## 应用背景

滑动操作一般是两点之间的滑动，而实际使用过程中用户可能要进行一些多点连续滑动操作。如九宫格滑动操作，连续拖动图片移动等场景。那么在Appium中该如何模拟这类操作呢？

## TouchAction

Touch Action包含一些列操作，比如按压、长按、点击、移动、暂停。由着些不同操作可以组成一套动作。使用TochAction需要先导入对应的模块

from appium.webdriver.common.touch\_action import TouchAction

#### 按压

方法：press() 开始按压一个元素或坐标点（x,y）。通过手指按压手机屏幕的某个位置。 press也可以接收屏幕的坐标（x,y）。

press(self, el=None, x=None, y=None)

TouchAction(driver).press(x=0,y=308)

#### 长按

方法：longPress() 开始按压一个元素或坐标点（x,y）。 相比press()方法，longPress()多了一个入参，既然长按，得有按的时间吧。duration以毫秒为单位。1000表示按一秒钟。其用法与press()方法相同。

long\_press(self, el=None, x=None, y=None, duration=1000)

#### 点击

方法：tap() 对一个元素或控件执行点击操作。用法参考press()。

tap(self, element=None, x=None, y=None, count=1)

#### 移动

方法：move\_to() 将指针从上一个点移动到指定的元素或点。

move\_to(self, el=None, x=None, y=None)

注意：

移动到目位置有时是算绝对坐标点，有时是基于前面一个坐标点的偏移量，这个要结合具体App来实践。

#### 暂停

方法：Wait()

**wait**(self, ms=0)

暂停脚本的执行，单位为毫秒。

#### 释放

方法release() 结束的行动取消屏幕上的指针。

**release**(self)

#### 执行

perform() 执行的操作发送到服务器的命令操作。

**perform**(self)

## TouchAction实战——九宫格滑动操作

九宫格是一种比较常见的图案加密方式，目前很多App都支持设置图案锁，Android原生系统也支持设九宫格图案锁屏。那么我们该如何使用Appium进行滑动操作呢？

#### 测试场景

安装启动随手记App 启动App后在密码设置选项中开启手机密码并滑动九宫格设置如下图形密码：



#### 测试环境

* 夜神模拟器 Android 5.1.1
* 随手记Android版 V10.5.6.0
* Win 10 64bit
* Appium 1.7.2

#### 代码实现

touch\_action.py

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

**from** time **import** sleep

**from** appium.webdriver.common.touch\_action **import** TouchAction

**from** selenium.webdriver.support.ui **import** WebDriverWait

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

desired\_caps={}

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

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

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

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

desired\_caps['appPackage']='com.mymoney'

desired\_caps['appActivity']='com.mymoney.biz.splash.SplashScreenActivity'

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

driver.implicitly\_wait(5)

**def** **get\_size**():

x=driver.get\_window\_size()['width']

y=driver.get\_window\_size()['height']

**return** x,y

**def** **swipeLeft**():

l=get\_size()

x1=int(l[0]\*0.9)

y1=int(l[1]\*0.5)

x2=int(l[0]\*0.1)

driver.swipe(x1,y1,x2,y1,1000)

**def** **swipeUp**():

l = get\_size()

x1 = int(l[0] \* 0.5)

y1 = int(l[1] \* 0.95)

y2 = int(l[1] \* 0.35)

driver.swipe(x1, y1, x1, y2, 1000)

*#等待启动页面元素，然后向左滑动两次,跳过引导页面*

WebDriverWait(driver,6).until(**lambda** x:x.find\_element\_by\_id("com.mymoney:id/next\_btn"))

**for** i **in** range(2):

swipeLeft()

sleep(1)

*#点击“开始随手记”按钮*

driver.find\_element\_by\_id('com.mymoney:id/begin\_btn').click()

*#检测是否有活动页面弹窗，如果有就点击关闭*

**try**:

closBtn=driver.find\_element\_by\_id('com.mymoney:id/close\_iv')

**except** NoSuchElementException:

**pass**

**else**:

closBtn.click()

*#点击更多菜单*

driver.find\_element\_by\_id('com.mymoney:id/nav\_setting\_btn').click()

*#等待界面菜单加载出来，然后向上滑动*

WebDriverWait(driver,6).until(**lambda** x:x.find\_element\_by\_id("com.mymoney:id/content\_container\_ly"))

swipeUp()

*#点击高级菜单*

driver.find\_element\_by\_android\_uiautomator('new UiSelector().text("高级")').click()

*#点击密码与手势密码菜单*

driver.find\_element\_by\_id('com.mymoney:id/password\_protected\_briv').click()

*#点击手势密码保护*

driver.find\_element\_by\_id('com.mymoney:id/lock\_pattern\_or\_not\_sriv').click()

*#连续滑动两次设置图案密码*

**for** i **in** range(2):

TouchAction(driver).press(x=243,y=381).wait(2000)\

.move\_to(x=455,y=390).wait(1000)\

.move\_to(x=643,y=584).wait(1000)\

.move\_to(x=647,y=784).wait(1000)\

.release().perform()

#### 参考资料：

<https://blog.csdn.net/weixin_40180628/article/details/79170053>