# 上次作业

<https://github.com/jcyrss/songqin-testdev/blob/master/appium/task/task2.md>

下面这些参数设置

desired\_caps['platformVersion'] **=** 'xxxxxxxx'  
desired\_caps['app'] **=** r'xxxxxxx'  
desired\_caps['appPackage'] **=** 'xxxxx'  
desired\_caps['appActivity'] **=** 'xxxx'

需要我们填入正确的内容

- 首先我们需要用 aapt.exe 命令查看 apk包的 appPackage 信息和 主 Activity 信息

d:\tools\Android\sdk\build-tools\25.0.3\aapt.exe dump badging d:\apk\duoduoCalculators.apk

最后这样填入

desired\_caps['platformVersion'] **=** '7'  
desired\_caps['deviceName'] **=** 'test'  
desired\_caps['app'] **=** r'd:\apk\duoduoCalculators.apk'  
desired\_caps['appPackage'] **=** 'com.ibox.calculators'  
desired\_caps['appActivity'] **=** '.SplashActivity'

--------------------

- 接下来我们要操作界面了， 看看这些要操作的界面元素的特点。

用 UIAutomator Viewer 查看应用界面元素信息，

发现 数字、加减乘除、等号 都有id， 是我们期望的。

代码如下，

equal **=** driver.**find\_element\_by\_id**('com.ibox.calculators:id/equal')  
  
driver.**find\_element\_by\_id**('com.ibox.calculators:id/digit3').**click**()  
driver.**find\_element\_by\_id**('com.ibox.calculators:id/plus').**click**()  
driver.**find\_element\_by\_id**('com.ibox.calculators:id/digit9').**click**()  
equal.**click**()  
driver.**find\_element\_by\_id**('com.ibox.calculators:id/mul').**click**()  
driver.**find\_element\_by\_id**('com.ibox.calculators:id/digit5').**click**()  
equal.**click**()

但是接下来需要我们去找结果对应的界面元素，

直接点击在 60 上，发现是对应的支出。

移动鼠标。 发现，鼠标放在下图中12数字上就可以找到60。

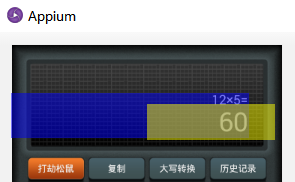
为什么？

因为 移动app开发，有时候会这样设计， 不同界面设计在同一个activity里面。

切换界面 的时候， 把先前界面上一些元素隐藏起来。 只是看不见，但是该元素还存在着。所以我们这里查看元素就发现这些隐藏元素。

没关系，晃动鼠标，还是可以找到我们想要的。

但是这个结果*这个TextView* 没有id*怎么办*



# 先查找父节点，  
# 再根据父节点元素 调用 find element 就是在父节点的范围内 查找

# 目标元素是第1个子节点，可以根据索引选取它

retLayout **=** driver.find\_element\_by\_id('com.ibox.calculators:id/cv')  
retTvs **=** retLayout.find\_elements\_by\_class\_name('android.widget.TextView')

retStr **=** retTvs[1].text  
print(retStr)

#注意是字符串不是数字

if retStr **==** '60'**:** print('pass')  
else**:** print('fail')

刚才的这段代码

是先找父元素， 然后再根据它 查找后代元素。

我们刚刚学过selenium，这种方法，我们以前web 自动化的时候，开始也这样用。

但是自从我们学会了css 和xpath后，最灵活的定位方式就是 css和xpath了。 可以一步到位。

那么appium是否有 css 和xpath的方法选择元素呢？

# 2 xpath

大部分情况还是可以用的。

对于一些比较复杂的元素（什么叫复杂的元素？上面的方法都没法定位，根据id，classname，accessbilityid都不行的时候 ）

的定位，我们可以用xpath

写法如下：

**driver.find\_element\_by\_xpath('//ele1/ele2[@attr="a1"]')**

**表达式的语法规则和 以前学习的Selenium里面一样的**

以前我们做web自动化的时候， xpath中每个节点名是html的tagname

但是在appium中， 每个节点名对应class

比如：要选择所有的文本节点

//android.widget.TextView

我们来看一个例子， 运行头条，我们要点击下面的发现按钮



研究可以发现其id不能用，其它的几个元素（阅读、消息、我的）id都一样。

那我们用xpath试试看，

我们找到resource-id='io.manong.developerdaily:id/tab\_bar

可以这样写xpath表达式

//\*[@resource-id='io.manong.developerdaily:id/tab\_bar']//android.widget.TextView

对应代码如下

*# 这一步是为了防止 界面刚出来的时候，还没显示全，就去获取所有的textview了*driver.find\_element\_by\_class\_name("android.widget.ImageButton")  
  
time.sleep(1)  
  
xpath **=** "//\*[@resource-id='io.manong.developerdaily:id/tab\_bar']//android.widget.TextView"  
  
eles **=** driver.find\_elements\_by\_xpath(xpath)  
for ele in eles**:** print (ele.text)

再根据 要找的是返回列表其中第3个元素

eles[2].click()

如果想直接用 表达式 找到 第3个textview

xpath **=** "//\*[@resource-id='io.manong.developerdaily:id/tab\_bar']//android.widget.RelativeLayout[4]//android.widget.TextView"  
ele **=** driver.find\_element\_by\_xpath(xpath)  
ele.click()

运行一下，发现也可以。

# Appium的Inspector

前面我们查看界面元素的时候，用的是UIAutomatorviewer，打开给大家看看

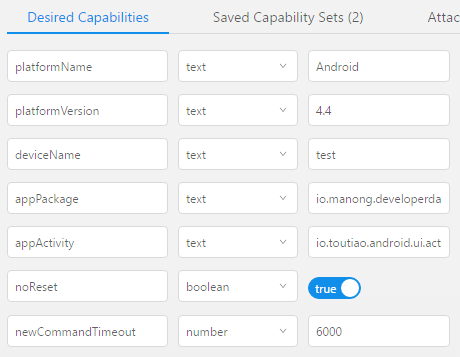
除了用uiautomatorviewer，我们还可以用appium desktop来查看

appium desktop是通过 一个叫Inspector 的功能来查看元素的

需要我们手动创建一个session，其实也就是一个客户端，和appium server连接



填入参数，告诉appium desktop 我们的session的参数



操作一下， 最后点击右下角 save as 保存为 toutiao

关闭， 以后再次start new session 就可以选择 保存的session，点击 StartSession

界面元素和uiautomatorviewer的看法大概相同，

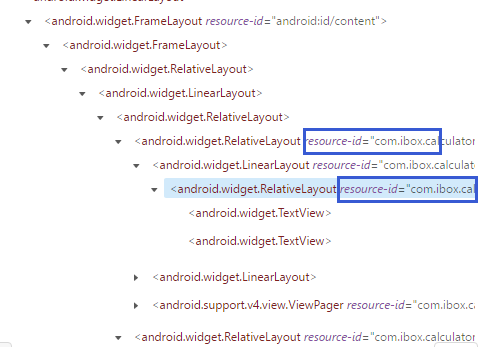
鼠标在界面游动，高亮显示，也可以显示元素的属性

刷新界面的按键是这个，如果界面更新，可以按它

inspector 特别好的地方是， 如果元素有ID， 直接就在树的节点里面添加了

这样一目了然， 方便我们知道，不用一个个点过去看。

和UIAutomatorviewer 对比一下



----------------

另外一个好处就是，我们选择

可以直接点击界面元素，操作一下

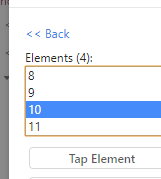
这样不用再手机和查看工具里面切来切去了

甚至，可以用这个 查看我们的选择表达式是否正确

比如

io.manong.developerdaily:id/tv\_tab\_title

可以选中 4个，



点击其中每个，可以对应显示界面高亮！！！

# 对元素的tap操作

一个新的操作界面的方法，是appium所特有的，就是tap方法

tap 方法和 click类似，都是点击界面。

但是最大的区别是， tap是针对坐标而不是针对找到的元素。

我们怎么知道这个元素的坐标呢？

可以UIAutomatorviewer 直接光标移动，看右边的属性提示。

打开lesson3/tap.py，

点击tap查看它的注释，解释一下

两个参数：

第一个表示点击的坐标 ，注意最多可以有五个坐标，代表5根手指。所以是list类型

第二个表示 按住停留的时间。

现在我们要点击一个元素，list中只要一个值就可以了。 停留时间也可以比较短

不到万不得已不要用tap方法，因为不同手机屏幕分辨率不同，同样的元素在不同手机的位置不同。

如果必须要用坐标的方式，又有多种手机要测试

可以把这些元素的坐标写到一个配置文件中

打开lesson3/cfg.py

coordinates **=** {  
 '1080p'**:**{  
 '首页\_+' **:** (548, 1839),  
 '登录页\_邮箱图标' **:** (147, 1802),  
 '登录页\_邮箱地址' **:** (264, 370),  
 '登录页\_邮箱密码' **:** (264, 570),  
 '登录页\_登录按钮' **:** (264, 780),  
 },  
  
 '720p'**:** {  
 '首页\_+'**:** (640, 1250),  
 '登录页\_邮箱图标'**:** (147, 1802),  
 '登录页\_邮箱地址'**:** (264, 370),  
 '登录页\_邮箱密码'**:** (264, 570),  
 '登录页\_登录按钮'**:** (264, 570),  
 }  
  
}  
  
cfg\_phone\_resolution **=** '1080p'

这样， 使用的代码，比如lesson3/tap2.py

就可以这样写了

from appium import webdriver  
  
desired\_caps **=** {}  
desired\_caps['platformName'] **=** 'Android' *# 测试平台*desired\_caps['platformVersion'] **=** '7' *# 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\toutiao.apk'*desired\_caps['appPackage'] **=** 'io.manong.developerdaily' *# app package名*desired\_caps['appActivity'] **=** 'io.toutiao.android.ui.activity.LaunchActivity' *# app默认Activity*desired\_caps['unicodeKeyboard'] **=** True *# 一定要有该参数，否则unicode 输入的中文无效*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)  
print(driver.session\_id)  
  
*# 不得不用sleep*import time  
time.**sleep**(10)  
  
from cfg import cfg\_phone\_resolution,coordinates  
  
coordinate **=** coordinates[cfg\_phone\_resolution]  
  
  
driver.**tap**([coordinate['首页\_+']], 300)  
time.**sleep**(2)  
  
  
driver.**tap**([coordinate['登录页\_邮箱图标']], 300)  
time.**sleep**(1)  
  
driver.**tap**([coordinate['登录页\_邮箱地址']], 300)  
time.**sleep**(1)  
  
*# 没有WebElement对象 ，如何输入字符？  
# adb shell input text "<your string>"*import os  
os.**system**('adb shell input text "jcyrss@163.com"')  
*#注意，如果要输入中文，需要下载一个adb键盘应用，  
# 参考 https://blog.csdn.net/slimboy123/article/details/54140029*time.**sleep**(1)  
  
driver.**tap**([coordinate['登录页\_邮箱密码']], 300)  
time.**sleep**(1)  
os.**system**('adb shell input text "sdfsdf"')  
  
driver.**tap**([coordinate['登录页\_登录按钮']], 300)  
  
input('\*\*\*\* Press to quit..')  
driver.**quit**()