

Appium-阶段二 - appium的元素定位xpath,数据驱动及pageobject的引入

Appium 高手教程

上海-Terry
QQ:1740624443

第二节课程

- 1)使用xpath定位，及动态元素定位处理方式。
- 2)常用手机动作的使用swipe,draganddrop,tap等
- 3)数据驱动及pageobject的引入

一，xpath的使用，及动态xpath

和selenium一样通过id,name,tagname等50%是无法解决真正定位问题的，自动化最大的问题归根结底就是无法识别对象或者识别的对象是动态变化。

为了解决以上的问题我们常使用xpath来解决，只要appium能识别的话就能100%定位到。

下面我使用美团作为我们的这次课程的主要练习的对象：

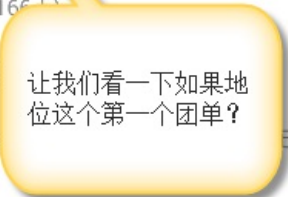
apk可以从下面的地址下载：

www.meituan.com

我们可以通过adb install <path>将apk安装到机器上去；

下面大家思考一下，我们如何定位团购网站的第一个团单的主标题，然后可以做后面的点击进入团购详情页，这里需要注意的是不同的地理位置这个团单是不一样的

入口是:美食->美食列表



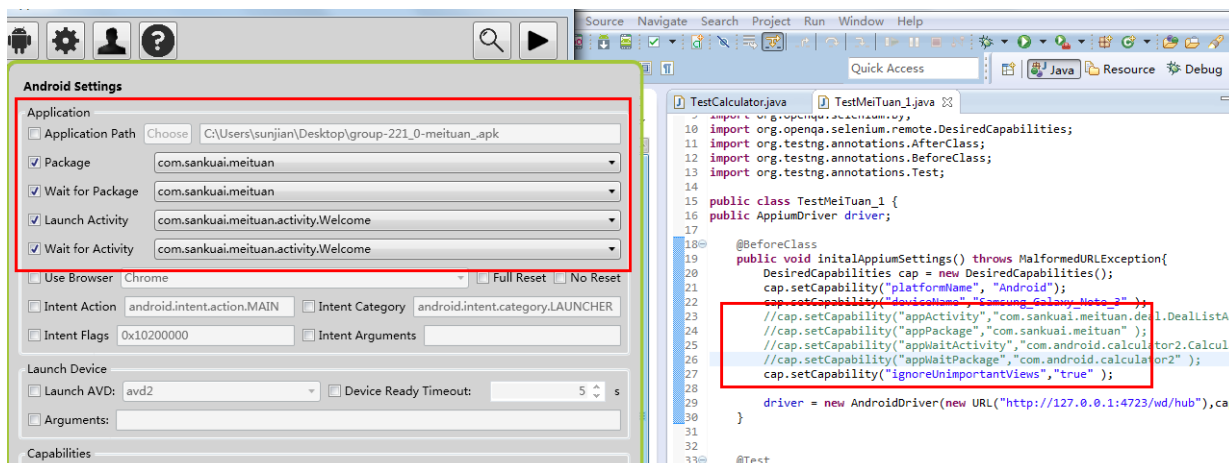
大家可以注意到，我们可以使用id来定位这个链接，但是这个会有多个，如果我们不使用xpath如何定位呢？

这里有点变化，我们在代码initialAppiumSettings中会有减少一些。

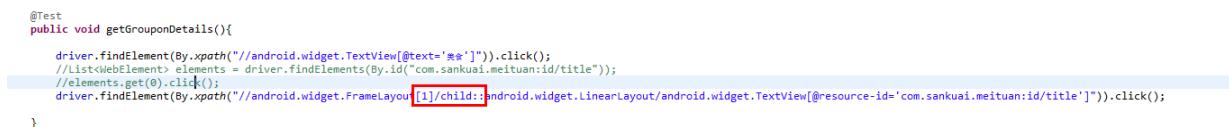
软件测试互动交流群：152015953

几点重要说明:

- 1)当你已经安装上了apk后application path可以不用选择,如果选择上了,appium会自动会将你的apk卸载后安装你选择的版本,注意版本一致性。
- 2)选择了application path和不选则的情况需要选择对应的launch activity和wait for activity因为很多情况下apk会有缓存策略。在不选择app path的情况下的wait for activity可以不会出现非缓存下的activity,如果不慎选择就会导致运行失败。
- 3)对应的我们的代码中可以去掉这几项,因为下拉列表中为我们自动抓取对应的activity。



通过上面的List集合类我们可以访问第一个元素;这种方式不好的地方是不方便管理对象;
下面我们通过xpath去实现:



动态元素的处理方式



大家可以看到上面美食,小吃快餐;
现在如果要求一个xpath的对应的变量可以访问到这里两个元素,那么我们怎么办呢?
大家可以寻找规律:
这两个元素的xpath如下

```
//android.widget.TextView[@text='美食']  
//android.widget.TextView[@text='小吃快餐']
```

这里其实就是替换掉text值中的内容,其他不变java中的String.format可以实现

大家可以看一下以下的写法,当然这个是一个比较推荐的推荐如果你想按照自己的方式那是一种鼓舞人的动力:
这里的小吃快餐可以替换为上面各种的。

```

@Test(description="动态元素")
public void getdynamicElements(){
    Action au = new Action(driver);

    au.click(By.xpath(homepage.getElement(homepage.category, "小吃快餐")));
    au.click(By.xpath(listitems.firstitem));
}

```

2)常用手机动作的使用swipe,hidekeyboard,sendkyes , tap等

上下滑动:

```

/**
 *
 * @param startElement
 * @param endElement
 * @param Up 是否向上滑动
 */
public void swipe(By startElement,By endElement,boolean Up){
    Point beginLoc;
    WebElement start = driver.findElement(startElement);
    beginLoc = start.getLocation();
    WebElement end;
    driver.manage().timeouts().implicitlyWait(1000, TimeUnit.MICROSECONDS);

    while(true){
        try{
            end = driver.findElement(endElement);
            break;
        }catch(Exception e){
            if(Up){
                driver.swipe(beginLoc.getX(), beginLoc.getY(), beginLoc.getX()+30, beginLoc.getY()+90, 4000);
            }else{
                driver.swipe(beginLoc.getX(), beginLoc.getY(), beginLoc.getX()-30, beginLoc.getY()-90, 4000);
            }
        }
    }

    driver.manage().timeouts().implicitlyWait(5000, TimeUnit.MILLISECONDS);
}

```

当target出现后就执行break;否则异常做滑动处理

点击：在webdriver时代，都是通过click来做，当然你也可以沿用webdriver的click方式，但偶尔情况会出现点击不准确的情况这就需要mobile自带的tap
tap一般会点击到元素的中央:

你可以将click的地方都换成tap，注意这里需要释放加入perform java中的build的设计模式使然；

```

public void tap(By by){
    au.tap(driver.findElement(by)).release().perform();
}

```

左右拖动：



大家看一下上面的可能需要一些拖动才能进入生活服务类

getY+0表示Y坐标没有动作。

```
public void categorySwipe(By source){
    WebElement start = driver.findElement(source);
    Point beginLoc = start.getLocation();

    driver.swipe(beginLoc.getX(), beginLoc.getY(), beginLoc.getX()-700, beginLoc.getY(), 8000);
}
```

输入值：有时候需要对输入框中清除后才能输入有效值，clear这个方法在appium中不支持可以通过下面的方式设置：

```
public void setValue(By by,String value){
    this.click(by);
    int size = driver.findElement(by).getText().length();
    driver.sendKeyEvent(123); //光标定位在最后
    for(int i =0;i<size;i++){
        driver.sendKeyEvent(AndroidKeyCode.BACKSPACE);
    }

    driver.findElement(by).sendKeys(value);
}
```

大家试试做一个登录的练习，这里有个输入的提示，然后输入完后光标无法定位到密码框，该怎么办呢？



```
@Test(description="login")
public void login() throws InterruptedException{
    au.tap(By.xpath(homepage.mine));
    au.tap(By.id(homepage.login));
    au.setValue(By.id(loginpage.username), "Shanghai_Terry");
    au.tap(By.id(loginpage.login));
    au.setValue(By.id(loginpage.password), "admin123");
    au.tap(By.id(loginpage.login));
}
```

在我们可以实现对native app的操作后，接下来便是我们设置处理数据，断言的时候了
数据的参数化，这里使用的方式是csv大家亦可以使用其他的方式，不过这种方式比较快捷方便。

```
5
6
7 @BeforeSuite
8 public void AndroidSettings() throws MalformedURLException{
9     TestData td = new TestData(System.getProperty("user.dir")+"\\src\\main\\resources\\globalsettings.csv");
10    DesiredCapabilities cap = new DesiredCapabilities();
11    cap.setCapability("platformName", "Android");
12    cap.setCapability("deviceName", td.getTestData("deviceName", "tc1"));
13    cap.setCapability("ignoreUnimportantViews", td.getTestData("ignoreUnimportantViews", "tc1"));
14    cap.setCapability("newCommandTimeout", td.getTestData("newCommandTimeout", "tc1"));
15
16    driver = new AndroidDriver(new URL("http://"+td.getTestData("server", "tc1")+"/wd/hub"), cap);
17    driver.manage().timeouts().implicitlyWait(Long.valueOf(td.getTestData("timeout", "tc1")), TimeUnit.MILLISECONDS);
18    au = new Action(driver);
19
20
21    homepage = new HomePage(driver);
22    listitemspage = new ListItemsPage(driver);
23    loginpage = new LoginPage(driver);
24 }
25
26
27
28
```

断言:一个case执行完了不等于一个case是成功的，只有加入断言才能判断是否执行结束或者成功。

```
81
82
83 @Test(description="loginwithdatadriver")
84 public void testDataDriver(){
85     au.tap(By.xpath(homepage.mine));
86
87     if(au.isElementPresent(By.id(homepage.balance))){
88         au.tap(By.id(homepage.balance));
89         au.tap(By.id(homepage.myaccountpage.exit));
90         au.tap(By.xpath(homepage.myaccountpage.logout));
91     }
92     au.tap(By.id(homepage.login));
93     au.setValue(By.id(loginpage.username), td.getTestData("username", "tc1"));
94     au.tap(By.id(loginpage.login));
95     au.setValue(By.id(loginpage.password), td.getTestData("password", "tc1"));
96     au.tap(By.id(loginpage.login));
97     Assert.assertEquals(au.isElementPresent(By.id(homepage.balance)), true);
98
99 }
100
101
102
103 @AfterClass
104 public void realse(){
105     driver.quit();
106 }
```

All suites

Default suite

Info

- C:\Users\sunjian\AppData\Local\Temp\testng-eclipse-1597238325\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 1 method, 1 passed
- Passed methods (hide)
- testDataDriver

lesson2. TestMeiTuan_1

testDataDriver (loginwithdatadriver)

