全程87 appium 第三次

回顾：---API表

**娶对象 API 积累---appium 应用软件**

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** |
| **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** |

2 工具

Appium\_inspector 外往内查看

UIautomatorviewer 从内往外看

小白汉语做法 + 根据API

例子： 考研帮滚动功能

|  |
| --- |
| # 第一步：连接手机  #第二步 登录  # 1.获取desired capabilities信息  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  desired\_caps = {}  desired\_caps['platformName'] = 'Android'  desired\_caps['app'] = '/home/appium/apps/kaoyan3.1.0.apk'  desired\_caps['appPackage'] = 'com.tal.kaoyan'  desired\_caps['appActivity'] = 'com.tal.kaoyan.ui.activity.SplashActivity'  desired\_caps['noReset'] = False  # desired\_caps['automationName'] = 'uiautomator2'  # 2.启动app（包含安装过程）  driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)  driver.implicitly\_wait(15)  # 3.点击‘取消’按钮  sleep(2)  driver.find\_element(AppiumBy.ID, 'android:id/button2').click()  # 4.点击‘跳过’按钮  sleep(2)  driver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/tv\_skip').click()  # 5.输入用户名和密码  # 用户名  sleep(2)  driver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/login\_email\_edittext').send\_keys('神龙大侠999')  # 密码  sleep(2)  driver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/login\_password\_edittext').send\_keys('ksqb0177375')  # 6.点击‘登录’按钮  sleep(2)  driver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/login\_login\_btn').click()  # 7.验证是否登录成功  # 帐号下线提示  try:  element1 = driver.find\_element(AppiumBy.ID, "com.tal.kaoyan:id/tip\_commit")  except:  print("没有账号下线提示")  else:  element1.click()  # 增加新的广告处理  try:  element3 = driver.find\_element(AppiumBy.ID, "com.tal.kaoyan:id/view\_wemedia\_cacel")  except:  print("没有弹出广告提示２")  else:  element3.click()  # 检测是否有我知道了按钮  try:  element2 = driver.find\_element(AppiumBy.ID, "com.tal.kaoyan:id/task\_no\_task")  except:  print("没有弹出广告提示")  else:  element2.click()  # 根据界面是否有‘我’判断登录是否成功  try:  # 寻找下方4个tab中的‘我’  driver.find\_element(AppiumBy.ID, "com.tal.kaoyan:id/mainactivity\_button\_mysefl")  except:  print("登录失败")  else:  print("登录成功")  # 8.点击‘论坛’按钮  sleep(2)  driver.find\_element(AppiumBy.ID, 'com.tal.kaoyan:id/mainactivity\_button\_forum').click()  # 9.向上滑动  sleep(2)  #第三步滚动  sleep(2)  # text 定位比 id更好 但是嫌弃长===》借助xpath属性表达方式 表达text定位  # 文字容易打错 =完全一样  # contains(@属性名字，”属性的值“) 包含  #driver.find\_element(AppiumBy.XPATH,'//\*[@text="专业课学长学姐招募！知识变现等你来"]')  # xpath 路径唯一可以  sleep(2)  # 起点  start=driver.find\_element(AppiumBy.XPATH,'//\*[contains(@text,"专业课学长学姐招募！知识变现等你来")]')  #结束点  end=driver.find\_element(AppiumBy.XPATH,'//\*[contains(@text,"【真题免费送】80000套真题陆续上线中，含专业课真题")]')  driver.scroll(start,end,2000)  sleep(2) |

小结：

1 xpath 定位

# contains(@属性名字，”属性的值“) 包含

2 scroll 元素1元素2 swipe 坐标 起始坐标 到结束坐标

### 内容2：单指手势

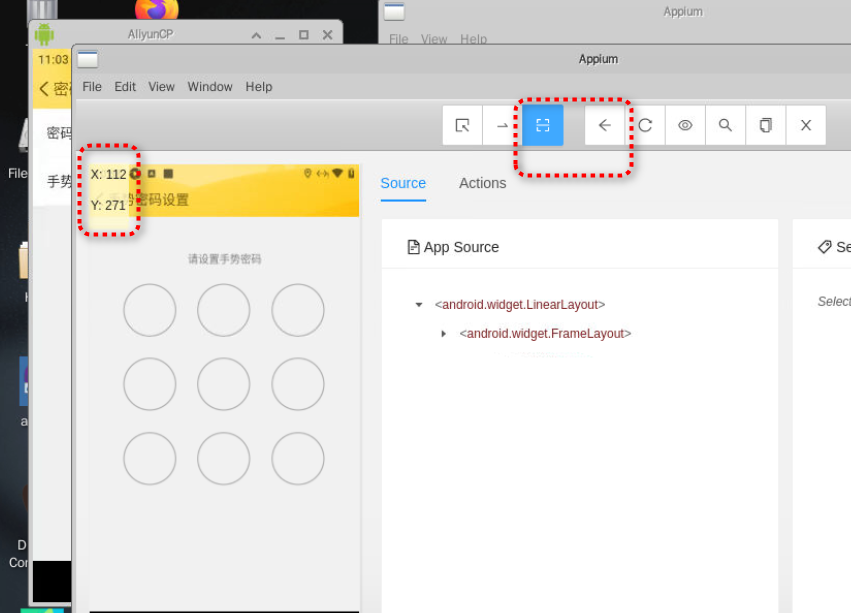
例子：mymoney app 九宫格 （绝对坐标法和相对坐标法整合方式）

**注意： app定位 找界面已经展示元素 元素一定展示**

上划

起始点 ： x1=x2

结束点： y1起始>y2 结束 越来越小 相对坐标法

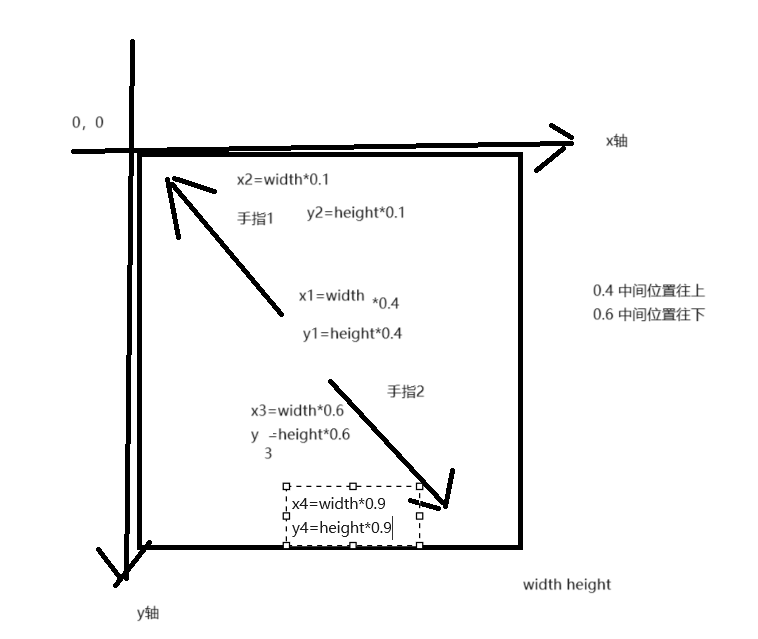


|  |
| --- |
| # 1.获取desired capabilities信息  from time import sleep  from appium import webdriver  from appium.webdriver.common.appiumby import AppiumBy  from appium.webdriver.common.touch\_action import TouchAction  desired\_caps = {}  desired\_caps['platformName'] = 'Android'  desired\_caps['platformVersion'] = '9'  desired\_caps['app'] = '/home/appium/apps/mymoney.apk'  desired\_caps['appPackage']='com.mymoney'  desired\_caps['appActivity'] = 'com.mymoney.biz.splash.SplashScreenActivity'  desired\_caps['noReset'] = False  # 2.启动app（包含安装过程）  driver = webdriver.Remote('http://localhost:4723/wd/hub', desired\_caps)  driver.implicitly\_wait(15)  # 3.允许媒体访问×2  sleep(2)  for i in range(2):  driver.find\_element(AppiumBy.ID,'com.android.packageinstaller:id/permission\_allow\_button').click()  sleep(2)  # 4.点击‘下一步’按钮×2  sleep(2)  for i in range(2):  driver.find\_element(AppiumBy.ID,'com.mymoney:id/next\_btn').click()  sleep(2)  # 5.点击‘开始随手记’按钮  sleep(2)  driver.find\_element(AppiumBy.ID,'com.mymoney:id/begin\_btn').click()  # 6.点击‘更多’按钮( 注意查找更多的时间是比较长的)  sleep(2)  driver.find\_element(AppiumBy.ID,'com.mymoney:id/nav\_setting\_btn').click()  # 7.向上滑动屏幕，点击‘高级’按钮  sleep(2)  size = driver.get\_window\_size()  width = size['width'] #宽  height = size['height'] # 高  #起始点  x1 = width\*0.5  y1 = height\*0.9  # 结束点 y小  x2 = x1  y2 = height\*0.1  driver.swipe(x1,y1,x2,y2,500)  sleep(2)  driver.find\_element(AppiumBy.ANDROID\_UIAUTOMATOR,'text("高级")').click()  # 8.点击‘密码与手势密码’按钮  sleep(2)  driver.find\_element(AppiumBy.ANDROID\_UIAUTOMATOR,'text("密码与手势密码")').click()  # 9.点击‘手势密码保护’按钮  sleep(2)  driver.find\_element(AppiumBy.ANDROID\_UIAUTOMATOR,'text("手势密码保护")').click()  # 10.设置手势×2  sleep(2)  # 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  #四个点  x1=115  y1=287  x2=408  y2=y1  x3=x2  y3=577  x4=x1  y4=y3 # 第三个点和第四个点关系～  # \ 折行  for i in range(2):  TouchAction(driver).press(x=x1,y=y1).wait(2000).\  move\_to(x=x2,y=y2).wait(2000).\  move\_to(x=x3,y=y3).wait(2000).\  move\_to(x=x4,y=y4).wait(2000).release().perform()  sleep(5) |

### 内容3： 多指手势（双指）

场景： 百度地图app 放大和缩小

分析： 绝对还是相对？根据大小关系



基础弱 照抄 基础 尝试一把多指 下周实现 （API+小框架）