# java UIAutomator 选元素（50）

前面我们学习的选择 界面元素的方式，

根据id，classname， accessibilityid，xpath，

这些方法，其实底层都是利用了 uiautomator的API功能实现的

我们来看一下安卓官网是如何描述uiautomator的API的

打开下面的连接，

<https://developer.android.google.cn/topic/libraries/testing-support-library/index.html#UIAutomator>

UI Automator 测试框架的主要功能包括：

* 用于检查布局层次结构的查看器（就是UIAutomatorviewer）。如需了解详细信息，请参阅 UI Automator 查看器。
* 在目标设备上检索状态信息并执行操作的 API（指对设备进行各种操作的API，我们要知道，所谓API 就是库的编程接口）。如需了解详细信息，请参阅访问设备状态。

这些操作，大都可通过appium 的webdriver的方法实现

* 支持跨应用 UI 测试的 API(这个就是选择和操作界面元素的API)。如需了解详细信息，请参阅 UI Automator API。

## UI Automator API

利用 UI Automator API，您可以编写稳健可靠的测试，而无需了解app的实现详情。您可以使用这些 API 在多个应用中捕获（就是选择）和操作 UI 组件

这些API 主要是通过下面这些类来实现的

UiCollection UiObject UiScrollable UiSelector Configurator

其中UiSelector 是我们appium自动化代码选择元素的时候，直接使用的 。

UiObject 是选择到的元素对象

* [UiSelector](https://developer.android.com/reference/android/support/test/uiautomator/UiSelector.html)： UiSelector类就是用来选择的web元素的。

我们appium 自动化程序里面driver.select\_element\_byxxx, 发给到手机上的‘木马’时， 转换为相应的这个UiSelector调用。

打开链接到UiSelector类<https://developer.android.com/reference/android/support/test/uiautomator/UiSelector.html>

刚才说我们前面的根据id，classname， accessibility id，xpath，其实底层都是利用了 uiautomator的这个selecotor功能实现的。

比如，我们要根据classname查询， appium server就发送类似

findbyclassname：android.widget.Button

的命令给 木马 程序，里面会调用



方法

根据id呢？对应resourceId 方法。。。。。

根据accessibilityid呢？对应description 方法。。。。。

根据xpath呢？ UIAutomator API里面没有直接对应的方法,

这是appium 自己分析xpath的表达式计算的， 比如 //\*[@resource-id='abc'] ，就调用resourceId 方法。

最终都是通过 这个UI Automator API实现的。

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

所以**如果我们能直接使用调用UI Automator API的代码**，就可以实现最为直接的控制。

所以，appium确实也提供了这样的方法， 可以直接发送 java代码 字符串给

‘木马’，‘木马’ 会在 在手机上直接执行这些代码,这样我们可以精确的使用底层的方法选择界面元素.

## 根据class name 和text属性

比如，上次 开发者头条的应用 我们想点击 我的 按钮，

UIAutomatorviewer查看属性

可以利用它的文本值为我的， class为android.widget.TextView，

code **=** 'new UiSelector().text("我的").className("android.widget.TextView")'  
ele **=** driver.find\_element\_by\_android\_uiautomator(code)  
  
ele.click()

UiSelector里面有些元素选择的方法是以前所不具备的，

比如根据 正则表达式搜索文本，就是其他方法不支持的。

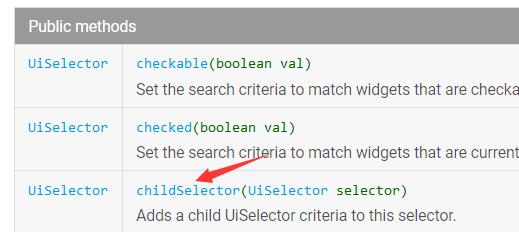
## childSelector 和 Instance

像xpath这种// 表示后代关系的，

UIAutomator通过childSelector 和 instance ，index 来实现

打开文档查看

<https://developer.android.com/reference/android/support/test/uiautomator/UiSelector.html>



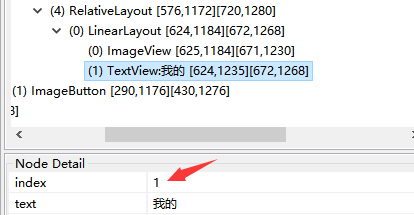
childSelector用来选择后代元素（可以是非直接子元素），

instance 和 index的区别：

都是从0开始计数，但是：

其中instance是匹配的结果所有元素里面 的第几个元素

而index则是其父元素的几个节点，类似xpath 里面的\*[n]



所以开发者头条里面的 我的，是下面整个tab bar 的后代元素，可以这样

code **=** 'new UiSelector().resourceId("io.manong.developerdaily:id/tab\_bar").childSelector(new UiSelector().className("android.widget.TextView").instance(3))'  
ele **=** driver.find\_element\_by\_android\_uiautomator(code)  
  
print(ele.text)  
ele.click()

# 操作界面

## swipe

手机最常见的操作之一就是滑动。

我们做移动app测试的时候，经常需要滑动界面。

怎么模拟滑动呢？

Appium 的 WebDriver的 swipe方法，就提供了这个功能

我们来看一下这个方法的说明和参数

查看swipe的代码注释里面的说明

*# convenience method added to Appium (NOT Selenium 3)*def swipe(self, *start\_x*, *start\_y*, *end\_x*, *end\_y*, *duration***=**None)**:** *"""Swipe from one point to another point, for an optional duration.  
  
 :Args:  
 - start\_x - x-coordinate at which to start  
 - start\_y - y-coordinate at which to start  
 - end\_x - x-coordinate at which to stop  
 - end\_y - y-coordinate at which to stop  
 - duration - (optional) time to take the swipe, in ms.  
  
 :Usage:  
 driver.swipe(100, 100, 100, 400)  
 """  
 # `swipe` is something like press-wait-move\_to-release, which the server  
 # will translate into the correct action*

坐标就是x/y坐标

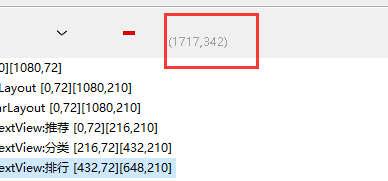
duration是滑动从起点到终点坐标所耗费的时间。

注意这个时间非常重要。

比如手机翻看新闻，如果很快会导致，飞速上滑（扫动）很多，而很慢就不同，

怎么查看到坐标？

UIAutomatorviewer 可以右上角直接看到鼠标当前位置的坐标。



比如华为应用市场，要翻动广告页，就可以

ele **=** driver.find\_element\_by\_id('com.huawei.appmarket:id/backimage')  
  
for i in range(10)**:** driver.swipe(start\_x**=**548, start\_y**=**200, end\_x**=**144, end\_y**=**200, duration**=**500)  
 time.sleep(0.5)

因为Y坐标不变

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

但是

如果屏幕分辨率不同， 代码就不对了，比如2k屏

有一种解决方法是，

动态获取元素坐标，而不是查看当前手机的固定坐标。

怎么动态获取呢？

根据元素的location属性，是个dict，

里面有两个key x y

保存了该元素的左上角坐标

但是这个例子里面我需要的右下角的坐标，怎么得到右下角的坐标呢？

根据元素的size属性，是个dict，里面有两个key ：width，height 保存了该元素的宽、高

就可以得到右下角的坐标

改进代码如下

ele **=** driver.**find\_element\_by\_id**('com.huawei.appmarket:id/backimage')  
time.**sleep**(2)  
  
location **=** ele.location  
size1 **=** ele.size  
print(location, size1)  
x1 **=** location['x'] **+** int(size1['width'] **\*** 0.8)  
x2 **=** location['x'] **+** int(size1['width'] **\*** 0.2)  
y **=** location['y'] **+** int(size1['height'] **\*** 0.5)  
  
for i in range(10)**:** driver.**swipe**(start\_x**=**x1, start\_y**=**y, end\_x**=**x2, end\_y**=**y, duration**=**500)  
 time.**sleep**(0.5)