## 元素定位

与Web自动化测试一样，app自动化测试过程中最重要一个环节就是元素定位，只有准确定位到了元素才能进行相关元素的操作，如输入、点击、拖拽、滑动等。appium提供了许多元素定位的方法，如id定位、name定位、class定位、层级定位等等.... 接下来将会给大家来实践运用这些定位技巧。

## 元素定位方式

* id
* name
* class
* List定位
* 相对定位
* Xpath定位
* H5页面元素定位
* Uiautomator定位

### id定位

日常生活中身边可能存在相同名字的人，但是每个人的身份证号码是唯一的，在app界面元素中也可以使用id值来区分不同的元素，然后进行定位操作。Appium中可以使用 **find\_element\_by\_id()** 方法来进行id定位。

###### 测试场景1

1. 安装考研帮kaoyan3.1.0.apk
2. 点击升级页面取消按钮
3. 点击引导页面的跳过按钮

kyb\_cancel\_skip.py

**from** appium **import** webdriver

desired\_caps={}

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

desired\_caps['deviceName']='127.0.0.1:62025'

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

desired\_caps['app']=r'C:\Users\Shuqing\Desktop\kaoyan3.1.0.apk'

desired\_caps['appPackage']='com.tal.kaoyan'

desired\_caps['appActivity']='com.tal.kaoyan.ui.activity.SplashActivity'

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

driver.implicitly\_wait(5)

driver.find\_element\_by\_id('android:id/button2').click()

driver.find\_element\_by\_id('com.tal.kaoyan:id/tv\_skip').click()

相关阅读：[selenium元素定位](http://www.51zxw.net/list.aspx?page=3&cid=615) 还记得那年的selenium元素定位吗？？

### 思考

* 如果安装的版本最新的包，或者升级到了最新的版本，则启动后没有升级弹窗元素该如何处理？
* 跳过引导页面首次启动和非首次启动场景该如何处理？

##### 方案探索1——if条件判断

有同学可能想到用if来做条件判断，判断元素是否存在，存在则点击，不存在则跳过。

kyb\_cancel\_skip\_if.py

**from** appium **import** webdriver

desired\_caps={}

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

desired\_caps['deviceName']='127.0.0.1:62025'

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

desired\_caps['app']=r'C:\Users\Shuqing\Desktop\kaoyan3.1.0.apk'

desired\_caps['appPackage']='com.tal.kaoyan'

desired\_caps['appActivity']='com.tal.kaoyan.ui.activity.SplashActivity'

desired\_caps['noReset']='True'

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

driver.implicitly\_wait(5)

cancelBtn=driver.find\_element\_by\_id('android:id/button2')

skipBtn=driver.find\_element\_by\_id('com.tal.kaoyan:id/tv\_skip')

**if** cancelBtn:

cancelBtn.click()

**else**:

print('no cancelBtn')

**if** skipBtn:

skipBtn.click()

**else**:

print('no skipBtn')

那么执行结果究竟如何呢？详见视频操作演示。

##### 方案探索2——异常捕捉

既然上面的if语句判断无法生效，但是我们发现一个突破口，那就是捕捉NoSuchElementException异常。

[Python异常处理视频教程](http://www.51zxw.net/show.aspx?id=60320&cid=615)

**kyb\_cancel\_skip\_try.py**

**from** appium **import** webdriver

**from** selenium.common.exceptions **import** NoSuchElementException

desired\_caps={}

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

desired\_caps['deviceName']='127.0.0.1:62025'

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

desired\_caps['app']=r'C:\Users\Shuqing\Desktop\kaoyan3.1.0.apk'

desired\_caps['appPackage']='com.tal.kaoyan'

desired\_caps['appActivity']='com.tal.kaoyan.ui.activity.SplashActivity'

desired\_caps['noReset']='True'

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

driver.implicitly\_wait(5)

**def** **check\_cancelBtn**():

print("check\_cancelBtn")

**try**:

cancelBtn = driver.find\_element\_by\_id('android:id/button2')

**except** NoSuchElementException:

print('no CancelBtn')

**else**:

cancelBtn.click()

**def** **check\_skipBtn**():

print("check\_skipBtn")

**try**:

skipBtn = driver.find\_element\_by\_id('com.tal.kaoyan:id/tv\_skip')

**except** NoSuchElementException:

print('no skipBtn')

**else**:

skipBtn.click()

check\_updateBtn()

check\_skipBtn()

### id定位综合实践——自动登录

##### 测试场景

1. 启动App，进入到登录界面
2. 在登录页面输入用户名“自学网2018”，密码‘zxw2018’ 然后点击登录。

##### 需求分析

1. 可以把启动后检测升级弹窗和引导页面的模块抽离作为独立的模块被其他模块调用，提高代码复用率。
2. 获取用户名密码输入框和登录按钮的元素id属性,另外要考虑启动时App之前是否登录过账号，已经登录过和未登录场景流程不一样。
3. 注意：send\_keys()传入中文时需要在capability中配置如下内容：

desired\_caps['unicodeKeyboard']="True"

desired\_caps['resetKeyboard']="True"

设置之后会有Appium的输入法守护来执行输入操作

##### 代码实现

kby\_login.py

**from** **find\_element**.capability **import** **driver**,**NoSuchElementException**

**def** **login**():

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/login\_email\_edittext').clear()

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/login\_email\_edittext').send\_keys('自学网2018')

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/login\_password\_edittext').send\_keys('zxw2018')

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/login\_login\_btn').click()

**try**:

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/mainactivity\_button\_mysefl')

**except** **NoSuchElementException**:

**login**()

**else**:

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/mainactivity\_button\_mysefl').click()

**driver**.find\_element\_by\_id('com.tal.kaoyan:id/activity\_usercenter\_username').click()

**login**()

注意：

使用Appium做了输入操作之后，如果出现输入法无法唤起，可以在系统设置——语言和输入法——将当前输入法替换为系统输入法或者其他输入法。

### name定位

根据name进行定位，对于android来说，就是text属性

用法

**from** **find\_element**.capability **import** \*

**driver**.find\_element\_by\_name('请输入用户名').send\_keys('自学网2017')

**driver**.find\_element\_by\_name('登录').click()

说明：由于text稳定性不是很好，所以appium 1.5开始废弃了该方法。

### classname定位

classname定位是根据元素类型来进行定位，但是实际情况中很多元素的classname都是相同的，

如上例中登录页面中的用户名和密码都是clasName属性值都是：**“android.widget.EditText”** 因此只能定位第一个元素也就是用户名，而密码输入框就需要使用其他方式来定位，这样其实很鸡肋.一般情况下如果有id就不必使用classname定位。

**by\_classname.py**

from find\_element.capability **import** driver

driver.find\_element\_**by**\_**class**\_name('android.widget.EditText').send\_keys('自学网2018')

driver.find\_element\_**by**\_**class**\_name('android.widget.EditText').send\_keys('zxw2018')

driver.find\_element\_**by**\_**class**\_name('android.widget.Button').click()

### 相对定位

相对定位是先找到该元素的有对应属性的父元素节点，然后基于父元素进行元素定位。

##### 测试案例

不使用id元素定位方式，在新用户注册界面点击添加头像按钮。

##### 代码实现

by\_relative.py

from find\_element.capability **import** driver

driver.find\_element\_**by**\_id('com.tal.kaoyan:id/login\_register\_text').click()

root\_element=driver.find\_element\_**by**\_id('com.tal.kaoyan:id/activity\_register\_parentlayout')

root\_element.find\_element\_**by**\_**class**\_name('android.widget.ImageView').click()

### xpath定位

xpath定位是一种路径定位方式，主要是依赖于元素绝对路径或者相关属性来定位，但是绝对路径xpath执行效率比较低（特别是元素路径比较深的时候），一般使用比较少。通常使用xpath相对路径和属性定位。

1.xpath路径表达式

| **表达式** | **描述** |
| --- | --- |
| / | 从根节点选取。 |
| // | 从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。 |
| nodename | 选取此节点的所有子节点。 |
| . | 选取当前节点。 |
| .. | 选取当前节点的父节点。 |
| @ | 选取属性。 |

| **通配符** | **描述** |
| --- | --- |
| \* | 匹配任何元素节点。 |
| @\* | 匹配任何属性节点。 |
| node() | 匹配任何类型的节点。 |

##### 实践案例

使用xpath定位元素来进行登录操作。

by\_xpath.py

**from** **find\_element**.capability **import** **driver**

**driver**.find\_element\_by\_xpath('//android.widget.EditText[@text="请输入用户名"]').send\_keys('zxw1234')

**driver**.find\_element\_by\_xpath('//\*[@class="android.widget.EditText" and @index="3"]').send\_keys('zxw123456')

**driver**.find\_element\_by\_xpath('//android.widget.Button').click()

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

扩展资料：[xpath语法](http://www.w3school.com.cn/xpath/xpath_syntax.asp)

### List定位

前面我们提到相同的classname属性值元素无法区分定位，那么在本节课将使用List定位来解决这个问题。List定位首先是使用**find\_elements\_by\_XX**获取一组相同的class属性的元素，然后使用数组下标来区分标记不同元素进行相关操作。

##### 测试案例1

在新用户注册界面点击添加头像按钮后，选择指定的图片保存作为头像。

by\_list.py

**from** find\_element.capability import driver

driver.find\_element\_by\_id('com.tal.kaoyan:id/login\_register\_text').click()

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_userheader').click()

images=driver.find\_elements\_by\_id('com.tal.kaoyan:id/item\_image')

images[10].click()

driver.find\_element\_by\_id('com.tal.kaoyan:id/save').click()

### list定位综合案例——用户注册

##### 测试场景

1. 进入注册界面设置头像
2. 输入注册信息：用户名、密码、邮箱
3. 完善院校和专业信息 （院校：上海-同济大学 专业：经济学类-统计学-经济统计学）
4. 完成注册

##### 代码实现

kyb\_register.py

from find\_element.capability **import** driver

**import** random

#进入注册界面选择并设置头像

driver.find\_element\_by\_id('com.tal.kaoyan:id/login\_register\_text').click()

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_userheader').click()

images=driver.find\_elements\_by\_id('com.tal.kaoyan:id/item\_image')

images[10].click()

driver.find\_element\_by\_id('com.tal.kaoyan:id/save').click()

#注册信息填写

username='zxw2018'+'FLY'+str(random.randint(1000,9000))

print('username: %s' %username)

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_username\_edittext').send\_keys(username)

password='zxw'+str(random.randint(1000,9000))

print('password: %s' %password)

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_password\_edittext').send\_keys(password)

email='51zxw'+str(random.randint(1000,9000))+'@163.com'

print('email: %s' %email)

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_email\_edittext').send\_keys(email)

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_register\_register\_btn').click()

#院校选择

driver.find\_element\_by\_id('com.tal.kaoyan:id/perfectinfomation\_edit\_school\_name').click()

#选择省份

driver.find\_elements\_by\_id('com.tal.kaoyan:id/more\_forum\_title')[1].click()

#选择具体院校--同济大学

driver.find\_elements\_by\_id('com.tal.kaoyan:id/university\_search\_item\_name')[1].click()

#专业选择

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_perfectinfomation\_major').click()

#选择经济学类-统计学-经济统计学

driver.find\_elements\_by\_id('com.tal.kaoyan:id/major\_subject\_title')[1].click()

driver.find\_elements\_by\_id('com.tal.kaoyan:id/major\_group\_title')[2].click()

driver.find\_elements\_by\_id('com.tal.kaoyan:id/major\_search\_item\_name')[1].click()

#点击“进入考研帮”按钮

driver.find\_element\_by\_id('com.tal.kaoyan:id/activity\_perfectinfomation\_goBtn').click()

注意：运行前记得将noRest设置为：desired\_caps['noReset']='False' 以免之前的注册残留信息干扰。

### 报错&解决方案

###### 元素定位报错

selenium.common.exceptions.NoSuchElementException: Message: An element could **not** be located on the page using the given search parameters.

【解决方案】检查元素id值是否写错。

#### 参考资料

<https://blog.csdn.net/u011541946/article/details/77922304>