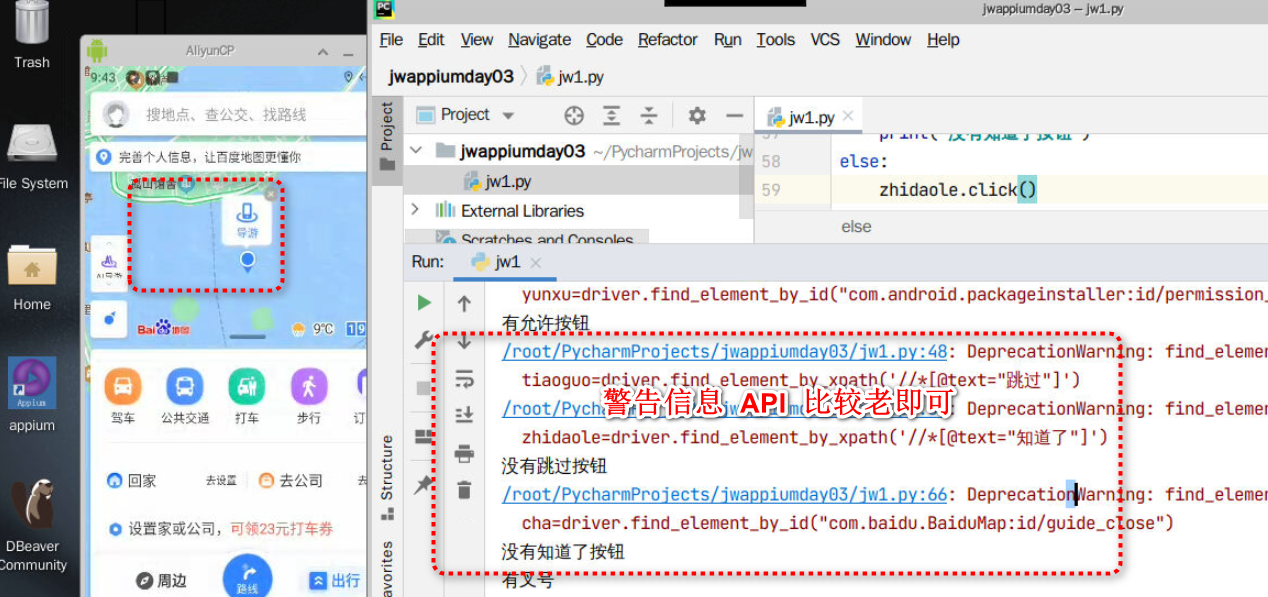
全程87期appium 第四次

回顾： API 表

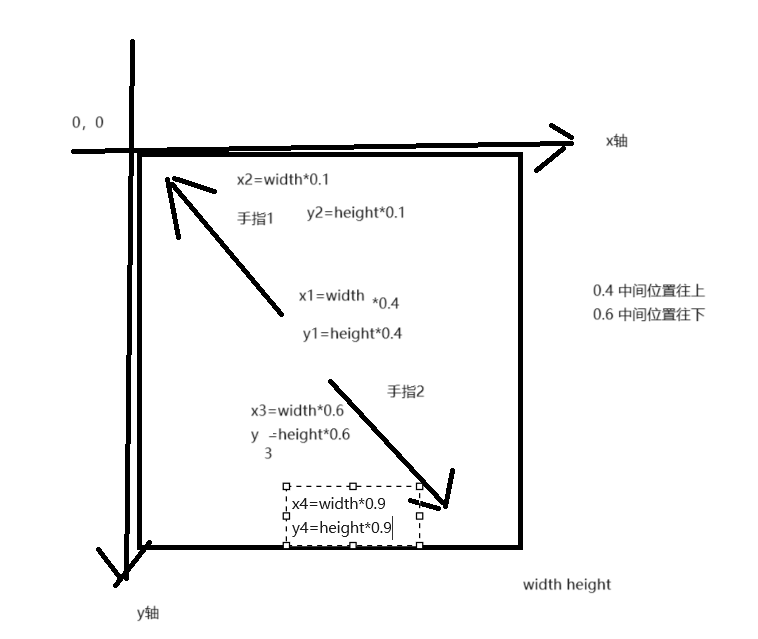
|  |  |
| --- | --- |
| **对象名字** | **对象功能方法** |
| **webdriver** | Remote("http://localhost:4723/wd/hub",capabilities) server  # 参数1： appium server地址  # 参数2： 字典 capabilities 包含 被测试手机 包名活动 apk 文件地址  {  "platformName": "平台名称",  "platformVersion": "平台版本",  "deviceName": "设备名称",  "app": apk包路径",  "appPackage": "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** |
| **TouchAction(webdriver) 新对象 单指对象** | **press(x=x坐标，y=y坐标)按下**  **longpress(坐标，时间)长按**  **release() 松开**  **move\_to（坐标）移动**  **wait(毫秒) 等待时间**  **以上API 形成运动轨迹**  **perform() 执行轨迹** |
| **m\_action=MultiAction(webdriver)对象** | **思路：**  **1 action1=TouchAction 手指1+轨迹 不执行**  **2 action2=TouchAction 手指2 + 轨迹 不执行**  **方法1接收所有手指的轨迹**  **m\_action.add(action1,action2)**  **方法2 执行轨迹**  **m\_action.perform() 执行所有手指轨迹** |
| **webdriver** | **升级Capability信息 字典**  **1**  **desired\_caps['automationName']='uiautomator2'**  **2 麻烦：**  **元素信息 Inspector工具 同步界面 耗时**  **toast 问题一闪而过**  **可能元素信息捕获不到 id可能没有**  **toast 采用xpath 文本定位**  **//\*[@text=文本信息]**  **//\*[contains(@属性，属性值)]**  **//正则相对路径 \* 任意标签** |
|  | **close\_app() 关闭app**  **launch\_app() 启动app**  **background\_app(时间)**  **app后台运行 当满足时间app自动出现~**  **冷启动**  **close\_app() 再launch\_app()**  **热启动**  **background\_app(时间)**  **切换app**  **star\_activity(切换app的包名，活动名)** |

例子 准备代码 -----》 Capability 包名和活动名 -----》放大图片

|  |
| --- |
| from time import sleep  from appium import webdriver  from appium.webdriver.common.multi\_action import MultiAction  from appium.webdriver.common.touch\_action import TouchAction  desired\_caps = {}  desired\_caps['platformName'] = 'Android'  desired\_caps['platformVersion'] = '9'  desired\_caps['deviceName'] = '127.0.0.1:10027'  # aapt  desired\_caps['appPackage'] = 'com.baidu.BaiduMap'  desired\_caps['appActivity'] = 'com.baidu.baidumaps.WelcomeScreen'  # 首次方式打开  desired\_caps['noReset'] = False  desired\_caps['app'] = r'/home/appium/apps/baidumap.apk'  # 中文问题处理  desired\_caps['unicodeKeyboard'] = True  desired\_caps['resetKeyboard'] = True  # 2.启动app(包含安装过程)  driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)  driver.implicitly\_wait(8)  # 3.点‘同意’按钮  driver.find\_element\_by\_id('com.baidu.BaiduMap:id/ok\_btn').click()  try:  yunxu=driver.find\_element\_by\_id("com.android.packageinstaller:id/permission\_allow\_button")  except:  print("没有允许按钮")  else:  print("有允许按钮")  yunxu.click()  # 点击立即体验  try:  lijitiyan=driver.find\_element\_by\_id("com.baidu.BaiduMap:id/btn\_enter\_map")  except:  print("没有立即体验")  else:  print("有立即体验")  lijitiyan.click()  try:  yunxu=driver.find\_element\_by\_id("com.android.packageinstaller:id/permission\_allow\_button")  except:  print("没有允许按钮")  else:  print("有允许按钮")  yunxu.click()  # 4.点‘跳过’  try:  tiaoguo=driver.find\_element\_by\_xpath('//\*[@text="跳过"]')  except:  print("没有跳过按钮")  else:  tiaoguo.click()  # 5.点‘知道了’  try:  zhidaole=driver.find\_element\_by\_xpath('//\*[@text="知道了"]')  except:  print("没有知道了按钮")  else:  zhidaole.click()  # 关闭随机生成的广告  #com.baidu.BaiduMap:id/guide\_close  try:  cha=driver.find\_element\_by\_id("com.baidu.BaiduMap:id/guide\_close")  except:  print("没有叉号")  else:  print("有叉号")  cha.click()  # 汉语思路 ---》查API  # 1 计算 屏幕的宽和高  jwsize=driver.get\_window\_size()  width=jwsize['width']  height=jwsize['height']  #2.1 手指1坐标 （越来越小）  x1=width\*0.4  y1=height\*0.4  x2=width\*0.1  y2=height\*0.1  #2.1 手指1轨迹 x y  action1=TouchAction(driver).press(x=x1,y=y1).wait(1000).move\_to(x=x2,y=y2).wait(1000).release()  #3.1 手指2坐标（越来越大）  #3.2 手指2轨迹  x3=width\*0.6  y3=height\*0.6  x4=width\*0.9  y4=height\*0.9  #2.1 手指1轨迹 x y  action2=TouchAction(driver).press(x=x3,y=y3).wait(1000).move\_to(x=x4,y=y4).wait(1000).release()  # 4 两根手指同时运行  for i in range(2):  jw\_m\_action=MultiAction(driver)  jw\_m\_action.add(action1,action2)  jw\_m\_action.perform()  sleep(3) |



多指放大API -----



小结 1API整理 2 思路

# 内容1： 其他API

例子1： 吐司(toast) 定位 ---测试为什么定位toast？

toast 概念 短暂出现消息提示框~

场景 ：考研帮登录

登录成功-----》python： 出现登录成功后的信息（我）

登录失败 （无效等价类用例）---》需要捕获是否出现 toast 作为 测试用例是否通过的标准~

|  |
| --- |
| # 1.获取appium capabilities  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  desired\_caps = {}  desired\_caps['platformName'] = 'Android'  desired\_caps['platformVersion'] = '9'  desired\_caps['deviceName'] = '127.0.0.1:10027'  desired\_caps['appPackage'] = 'com.tal.kaoyan'  desired\_caps['appActivity'] = 'com.tal.kaoyan.ui.activity.SplashActivity'  desired\_caps['noReset'] = False  desired\_caps['app'] = r'/home/appium/apps/kaoyan3.1.0.apk'  #toast的配置  desired\_caps['automationName'] = 'uiautomator2'  # 中文问题处理  desired\_caps['unicodeKeyboard'] = True  desired\_caps['resetKeyboard'] = True  # 2.启动app(包含安装过程)  driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)  driver.implicitly\_wait(10)  # 3.点击‘取消’按钮  cancel\_btn = driver.find\_element\_by\_id('android:id/button2')  cancel\_btn.click()  # 4.点击‘跳过’按钮  skip\_btn = driver.find\_element\_by\_id('com.tal.kaoyan:id/tv\_skip')  skip\_btn.click()  # 用户名密码 不能输入～～用例  # 5.点击‘登录’按钮  login\_btn = driver.find\_element\_by\_id('com.tal.kaoyan:id/login\_login\_btn')  login\_btn.click()  #toast 定位  try:  jw\_toast=driver.find\_element(AppiumBy.XPATH,'//\*[@text="邮箱帐户密码不能为空"]')#预期结果  except:  print('用例不通过 失败～')  else:  print('用例通过') |

### 例子2： 切换app api

**场景 考研帮app（被测） ----随手记app**

**测试点 app中途退出 app是否恢复**

|  |
| --- |
| #1启动考研帮 2关闭 3 重新打开 app 4切换app随手记 5 后台运行 6 退出appium  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  desired\_caps = {}  desired\_caps['platformName'] = 'Android'  desired\_caps['platformVersion'] = '9'  desired\_caps['deviceName'] = '127.0.0.1:10027'  desired\_caps['appPackage'] = 'com.tal.kaoyan'  desired\_caps['appActivity'] = 'com.tal.kaoyan.ui.activity.SplashActivity'  desired\_caps['noReset'] = False  desired\_caps['app'] = r'/home/appium/apps/kaoyan3.1.0.apk'  #toast的配置  #desired\_caps['automationName'] = 'uiautomator2'  # 中文问题处理  desired\_caps['unicodeKeyboard'] = True  desired\_caps['resetKeyboard'] = True  # 2.启动app(包含安装过程)  driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)  #2关闭  sleep(2)  driver.close\_app()  # 3 重新打开 app  sleep(2)  driver.launch\_app() # 开的 desired\_caps app当前考研帮  # 4切换app随手记  sleep(2)  driver.start\_activity('com.mymoney','com.mymoney.biz.splash.SplashScreenActivity')  # 5后台运行  sleep(2)  driver.background\_app(3) #3秒后 app 重新返回到前台 #手机兼容问题～  # 6 退出appium  #sleep(3)  driver.quit() |

恭喜 ： appium 常见API 结束 ~

[Write a Test (Python) - Appium Documentation](https://appium.io/docs/en/latest/quickstart/test-py/)

1 新的功能 --三指截屏 套 API （问人工智能） +发现API没有实现（手机兼容）手工做

====》和编程能力（appium 也在不断的更新中）

任何工具 使用本质 ----》套API

实现了自动化

代码 ： 一劳永逸

**2还得测试**

### 内容2： 自动化测试小框架（入门框架）下午

**需求1：**

**测试 app或者手机 更换的 Capability 信息频繁更换**

**需求2：**

**组织测试用例 用例中 进行测试断言 完成用例是否通过**

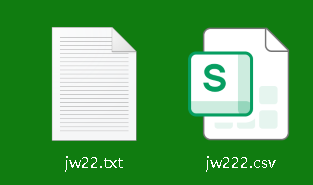
**需求3：**

**用例 设计到不同的用例数据 （CSV用例数据 ）**

**excel文件无法跨平台 上传到linux 没法用了**

**csv 保留了 编辑功能 跨平台 文件使用~**

**zhangsan1 ~zhangsan10**



**需求4 ：图形化报告**

**需求5： 分工 （一拨人测试业务上元素定位 另一拨人组织用例）**