全程87appium第二次

回顾：

0 appium=selenium+uiautomator

1脚本 Capabilities 信息 +API表

**娶对象 API 积累**

from XXX import 对象 基于对象 有对象使用对象~

|  |  |
| --- | --- |
| **对象名字** | **对象功能方法** |
| **webdriver** | Remote("http://localhost:4723/wd/hub",capabilities) server  # 参数1： appium server地址  # 参数2： 字典 capabilities 包含 被测试手机 包名活动 apk 文件地址  {  "platformName": "平台名称",  "platformVersion": "平台版本",  "deviceName": "设备名称",  "app": apk包路径",  "appPackage": "包名",  "appActivity": "活动名",  "noReset":是否全新方式打开  }  补充 包名  aapt dump badging 包名活动名  adb devices 设备名称 |
|  | is\_app\_installed("包名") 判断包存在  remove\_app("包名") 卸载包 |
|  | **滑动API 三个**  **swipe(开始坐标x1，y1，结束x2，y2，时间)**  **scroll（元素1，元素2，时间） 慢慢划**  **drag\_and\_drop（元素1，元素2） 猛划 部分手机型号 不兼容**  **get\_window\_size() 计算出屏幕宽和高** |

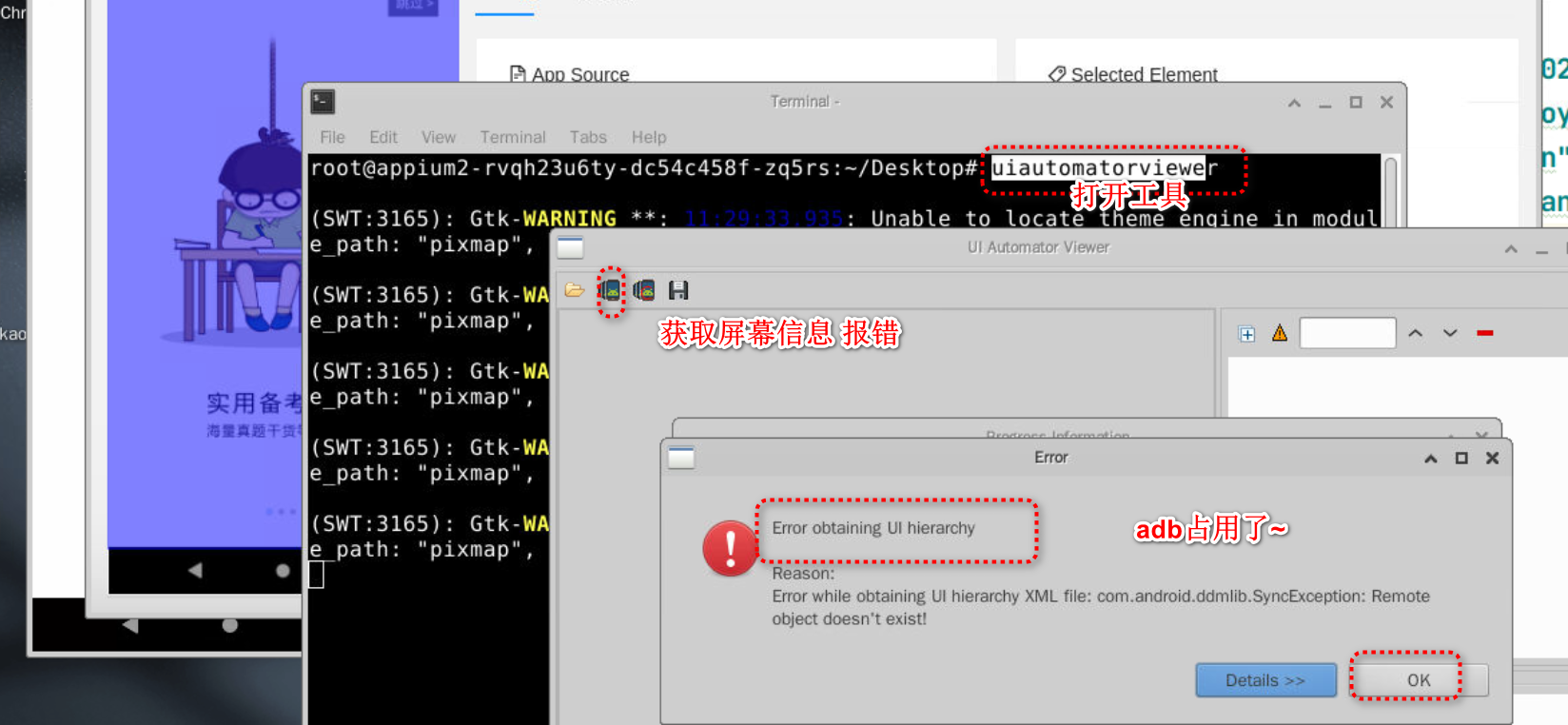
2 工具

Appium\_inspector 外往内查看

UIautomatorviewer 从内往外看

3 出错积累

adb抢占问题



### 内容1： UI自动化 （id定位）

例子 写一个考研帮登录脚本

|  |
| --- |
| # 1. capability 信息 连接appium server  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  jw\_capabilities={  "platformName": "Android",  "platformVersion": "9",  "deviceName": "127.0.0.1:10027",  "app": "/home/appium/apps/kaoyan3.1.0.apk",  "appPackage": "com.tal.kaoyan",  "appActivity": "com.tal.kaoyan.ui.activity.SplashActivity",  "noReset":False  }  #保险不乱码～  jw\_capabilities['unicodekeyboard']=True  jw\_capabilities['resetkeyboard']=True  # 参数1： appium server地址  # 参数2： 字典 capabilities 包含 被测试手机 包名活动 apk 文件地址  jwdriver=webdriver.Remote("http://localhost:4723/wd/hub",jw\_capabilities)  #2 获取元素定位信息 find\_element(By.id)  # 取消id android:id/button2  # 跳过id com.tal.kaoyan:id/tv\_skip  # ---提示你的基础库selenium 新 不建议  #jwdriver.find\_element\_by\_id('android:id/button2')  # 定位方式 定位信息  sleep(2)  # 取消更新按钮 可能出现  try:  cancelBtn=jwdriver.find\_element(AppiumBy.ID,'android:id/button2')  except:  print('主人没有取消更新～')  else:  print('主人，找到更新按钮')  cancelBtn.click()  finally:  print('是否更新处理完毕')  sleep(2)  #1这个代码 改成try形式  try:  skipBtn=jwdriver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/tv\_skip')  except:  print('没有找到跳过按钮')  else:  print('找到跳过按钮')  skipBtn.click()  sleep(2)  # 用户名 com.tal.kaoyan:id/login\_email\_edittext 神龙大侠999  jwdriver.find\_element(AppiumBy.ID,'com.tal.kaoyan:id/login\_email\_edittext').send\_keys('神龙大侠999')  # 密码 com.tal.kaoyan:id/login\_password\_edittext ksqb0177375  sleep(2)  jwdriver.find\_element(AppiumBy.ID,'com.tal.kaoyan:id/login\_password\_edittext').send\_keys('ksqb0177375')  # 按钮 com.tal.kaoyan:id/login\_login\_btn  jwdriver.find\_element(AppiumBy.ID,'com.tal.kaoyan:id/login\_login\_btn').click()  #3 对元素操作 点击 sendkeys click |

**小结 1:**

**代码优化1 如果出现中文**

|  |
| --- |
| #保险不乱码～  jw\_capabilities['unicodekeyboard']=True  jw\_capabilities['resetkeyboard']=True |

**代码优化2 手机上代码特点**

**需求： 手机上元素 随机出现~ 比如是否更新 比如过节广告XXX随机 喜好 随机？**

**出现随机产生元素 导致代码报错 无法执行？**

**解决： python语法 try except 降低代码的风险提升 健壮性~**

**try:#试一试**

**可能报错的信息**

**比如 查找元素 元素随机出现**

**except: # 接收运行时错误进一步处理 确保代码 继续运行**

**# 当try中代码报错时候 执行本分支~**

**可选else:**

**# # 当try中代码不报错时候 执行本分支~ 互斥**

**可选finally:**

**# 不管代码是否报错 都会执行本分支~**

### 内容2：元素定位

回顾：

手机

1 id定位

2 列表定位

id可以重复的 所有相同id元素一起挑选

3 text 文本信息方式 单独使用

find\_element\_by\_android\_uiautomator（'new UiSelector().text(文本信息)'）

4 class定位

find\_element(AppiumBy.CLASS\_NAME,'class元素定位信息')

5 xpath

appium 不处理 绝对路径

能处理相对路径

//\*[@属性=属性值]

find\_element(AppiumBy.XPATH,'//\*[@text="注册"]').click()

6 方式 android\_automator定位 （google 官方）

6.1 id定位

6.2 class定位

find\_element\_by\_android\_uiautomator('resouceId("id")')

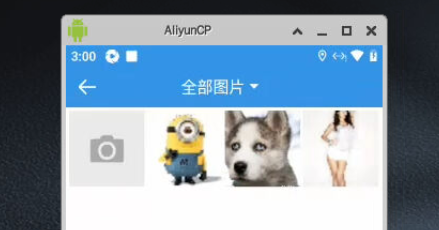
.find\_element\_by\_android\_uiautomator('className("class信息")')

6.3 父子定位和兄弟定位（提升超找元素速度）

**例子： 覆盖上面定位 即可 -注册脚本**

app 上传图片

|  |
| --- |
| adb push /home/appium/pics/\* /sdcard/DCIM/  adb shell am broadcast -a android.intent.action.MEDIA\_SCANNER\_SCAN\_FILE -d file:///sdcard/DCIM/xiaohuangren.jpg  adb shell am broadcast -a android.intent.action.MEDIA\_SCANNER\_SCAN\_FILE -d file:///sdcard/DCIM/hashiqi.jpg  adb shell am broadcast -a android.intent.action.MEDIA\_SCANNER\_SCAN\_FILE -d file:///sdcard/DCIM/meinv.jpg |





|  |
| --- |
| # 1. capability 信息 连接appium server  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  jw\_capabilities = {  "platformName": "Android",  "platformVersion": "9",  "deviceName": "127.0.0.1:10027",  "app": "/home/appium/apps/kaoyan3.1.0.apk",  "appPackage": "com.tal.kaoyan",  "appActivity": "com.tal.kaoyan.ui.activity.SplashActivity",  "noReset": False  }  # 保险不乱码～  jw\_capabilities['unicodekeyboard'] = True  jw\_capabilities['resetkeyboard'] = True  # 参数1： appium server地址  # 参数2： 字典 capabilities 包含 被测试手机 包名活动 apk 文件地址  jwdriver = webdriver.Remote("http://localhost:4723/wd/hub", jw\_capabilities)  sleep(2)  # 取消更新按钮 可能出现  try:  cancelBtn = jwdriver.find\_element(AppiumBy.ID, 'android:id/button2')  except:  print('主人没有取消更新～')  else:  print('主人，找到更新按钮')  cancelBtn.click()  finally:  print('是否更新处理完毕')  sleep(2)  # 1这个代码 改成try形式  try:  skipBtn = jwdriver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/tv\_skip')  except:  print('没有找到跳过按钮')  else:  print('找到跳过按钮')  skipBtn.click()  sleep(2)  # 注册：  #id  #jwdriver.find\_element\_by\_id('com.tal.kaoyan:id/login\_register\_text')  # class 定位  #jwdriver.find\_element(AppiumBy.CLASS\_NAME,'android.widget.TextView')  # xpath定位只能相对路径 //\*[@属性名字=属性的值]  jwdriver.find\_element(AppiumBy.XPATH,'//\*[@text="注册"]').click()  #添加头像  #需求 id 定位 速度慢～  # 能不能找一个区域 再从区域元素 慢 接口执行快～  #解决： 兄弟定位 父子定位 uiautomator\_android  #父子定位  # 父元素 找儿子.childSelector  # son='resourceId("com.tal.kaoyan:id/activity\_register\_parentlayout").childSelector(className("android.widget.ImageView"))'  # jwdriver.find\_element(AppiumBy.ANDROID\_UIAUTOMATOR,son)  # 兄弟定位 到父母找兄弟fromParent  brother='resourceId("com.tal.kaoyan:id/activity\_register\_username\_edittext").fromParent(className("android.widget.ImageView"))'  jwdriver.find\_element(AppiumBy.ANDROID\_UIAUTOMATOR,brother).click()  sleep(3)  # 列表定位 ===》id重复的时候进行使用～  # 所有同名元素信息 存放到了列表～  jieguo=jwdriver.find\_elements(AppiumBy.ID, 'com.tal.kaoyan:id/item\_image')  print(type(jieguo))  print(len(jieguo))  print(jieguo)  jieguo[2].click()#点击狗头 |

小结：例子涉及各种定位方式 --只是为了覆盖各种定位API

实际工作中 id +xpath +列表定位

### 内容3： 手势操作----（appium ）

手势：

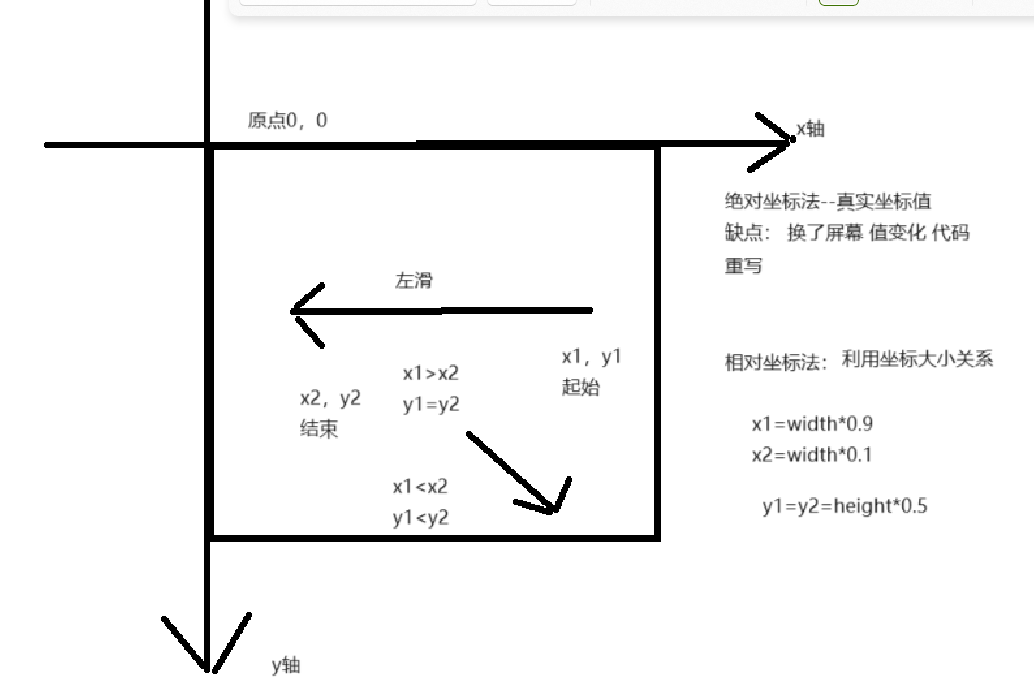
1拖动 滑动 九宫格

2 多指手势 图片放大缩小

### 例子1 滑动 ---swipe

android手机坐标系~~

-----处理坐标？



例子

|  |
| --- |
| # 1. capability 信息 连接appium server  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  jw\_capabilities = {  "platformName": "Android",  "platformVersion": "9",  "deviceName": "127.0.0.1:10027",  "app": "/home/appium/apps/kaoyan3.1.0.apk",  "appPackage": "com.tal.kaoyan",  "appActivity": "com.tal.kaoyan.ui.activity.SplashActivity",  "noReset": False  }  # 保险不乱码～  jw\_capabilities['unicodekeyboard'] = True  jw\_capabilities['resetkeyboard'] = True  # 参数1： appium server地址  # 参数2： 字典 capabilities 包含 被测试手机 包名活动 apk 文件地址  jwdriver = webdriver.Remote("http://localhost:4723/wd/hub", jw\_capabilities)  sleep(2)  # 取消更新按钮 可能出现  try:  cancelBtn = jwdriver.find\_element(AppiumBy.ID, 'android:id/button2')  except:  print('主人没有取消更新～')  else:  print('主人，找到更新按钮')  cancelBtn.click()  finally:  print('是否更新处理完毕')  sleep(2)  #1算出屏幕宽和高  jwsize=jwdriver.get\_window\_size()  width=jwsize['width']  height=jwsize['height']  #2 左滑相对坐标法大小关系  x1=width\*0.9  y1=height\*0.5  x2=width\*0.1  y2=y1  #3交给swipe运行  for i in range(2):  # 开始 结束  jwdriver.swipe(x1,y1,x2,y2,2000)# 毫秒  sleep(3) |

预告：明天继续api