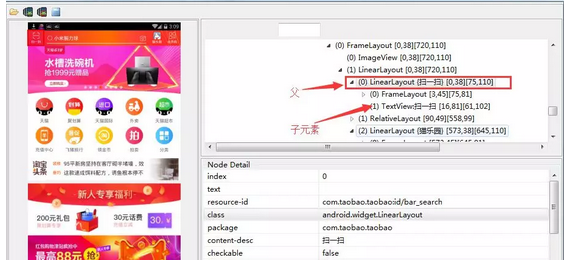


# 父元素下第2个儿子 微淘

fu\_sun2 = '//\*[@resource-id="com.taobao.taobao:id/ll\_navigation\_tab\_layout"]/android.widget.FrameLayout[2]'

driver.find\_element\_by\_xpath(fu\_sun2).click()

2）、相反的，可以**通过儿子定位父亲**



# 通过子元素定位父元素

# 方法一： ..

sun\_fa1 = '//\*[@resource-id="com.taobao.taobao:id/tv\_scan\_text"]/..'

c = driver.find\_element\_by\_xpath(sun\_fa1).tag\_name

print(c)

# 方法二 parent::\*

sun\_fa2 = '//\*[@resource-id="com.taobao.taobao:id/tv\_scan\_text"]/parent::\*'

d = driver.find\_element\_by\_xpath(sun\_fa1).tag\_name

print(d)

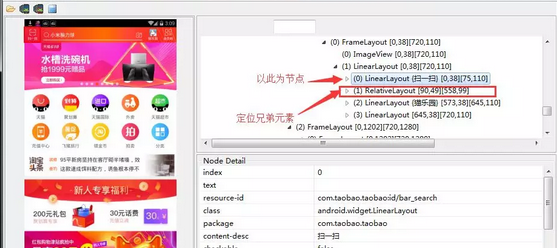
# 方法三 parent::android.widget.LinearLayout

sun\_fa3 = '//\*[@resource-id="com.taobao.taobao:id/tv\_scan\_text"]/parent::android.widget.LinearLayout'

e = driver.find\_element\_by\_xpath(sun\_fa1).tag\_name

print(e)

3）、通过子元素，先找到父元素，再找父元素下的子元素，同样可以**进行兄弟元素定位**



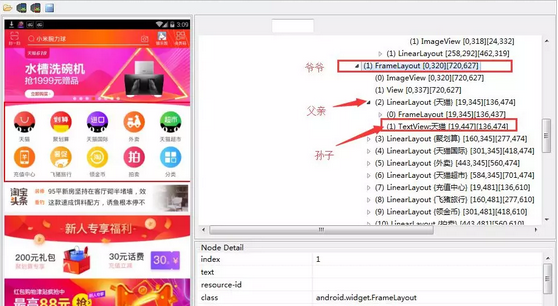
# 兄弟元素

xiongdi = '//\*[@resource-id="com.taobao.taobao:id/bar\_search"]/../android.widget.RelativeLayout'

x = driver.find\_element\_by\_xpath(xiongdi).tag\_name

print(x)

4）、通过层级关系，一层一层的往下找，同样可以**通过爷爷元素，定位到孙子元素**



[复制代码](javascript:void(0);)

#爷爷元素FrameLayout---第一个FrameLayout儿子---孙子TextView

x = '//android.widget.FrameLayout/android.widget.LinearLayout[1]/android.widget.TextView'

t = driver.find\_elements\_by\_xpath(x)

print(len(t))

# 打印文本信息

print(t[0].text

1. **Webdriver，Appium的原理**

**Selenium2:**

将浏览器原生的API封装成WebDriver API ，webdriver 是基于 http协议的；

**Appium:**

是基于 webdriver 协议添加对移动设备自动化api扩展而成的，基于tcp/ip协议（使用了socket接口）

**推论：** 凡是牵扯到客户端和服务器交互的，都有Http 协议

凡是牵扯到移动端社保的，都有tcp/ip 协议（封装成socket接口）

**appium的原理：**

appium-IOS 和安卓都差不多，有细小差别，分PC和手机两块讲：

1. 首先是PC端， 测试人员执行测试脚本（java,python等脚本）通过appium client 转换为json格式传递给appium server
2. appium server 启动了一个监听端口例如4724, 同时向手机端adb push 一个bootstrap.jar/bootstrap.js 的脚本，手机端通过该脚本同时监听端口4724

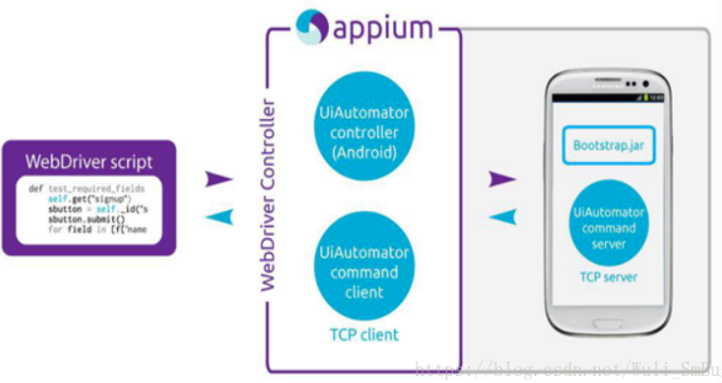
3、PC和手机端就通过这个端口实现了通信和交互，基于socket通信（一个封装了TCP/IP协议的接口）

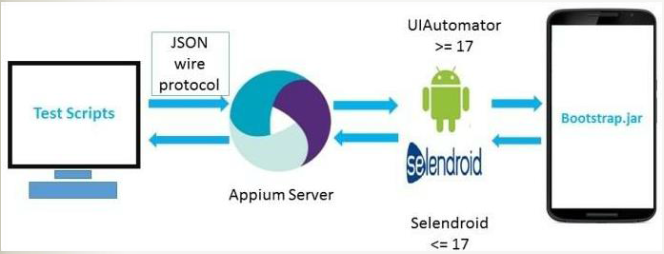
4、手机端通过该端口传输的命令执行APP， bootstrap里面封装了安卓和苹果的自动化测试框架UIautomator(低版本的安卓是instrumentation ) 执行相应的命令

5、执行完操作后通过端口返回给PC端，PC端根据返回结果 json 做校验，同时也知道了操作是否执行成功

**通俗的理解：由于客户端脚本(java, python, ruby)不能直接与浏览器通信，这时候可以把WebService当做一个翻译器，它可以把客户端代码翻译成浏览器可以识别的代码(比如js).客户端(也就是测试脚本)创建1个session，在该session中通过http请求向WebService发送restful的请求，WebService翻译成浏览器懂得脚本传给浏览器，浏览器把执行的结果返回给WebService,WebService把返回的结果做了一些封装(一般都是json格式)，然后返回给client,根据返回值就能判断对浏览器的操作是不是执行成功**

**四、 Util->desired\_setting.py[启动Appium,初始化界面]**





1. 调用webdriver客户端，向appium（服务器）请求，appium server会把请求转发给中间件Bootstrap.jar ，它是用java写的，安装在手机上.Bootstrap监听4724端口并接收appium 的命令，最终通过调⽤用UiAutomator的命令来实现。
2. Web请求时会通过设置的appPackage以及appActivity名称来初始化测试的app界面。



1. appPackage以及appActivity名称的获取方法：

**使用工具**：apktool

**方法**：使用apktool工具对要进行测试的apk进行反编译得到一个文件夹

**步骤**：

① 使用apktool工具对要进行测试的apk进行反编译得到一个文件夹

反编译方法参考：<https://blog.csdn.net/qq_27292113/article/details/79931268>

② 打开反编译文件夹中的AndroidManifest.xml文件

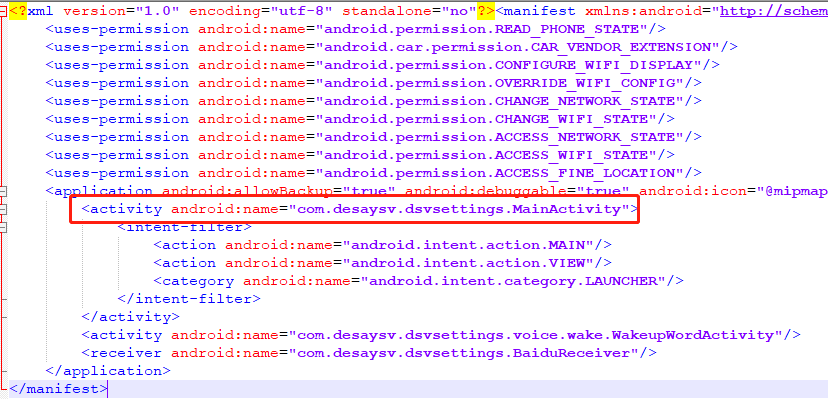
③ 查找appPackage名称

第一行的package字段的属性值即对应包名，如下包名为：com.desaysv.dsvsettings



④ 查看appActivity名称

中间部分activity的android:name的属性值即为appActivity名称，如下appActivity名为：com.desaysv.dsvsettings.MainActivity



1. 泥淖
2. 噶

注意事项：

1. 图片名称或者路径名不要包含中文（建议也不要用空格，可以使用”\_”,”-“间隔）