

- 参照视频教程，cmd 中运行 `appium --session-override`

```
C:\Users\54251>appium --session-override
[Appium] Welcome to Appium v1.10.1
[Appium] Non-default server args:
[Appium]   sessionOverride: true
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

- （启动动模拟器 Android Emulator）

名称	修改日期	类型	大小
1.apk	2017/7/24 15:04	APK 文件	3,556 KB
appcrawler-2.4.0.jar	2019/2/25 13:37	Executable Jar File	129,387 KB

- 另一 cmd 中运行 `java -jar appcrawler-2.4.0.jar`

```
e:\test>java -jar appcrawler-2.4.0.jar

-----
AppCrawler 2.4.0 [霍格沃兹测试学院特别纪念版]
Appium 1.8.1 Java8 tested
app爬虫, 用于自动遍历测试. 支持Android和iOS, 支持真机和模拟器
项目地址: https://github.com/seveniruby/AppCrawler
移动测试技术交流: https://testerhome.com
联络作者: seveniruby@testerhome.com (思寒)
致谢: 晓光 泉龙 杨榕 恒温 mikezhou yaming116 沐木

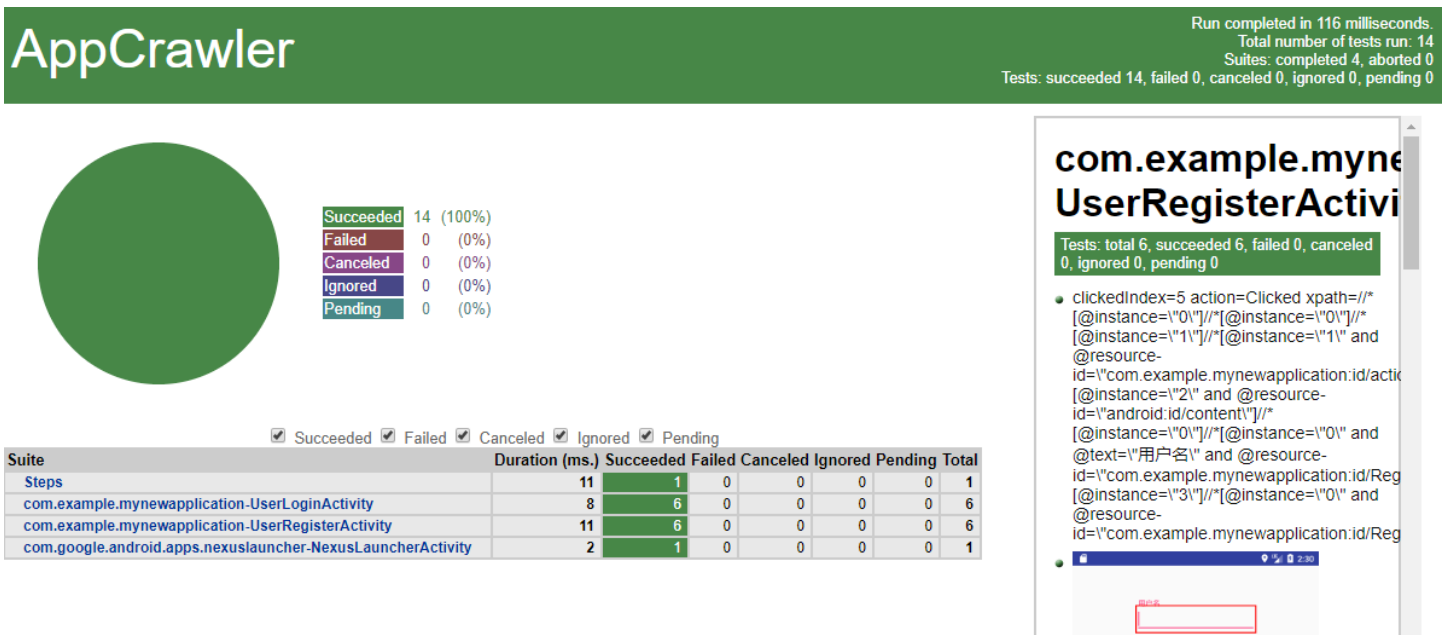
-----

Usage: appcrawler [options]

  -a, --app <value>      Android或者iOS的文件地址, 可以是网络地址
  -e, --encoding <value> set encoding, such as UTF-8 GBK
  -c, --conf <value>     配置文件地址
```

- 可以看到生成的截图等内容。

文档 (E:) > test > 20190225143011				
名称	修改日期	类型	大小	
css	2019/2/25 14:31	文件夹		
images	2019/2/25 14:31	文件夹		
js	2019/2/25 14:31	文件夹		
tmp	2019/2/25 14:31	文件夹		
0_com.example.mynewapplication-UserLoginActivity.tag=start.id=start.clicked.png	2019/2/25 14:30	PNG 文件	205 KB	
0_com.example.mynewapplication-UserLoginActivity.tag=start.id=start.dom	2019/2/25 14:30	DOM 文件	9 KB	
1_Steps.tag=.name=NOT_FOUND.click.png	2019/2/25 14:30	PNG 文件	205 KB	
1_Steps.tag=.name=NOT_FOUND.clicked.png	2019/2/25 14:30	PNG 文件	205 KB	
1_Steps.tag=.name=NOT_FOUND.dom	2019/2/25 14:30	DOM 文件	9 KB	
2_com.example.mynewapplication-UserLoginActivity.tag=EditText.0.depth=10.id=Login_UserName.click.png	2019/2/25 14:30	PNG 文件	236 KB	



- 问题：见下图。解决：将 appcrawler-2.1.3.jar 的改为 appcrawler-2.4.0.jar。

[illegible]

- 待做：测试配置文件.yml

视频笔记:

- ## ● Appium-android 常用命令

- adb devices、adb shell, aapt dump badging 1.apk、aapt list 1.apk (D:\Android\sdk\build-tools\26.0.0 中)

```
D:\Android\sdk\build-tools\26.0.0>aapt dump badging e:\test\1.apk
package: name='com.example.mynewapplication' versionCode='1' versionName='2.33' platformBuildVersionName='8.0.0'
sdkVersion:'15'
targetSdkVersion:'26'
uses-permission: name='android.permission.INTERNET'
uses-permission: name='android.permission.WRITE_EXTERNAL_STORAGE'
application-label:'?????????????'
application-label-af:'?????????????'
```

- adb logcat |grep ActivityManager

- D→debug, I→Info

- Inspector 使用

- 启动: deviceName、appPackage(通过 aapt dump badging 1.apk 查找 \ 使用 adb logcat |grep ActivityManager 并打开 app 后通过 cmp 查找)、platformName、appActivity (使用 adb logcat |grep ActivityManager 并打开 app 后通过 cmp 查找)
- Recorder、Search for element
- Appium 日志
 - (Custom Server 远程)
 - - info: [debug] [ADB] Running ‘/usr/local/opt/android-sdk/platform-tools/adb’ with args: [“-p”,5037,”-s”,”192.168.58.101:5555”,”shell”,”pm”,”list”,”packages”,”com.xueqiu.android”]

→ /usr/local/opt/android-sdk/platform-tools/adb -p 5037, -s ”192.168.58.101:5555” shell pm list packages com.xueqiu.android
 - - info: [debug] [ADB] Running ‘/usr/local/opt/android-sdk/platform-tools/adb’ with args: [“-p”,5037,”-s”,”192.168.58.101:5555”,”wait-for-device”]

→ /usr/local/opt/android-sdk/platform-tools/adb -p 5037, -s ”192.168.58.101:5555” wait-for-device
 - - info: [debug] [ADB] Running ‘/usr/local/opt/android-sdk/platform-tools/adb’ with args: [“-p”,5037,”-s”,”192.168.58.101:5555”,”shell”,”echo”,”ping”]
 - uninstall io.appium.setting、install /usr/.../1.apk、shell dumpsys package io.appiun.setting、shell ps (进程)
 - shell getprop ro.build.version.skd/ro.build.version.release/ro.product.model/ro.product.manufacturer、shell wm size、forward tcp:4724 tcp:4724 等
 - shell am force-stop io.appium.setttting/com.xueqiu.android、pm clear com.xueqiu.android
 - am (activity manager), pm (package manager)
 - 重点 push /usr/local/lib/.../bin//AppiumBootstrap.jar /data/local/tmp/
 - 重点 shell uiautomator runtest AppiumBootstrap.jar -c io.appium.android.bootstrap -e pkg com.xueqiu.android -e disableAndroidWatchers false -e acceptSslCerts false 从而支持自动化
 - 核心点 sheel am start -W -n com.xueqiu.android/.view.WelcomeActitivityAlias -S -a android.intent.action.MAIN -c android.intent.catagory.LAUNCHER -f 0x10200000 开始启动包
 - 以上是自动化的前置条件
 - [HTTP] <-- POST /wd/hub/session/d1601a25-.../elements

浏览器访问 127.0.0.1:4723/wd/hub/session/d1601a25-.../souce 和 Appium Desktop 中 App Source 一样
 - grep run appium.log、grep Running appium.log

● 自动化测试用例的编写

- {"platformName": "android","deviceName": "Google","appPackage": "io.appium.android.apis","appActivity": ".ApiDemos"}, 可加一项"app","e:\test\apiDemos.apk"或者在 cmd 使用 adb install e:\test\apiDemos.apk
- 安装某应用，提示 INSTALL_FAILED_NO_MATCHING_ABIS。是由于模拟器 ABI 为 x86。
- 延迟问题可使用 sleep(3)或 wait,(在 python 中)
- 框架改造和编写

1. 编写脚本：导入依赖、capabilities 设置（单独函数）、元素定位与操作 find+action、断言 assert。

测试步骤：定位、交互、断言
2. 改造参照 https://github.com/appium-boneyard/sample-code/blob/master/sample-code/examples/python/android_simple.py，包括 def setUp(self):、def tearDown(self):、def test_simple_actions(self):等
3. 定位：dom、dom 应用（节点和属性）和 xpath。

定位：id(resouce-id,accessibility id→content-desc,atring.xml id)、xpath、accessibityId:content-desc(一般用不到，但还是用了)等

```
curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/elements --data-binary '{"using":"id","value": "action_message"}' -H "Content-Type: application/json; charset=UTF-8" {"status":0, "value":[{"ELEMENT":"12"}],"sessionId":"d1601a25-..."}

curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/elements --data-binary '{"using":"xpath","value":"///*[@resource-id=\'com.xueqiu.android:id/action_message\']"}' -H "Content-Type: application/json; charset=UTF-8" {"status":0, "value":[{"ELEMENT":"12"}],"sessionId":"d1601a25-..."}
```

优先使用 find_element_by_id，id 不存在或重复时使用 xpath。find_element_by_xpath...需要把绝对路径改成相对("///*[@text=’推荐’]"), 如果重复可继续添加限制("///*[@text=’推荐’ and @resouce-id=’...’] ")。两种方法时间差别不大。

4. java 中的依赖，通过 github 中 sample-code 中的 pom.xml 或者参考右图。
5. 封装多种定位方法，右图。
6. driver.findElementById("com.example.mynewapplication:id/Register_UserName");

driver.findElementByXPath("//*[@contains(@text,'Search')]");

driver.findElementByXPath("//*[@test='Search']");

driver.findElementByAccessibilityId("Popup Menu").click();

```
public WebElement locate(String locator) {
    if (locator.matches("\\ \\V.*"))
    {
        return driver.findElementByXPath(locator);
    }
    else{return driver.findElementById(locator); }
}
driver.find_element_by_xpath("//*[@text='推荐']").click();
→locate("//*[@text='推荐']").click();
```

```
<dependencies>
<!--
https://mvnrepository.com/artifact/io.appium/java-client -->
<dependency>
<groupId>io.appium</groupId>
<artifactId>java-client</artifactId>
<version>6.1.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/junit/junit -->
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.12</version>
<scope>test</scope>
</dependency>
</dependencies>
```

● webdriver 协议

- http://127.0.0.1:4723/wd/hub/sessions

http://127.0.0.1:4723/wd/hub/session/d1601a25-.../source

http://127.0.0.1:4723/wd/hub/session/{session id}/element/{element id}/element
- curl -X POST http://127.0.0.1:4723/wd/hub/session/{session id}/elements --data-binary '{"using":"id","value":"com.xueqiu.android:id/tv_login_by_wx"}' -H "Content-Type: application/json; charet=UTF-8" -vv

→ {"statues":0, "value":[{"ELEMENT":"7"}],.....}
- curl -X GET http://127.0.0.1:4723/wd/hub/session/{session id}/element/7(为{emelent id})/text -H "Content-Type: application/json; charet=UTF-8" -vv

→ {"statues":0, "value":"微信登陆",.....}
- 点击请求: curl -X POST http://127.0.0.1:4723/wd/hub/session/{session id}/element/{emelent id}/click -H "Content-Type: application/json; charet=UTF-8" -vv

改造成函数: click(){curl -X POST http://127.0.0.1:4723/wd/hub/session/{session id}/element/\$1(第一个参数)/click -H "Content-Type: application/json; charet=UTF-8" -vv; } →click 7

● 调试分析方法

- Appium 的 log，所有请求和结果，desktop 或者 cmd 里。
- getPageSource，当前布局 dom 结构所有控件.xml 文件，右图。
- 脚本内调试，利用 xpath 获取空间所有元素，driver.findElementsByPath("//*[@*)")

```
try{
...
}finally{
System.out.println(driver.getPageSource());
}
```

```
try{
...
}finally{
for(AndroidElement e:driver.findElementsByXPath("//*[@*)")
{
System.out.println(e.getTagName());
System.out.println(e.getTagText());
///System.out.println(e.getAttribute("resource-id"));
//System.out.println(e.getAttribute("content-desc"));
}
}
```

toast 识别条件

- toast 在 ApiDemos 中 Views 的 Popup Menu 中有一个。
- 新的框架 aotumationName: uiautomator2。使用 xpath 查找，//*[@class='android.widget.Toast']和//*[contains(@text, "xxx")]

```
{"platformName": "android","deviceName": "Google","appPackage": "io.appium.android.apis","appActivity": ".ApiDemos","automationName":"uiautomator2"}
```

- 例如，while true;do sleep 0.5; curl http://localhost:8200/wd/hub/session/3e943631-f896-452c-9cd5-3063f3452792/elements --data-binary '{"strategy":"xpath","selector":"//*[@class=\"android.widget.Toast\"]","context":"","multiple:true}';echo ;done 可得到如下图所示。

```
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [ { "ELEMENT": "b28be9b5-19cb-4bbb-a05a-10777440af1b" } ] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [ { "ELEMENT": "ba280ed2-3785-40ed-962b-e3a8112683ec" } ] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [ { "ELEMENT": "29d2e8ff-bbbd-403a-8379-ea239a71fb63" } ] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [ { "ELEMENT": "85e02351-9b72-4c75-88a3-4729630cba13" } ] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [ { "ELEMENT": "74afd84e-62af-41b1-b0ab-c242c0cb7dd3" } ] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
{ "sessionId": "3e943631-f896-452c-9cd5-3063f3452792", "status": 0, "value": [] }
```

- Juint 中运行，提示 Error: Could not access the Package Manager. Is the system running?→模拟器死机了，重启。
- 录制比写代码快，不如录制。
- 代码在参照 10 测试\appium 自动化测试\appiumtest - findToast 中的 testToast1(), 可运行并得到 toast 的 text。改进，参照 testToast2()和 testToast3()。

等待

- 需等待控件加载，同时控件加载时位置会变动，是由其他控件还未加载出，需要等到位置处于稳定状态。
- 精确等待 Thread.sleep(1000);
- 全局隐式等待 driver.manage().timeouts().implicitlyWait(1,TimeUnit.SECONDS);
- 显式等待 WebDriverWait wait = new WebDriverWait(driver,1);
WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("xx")));

API

- 可封装为单独的函数
- driver.rotata(ScreenOrientation.LANDSCAPE);
driver.rotata(ScreenOrientation.PORTRAIT);
driver.navigate().back();
driver.openNotifications();
- System.out.println(driver.manage().logs().getAvailableLogTypes());
for(Object l: driver.manage().logs().get("logcat").getAll().toArray()){System.out.println(l);}
- System.out.println(driver.getSupportedPerformanceDataTypes());→[memoryinfo,cpuinfo,batteryinfo,networkinfo]
System.out.println(driver.getPerformanceData("io.appium.android.apis(是 packagename)","memoryinfo/cpuinfo /batteryinfo/networkinfo",10))

```
@Test
public void testPerformance() throws Exception {
System.out.println(driver.getPerformanceData("com.example.mynewapplication","memoryinfo",10));
System.out.println(driver.getPerformanceData("com.example.mynewapplication","cpuinfo",10));
}
```

xpath 定位

- https://www.freeformatter.com/xpath-tester.html 或者可以在 appium desktop 内通过放大镜测试
- driver.findElementByXPath("//*[@*)");
driver.findElementByXPath("//*[@@test='Search']");
driver.findElementByXPath("//*[@contains(@text,'Search')]");
driver.findElementByXPath("//*[@contains(@class,'Image') and @instance=3]"); instance 为同类型的序号
- //*[@contains(@text,'Search')] /ancestor::*//*[contains(name(), 'EditText')] 父类
//*[@clickable="true"] //android.widget.TextView[string-length(@text)>0 and string-length(@text)<20] 遍历可点击的指定长度范围内的文本菜单

webview 切换

- 自动向下滚动

```
public void testWebView() throws InterruptedException { //ApiDemos
driver.findElementByAccessibilityId("Views").click();
WebElement list = driver.findElement(By.id("android:id/list")); //not AndroidElement
MobileElement radioGroup = list.findElement(MobileBy
.AndroidUIAutomator("new UiScrollable(new UiSelector()).scrollIntoView("
+ "new UiSelector().text(\"WebView\");"));
}
```

待做

```
@Test
public void testWebView2() throws InterruptedException {
    driver.findElementByAccessibilityId( using: "Views").click();
    AndroidElement list = driver.findElement(By.id("android:id/list"));
    MobileElement webview = list
        .findElement(MobileBy
            .AndroidUIAutomator( uiAutomatorText: "new UiScrollable(new UiSelector()).scrollIntoView("
                + "new UiSelector().text(\"WebView\");"));
    webview.click();
    System.out.println(driver.getContextHandles());

    for(AndroidElement e : driver.findElementsByXPath( using: "//*[@*]")){
        System.out.println(e.getTagName());
        System.out.println(e.getText());
    }

    System.out.println(driver.getContextHandles());
}
```