

APP自动化测试

深圳市门道信息咨询有限公司
Shenzhen MT Information Consulting Co., LTD
版权所有. 侵权必究

Chapter 1 APP自动化测试

Chapter 2 Appium测试环境搭建

Chapter 3 Appium关键技术

Chapter 4 RF与Appium

Chapter 5 Appium测试农场

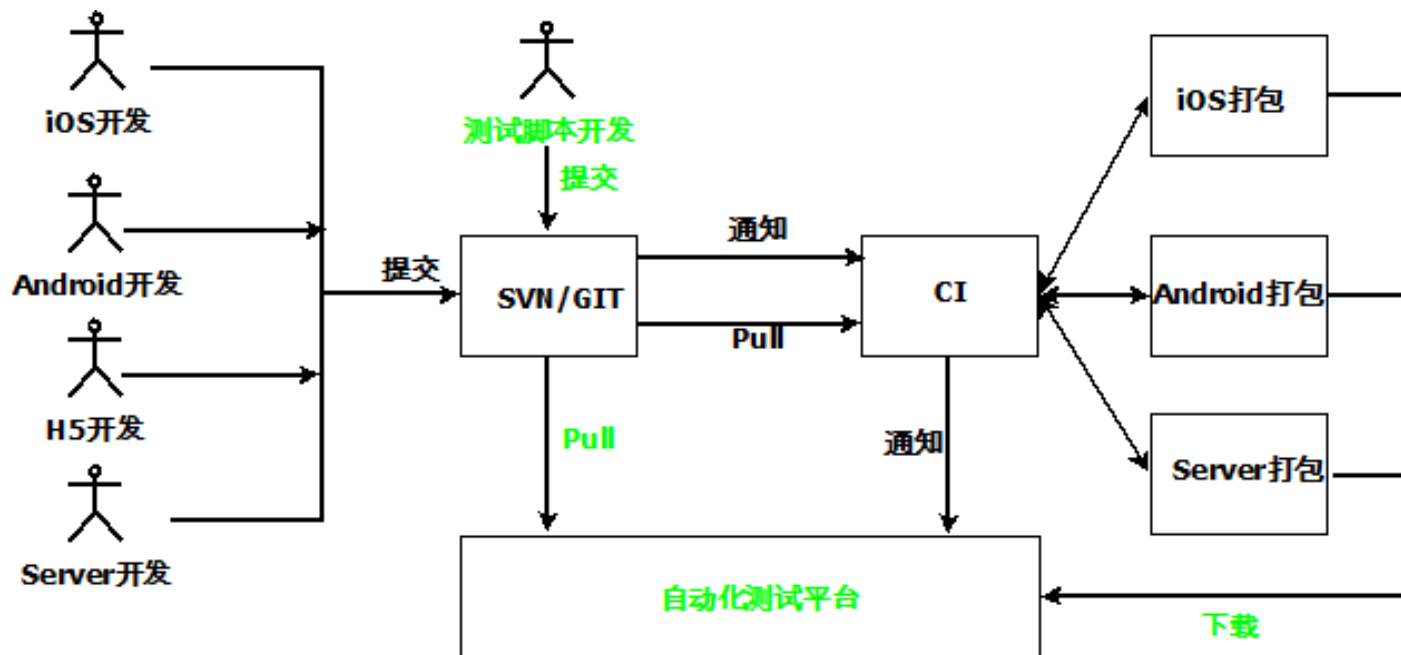
1.1 APP自动化测试优点

优点:

- 1, 脚本可重复执行, 测试效率高
- 2, 执行速度快
- 3, 脚本可以跨平台
- 4, 长时间运行时比人工可靠
- 5, 解放人力

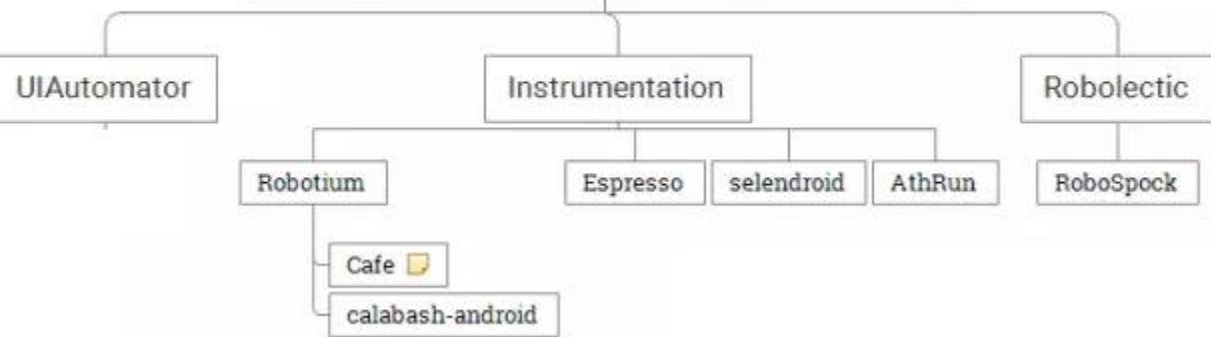
缺点:

- 1, 开发工作量大
- 2, 维护工作量大
- 3, 技能要求高
- 4, 失败用例仍然需要人工分析
- 5, 无效脚本很多
- 6, 不容易做稳定

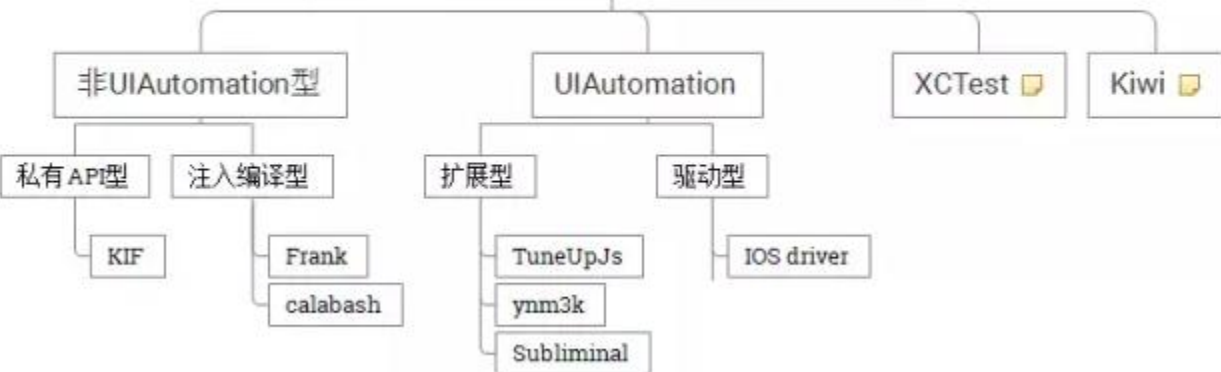


1.2 APP自动化测试

Android自动化功能测试框架



IOS自动化功能测试框架



通用APP自动化测试框架



2.1.1 Appium介绍

Appium是一个开源的、跨平台的APP自动化测试工具，支持原生、移动Web、Hybrid APPs测试。

Supported Platforms

- iOS
- Android
- FirefoxOS



<http://appium.io>

2.1.2 Appium社区

- Appium由SauceLabs支持
- Saucelabs是一家业界领先的云测试产品公司，提供Web/APP云测试服务
- 在Google举办的GATC2013大会上有过详细介绍
- 2013年12月Selenium官方组织宣布放弃自己的mobile项目, 改而支持Appium Selendroid iosdriver框架
- 2016年，SauceLabs将Appium捐赠给了Javascript基金会



Bossie Awards: Appium Named Best
Open Source Mobile Testing
Posted September 29, 2014 by Sauce Labs



Deloitte's 2016 Technology Fast 500 List
Posted November 17, 2016 by Sauce Labs



Sauce Labs Named Gartner 2014 Cool
Vendor
Posted October 15, 2014 by Sauce Labs



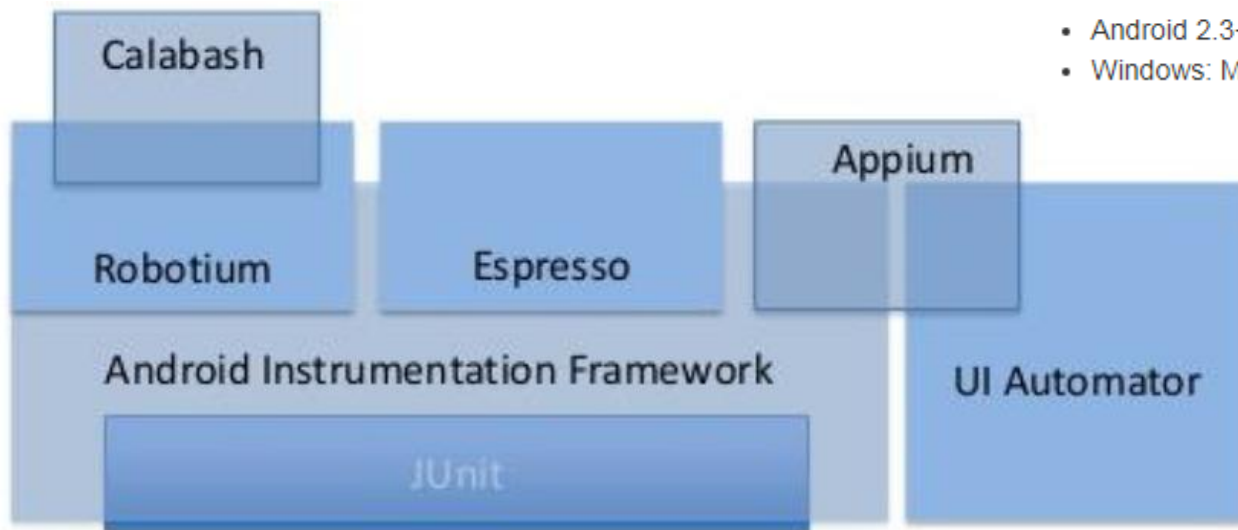
<https://saucelabs.com/>

<http://appium.io/docs/cn/about-appium/intro/>

2.1.3 Appium优点

- 跨架构, native hybrid Webview
- 跨设备, android, ios, firefoxs
- 跨语言, java, python, ruby, nodejs, php
- 跨进程, 可以在多个app之间交互
- 不依赖源代码
- 不限制测试框架和平台

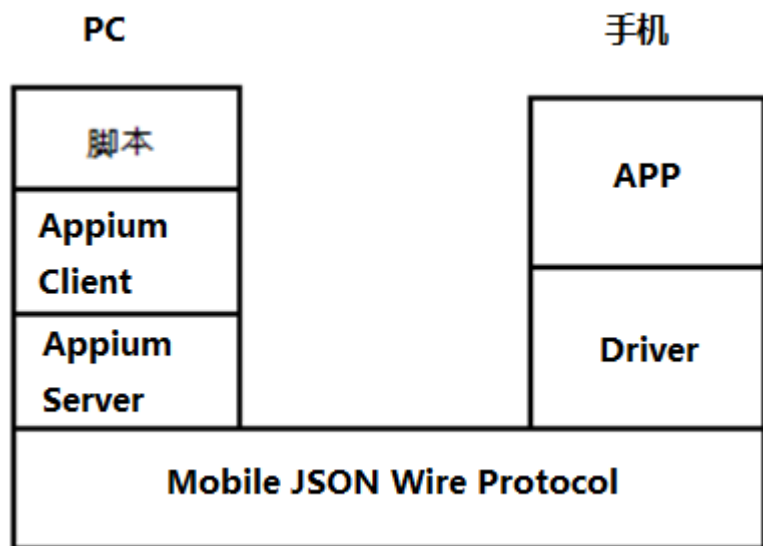
Android测试框架



- iOS 9.3+: Apple's [XCTest](#)
- Android 4.2+: Google's [UiAutomator](#)
- Android 2.3+: Google's [Instrumentation](#). [Selendroid](#)
- Windows: Microsoft's [WinAppDriver](#)

2.1.4 Appium原理

■Appium是C/S架构的测试工具，用户可以使用各种客户端链接server，server和位于被测设备上的driver之间通过JSON Wire协议传递控制指令，完成对被测设备上APP的自动化测试



Appium Client

Language/Framework	Github Repo and Installation Instructions
Ruby	https://github.com/appium/ruby_lib
Python	https://github.com/appium/python-client
Java	https://github.com/appium/java-client
JavaScript (Node.js)	https://github.com/admc/wd
Objective C	https://github.com/appium/selenium-objective-c
PHP	https://github.com/appium/php-client
C# (.NET)	https://github.com/appium/appium-dotnet-driver
RobotFramework	https://github.com/jollychang/robotframework-appiumlibrary

根据被测平台的不同，Appium适配各种不同的测试框架，采用统一的API提供测试能力

- iOS 9.3 and above: Apple's [XCUITest](#)
- iOS 9.3 and lower: Apple's [UIAutomation](#)
- Android 4.2+: Google's [UiAutomator](#)/[UiAutomator2](#)
- Android 2.3+: Google's [Instrumentation](#). ([Selendroid](#))
- Windows: Microsoft's [WinAppDriver](#)

2.2.1 Appium测试环境(Server)

纯Server版本:



- 安装nodejs

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。
Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型，使其轻量又高效。
Node.js 的包管理器 npm，是全球最大的开源库生态系统。

- 使用npm安装appium

```
> npm install -g cnpm --registry=https://registry.npm.taobao.org  
> cnpm install -g appium
```

- 启动appium server


```
D:\node_modules\.bin>appium.cmd  
[Appium] Welcome to Appium v1.7.2  
[Appium] Appium REST http interface listener started on 0.0.0.0:4723  
[Appium] Received SIGINT - shutting down
```

2.2.2 Appium测试环境(Server)

Desktop版本:


安全 | [https://github.com/appium...](https://github.com/appium/appium-desktop/releases) ☆

1.2.7

 triager released this on 21 Nov 2017 · 32 co

Assets

 [appium-desktop-1.2.7-mac.zip](#)

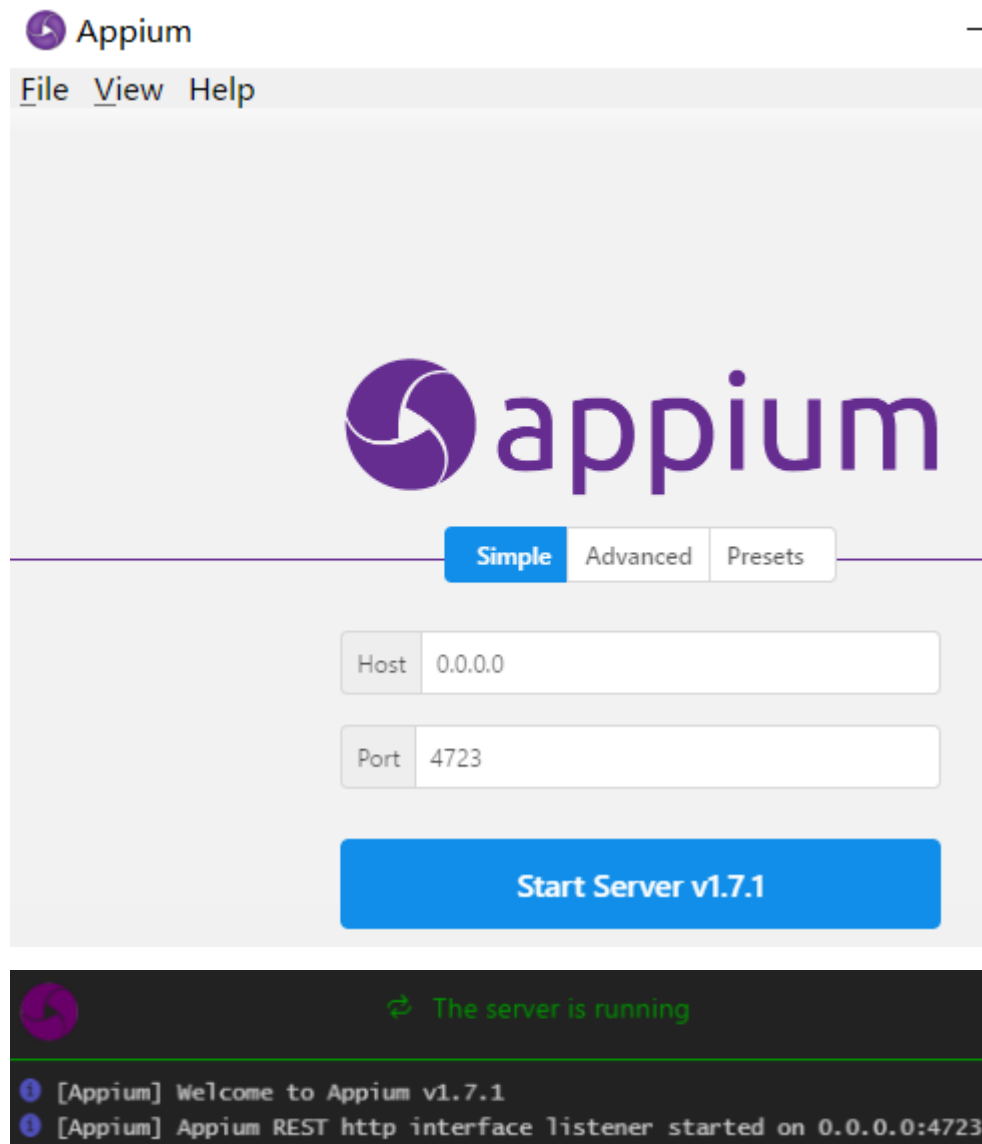
 [appium-desktop-1.2.7-x86_64.AppImage](#)

 [appium-desktop-1.2.7.dmg](#)

 [appium-desktop-Setup-1.2.7.exe](#)

 [latest-mac.json](#)

 [latest.yml](#)



<https://github.com/appium/appium-desktop/releases>

2.3 Appium被测环境(SUT)

模拟器:

■显示系统中全部Android平台:

```
\> android list targets
```

■显示系统中全部AVD（模拟器）:

```
\> android list avd
```

■启动模拟器:

```
\> emulator -avd 模拟器名称
```

■启动Activity :

```
\> adb shell am start -n 包名/包名+类名
```

■指定虚拟机安装app

```
\> adb -s emulator-554 install ..apk
```

3.1 Desired Capabilities

Desired Capabilities: 客户端以key:value字典的方式通知Appium Server，本次测试需要的测试设备的能力及APP的信息，Server根据这些能力选择系统接入的测试设备，启动相关的APP

可以使用JSON传参：

```
# 未处理的 JSON 作为一个参数
--default-capabilities '{"app":
"myapp.app", "deviceName": "iPhone
Simulator"}'
```

```
# 或一个 JSON 文件的名字
--default-capabilities
/path/to/file.json
```

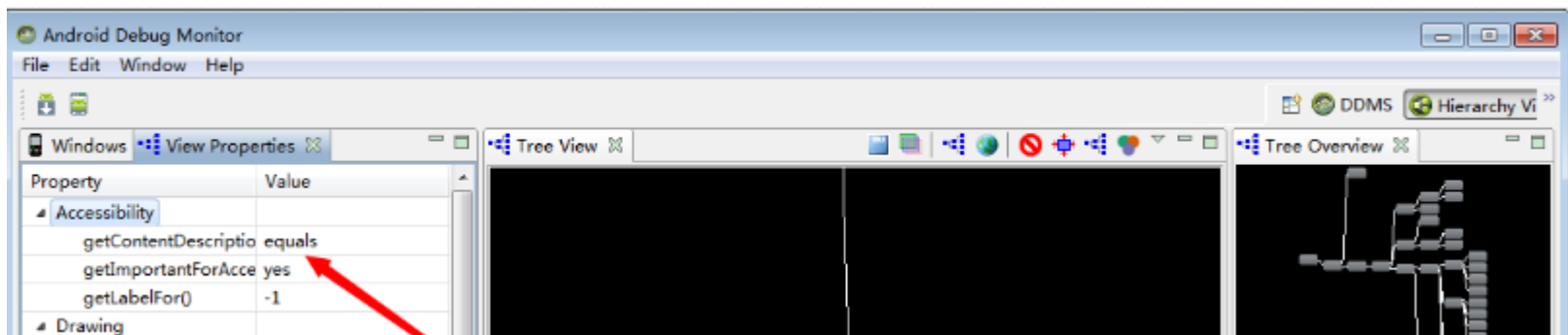
关键字	描述
platformName	手机操作系统
platformVersion	手机操作系统版本
deviceName	设备/模拟器名
app	被测APP绝对路径
browserName	web 浏览器名称，如果是测APP则为空
automationName	自动化测试引擎
appActivity	Android Activity 名称。
appPackage	Android应用的包名
appWaitActivity	等待启动的 Android Activity 名称
unicodeKeyboard	测试中是否切换到能支持多国语言输入的输入法
resetKeyboard	测试完成后是否自动切换回原有输入法

```
{
  "platformName": "Android",
  "deviceName": "emulator-5554",
  "platformVersion": "7.1.1",
  "app": "c:\\app-release.apk",
  "appPackage":
  "com.amazonaws.devicefarm.android.referenceapp"
}
```

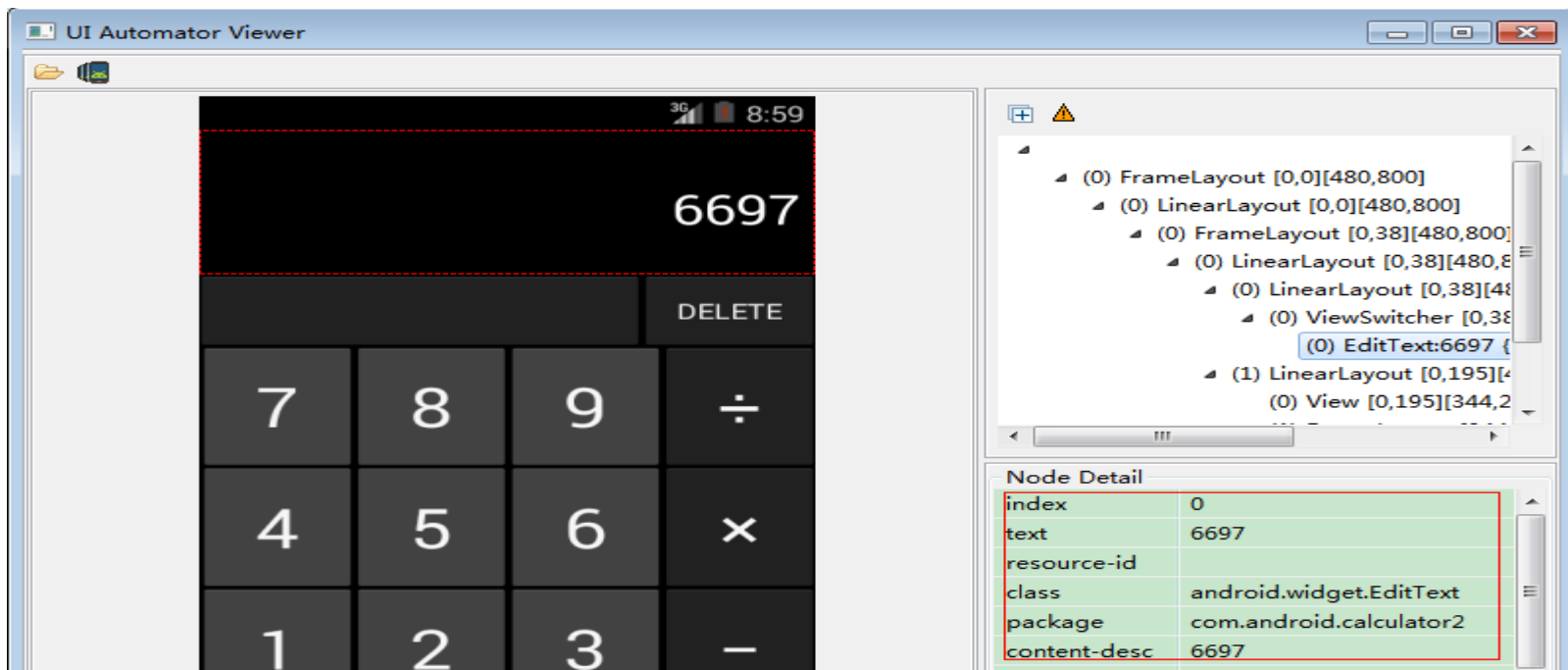
<http://appium.io/docs/en/writing-running-appium/default-capabilities-arg/index.html>

3.2 工具

Tools\monior.bat: 在**hierarchyviewer**中查询accessibility_id 定位符

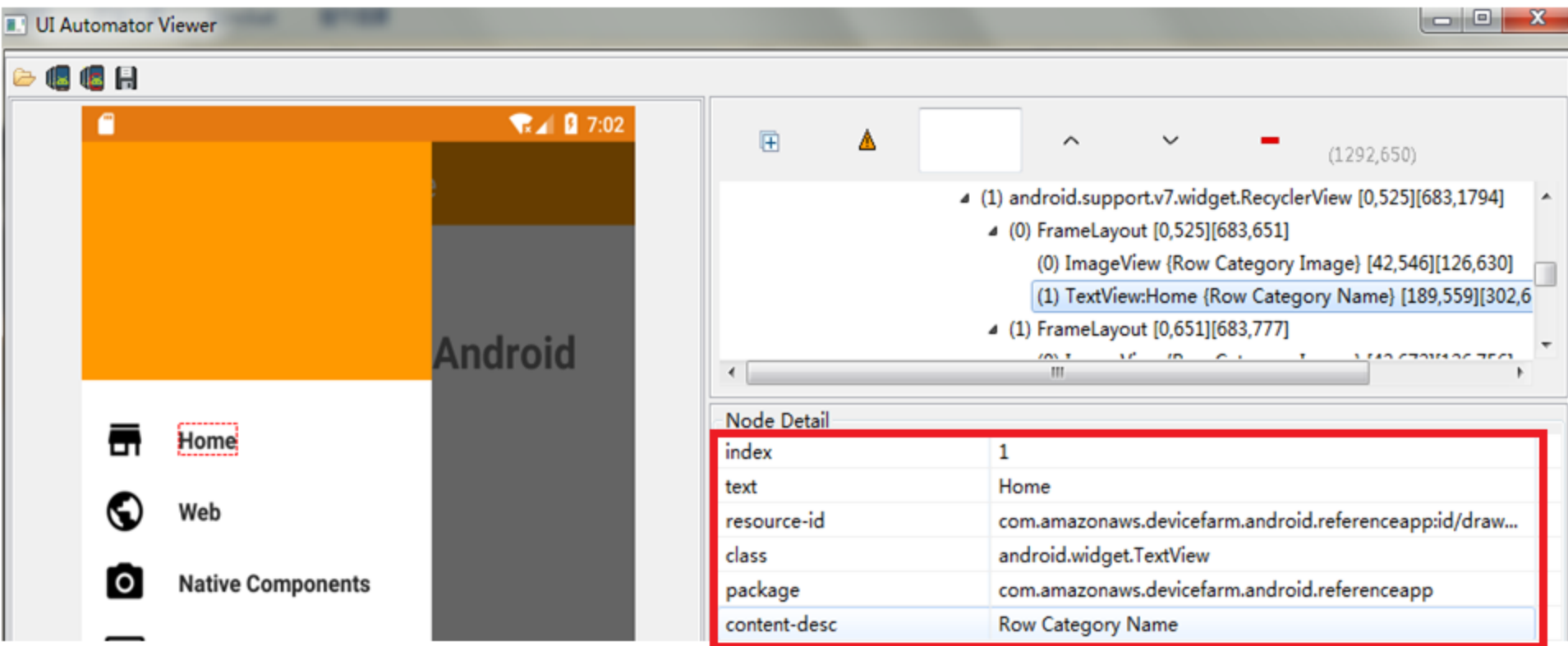


Tools\bin\uiautomatorviewer.bat:查看控件的属性，比如resource id，class name等，以及被测app的appPackage（caps中用）



3.3 定位技术

- 通过资源id查找
- 通过 "class" 查找 (例如: UI 组件的类型)
- 通过 "xpath" 查找 (例如: 一个元素的路径以抽象的方式去表达, 具有一定的约束)
- Accessibility id: 该字符串相当于利用原生的可访问性 (Accessibility) 选项, 使用 Id/Name 去递归地搜索元素。
- 通过UI Automator定位
- 通过css定位

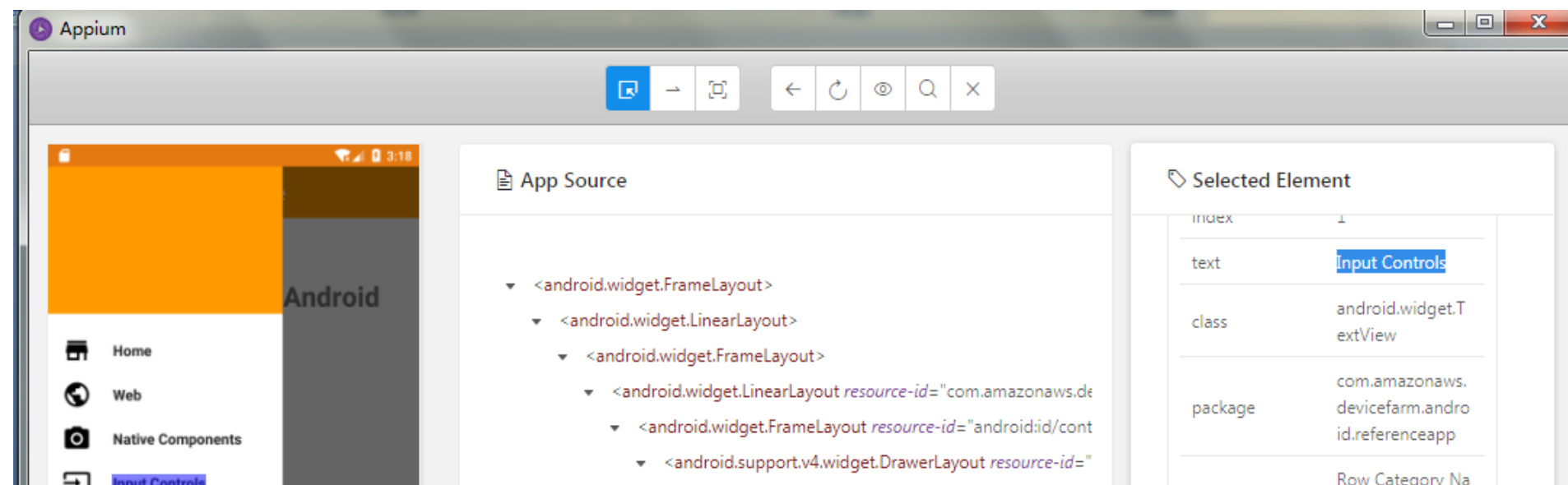


3.4 动作与手势

■ 键盘输入	Input xxx Press Keycode
■ 收起键盘	Hide Keyboard
■ 按压	
■ 长按	Long Press Long Press Keycode
■ 点击	Click xxx
■ 暂停	
■ 息屏	Lock
■ 滑动	Swipe
■ 摇一摇	Shake
■ 上传下载文件	Push Pull

4.1 定位符

Strategy	Example	Description	Note
identifier	Click Element /identifier=my_element	Matches by @id attribute	
id	Click Element /id=my_element	Matches by @resource-id attribute	
accessibility_id	Click Element /accessibility_id=button3	Accessibility options utilize.	
xpath	Click Element /xpath=//UIATableView/UIATableCell/UIAButton	Matches with arbitrary XPath	
class	Click Element /class=UIAPickerWheel	Matches by class	
android	Click Element /android=UiSelector().description('Apps')	Matches by Android UI Automator	
ios	Click Element /ios=.buttons().withName('Apps')	Matches by iOS UI Automation	
css	Click Element /css=.green_button	Matches by css in webview	
name	Click Element /name=my_element	Matches by @name attribute	Only valid for Selendroid



The screenshot displays the Appium desktop application interface. On the left, a mobile app mockup is shown with a navigation drawer containing 'Home', 'Web', 'Native Components', and 'Input Controls'. The 'Input Controls' item is selected. The central pane, titled 'App Source', shows a hierarchical tree of the app's XML structure, with the following path expanded: `<android.widget.FrameLayout>` `<android.widget.LinearLayout>` `<android.widget.FrameLayout>` `<android.widget.LinearLayout resource-id="com.amazonaws.de">` `<android.widget.FrameLayout resource-id="android:id/cont">` `<android.support.v4.widget.DrawerLayout resource-id="`. The right pane, titled 'Selected Element', displays the attributes of the selected element: `index` (1), `text` (Input Controls), `class` (android.widget.TextView), and `package` (com.amazonaws.devicefarm.android.referenceapp). The bottom of this pane shows 'Row Category Na'.

4.1 定位符

Android定位符:

API	说明
text(String text)	文本
textContains(String text)	文本包含
textMatches(String regex)	文本正则
textStartsWith(String text)	文本起始匹配
description(String desc)	描述
descriptionContains(String desc)	描述包含
descriptionMatches(String regex)	描述正则
descriptionStartsWith(String desc)	描述开始字符匹配
className(String className)	类名
classNameMatches(String regex)	正则类名
packageName(String name)	包名
packageNameMatches(String regex)	包名正则
index(int index)	索引
instance(int instance)	实例
checked(boolean val)	选择属性
clickable(boolean val)	可点击属性
enabled(boolean val)	enabled属性
focusable(boolean val)	焦点属性
focused(boolean val)	当前焦点属性
longClickable(boolean val)	长按属性
scrollable(boolean val)	滚动属性
selected(boolean val)	背景选择属性
childSelector(UiSelector selector)	从当前类中往下递归找符合条件的子类组件
formPrent(UiSelector selector)	从父类往下递归找符合条件的组件
resourceId(String id)	资源ID
resourceIdMatches(String regex)	资源ID正则

Click Element

`android=UiSelector().textContains('Input Controls')`

log

.复选框

Click Element

`android=UiSelector().text('Checkbox')`

4.2 WebView

■ WebView是基于Chrome浏览器内核的界面：

1	Click Element	class=android.widget.ImageButton	
2	\${x}	get_element_by_text	id=com.amazonaws.devicefarm.ar Web
3	click element	\${x}	
4	\${x}	Get Contexts	
5	: FOR	\${i}	IN @ {x}
6		log	\${i}
7	Input Text	class=android.widget.EditText	http://www.baidu.com
8	Press Keycode	\${66}	
9	Switch To Context	\${x[-1]}	
10	Wait Until Element Is Visible	id=word	
11	Input Text	id=word	门道
12	Switch To Context	\${x[0]}	
13	click element	xpath=//android.widget.Button	
14	sleep	5	
15	Comment	Click Element	xpath=//*[@class = 'android.view.View' and contains(@content-desc,'百度百科
16	Click Element	xpath=//android.view.View[contai	

4.3 Alerts

■ Alert和Toast是两种展示提示信息的技术：

1	Click Element	class=android.widget.ImageButton	
2	<code>\${x}</code>	get_element_by_text	id=com.amazonaws.devicefarm.ar Alerts
3	click element	<code>\${x}</code>	
4	Click Button	TOAST	
5	log	=====	
6	Comment	Wait Until Page Contains Element	xpath=//*[@text="1657231876"]
7	<code>\${x}</code>	Get Webelement	xpath=//*[@text="1657231876"]
8	Click Button	ALERT	
9	log	点击OK按钮	
10	Click Button	OK	
11	Comment	click element	class=android.widget.Button

4.4 swipe(滑动)

■ 处理滑动操作

1	Click Element	class=android.widget.ImageButton			
2	\${x}	get_element_by_text	id=com.amazonaws.devicefarm.ar	Native	
3	Click Element	\${x}			
4	\${x}	Get Window Width			
5	\${y}	Get Window Height			
6	Swipe	1000	700	100	700
7	sleep	5			
8	Swipe	400	1500	400	700
9	sleep	5			

4.5 Input

1	Click Element	class=android.widget.ImageButton		
2	\${x}	get_element_by_text	id=com.amazonaws.devicefarm.ar	Input
3	Click Element	\${x}		
4	Input Text	accessibility_id=Text Input Control	ravi huang	
5	\${x}	get_element_by_text	class=android.widget.TextView	Checkbox
6	Click Element	\${x}		
7	sleep	3		
8	Click Element	accessibility_id=Checkbox Control		
9	\${x}	Get Text	accessibility_id=Checkbox Display	
10	Should Be Equal	\${x}	Checked	

■ 导航键的处理方法

1	Click Element	class=android.widget.ImageButtor	
2	\${x}	get_element_by_text	id=com.amazonaws.devicefarm.arNested
3	Click Element	\${x}	
4	click button	UP NAVIGATION	
5	Press Keycode	\${4}	
6	click button	BACK NAVIGATION	
7	Press Keycode	\${4}	
8	sleep	10	

4.7 支持中文输入

■ 设置Caps，支持中文输入：

```
app={"platformName":"Android",  
    "deviceName":"emulator-5554",  
    "app": "%s/data/app-release.apk" % os.getcwd(),  
    "appPackage":"com.amazonaws.devicefarm.android.referenceapp",  
    "appActivity": ".Activities.MainActivity",  
    "platformVersion":"7.1.1",  
    "chromedriverExecutable":"c:/Python27/chromedriver.exe",  
    "unicodeKeyboard":"true",  
    "resetKeyboard":"true"}
```

5.1.1 STF

STF(Smartphone Test Farm)是用于Android设备管理的工具。

- 1、在Web上支持管理上百个android设备。
- 2、支持Android多个版本（2.3.3 ~ 5.1），而且不需要root。
- 3、实时屏幕操作和显示。
- 4、支持adb connect远程连接调试。
- 5、可以从PC机键盘输入到远程的android设备中。
- 6、安装卸载APK。
- 7、android设备信息的展示（如：网络状态、MIME、android版本、手机型号等等）。
- 8、远程开关机、远程开关WIFI、截屏、LogCat等等。



5.1.2 STF安装

■使用Docker安装STF

yum install docker

wget https://github.com/docker/compose/releases/download/1.19.0-rc1/docker-compose-Linux-x86_64

```
[root@localhost ~]# more docker-compose.yml
```

```
rethinkdb:
  image: rethinkdb:2.3
  ports:
    - "8080:8080"
    - "28015:28015"
    - "29015:29015"
  restart: always
  volumes:
    - "/srv/rethinkdb:/data"
  command: "rethinkdb --bind all --cache-size 2048"
adbd:
  image: sorccu/adbd
  privileged: true
  ports:
    - "5037:5037"
  restart: always
  volumes:
    - "/dev/bus/usb:/dev/bus/usb"
stf-local:
  image: openstf/stf
  links:
    - rethinkdb
    - adbd
  ports:
    - "7100:7100"
    - "7110:7110"
    - "7120:7120"
    - "7400-7500:7400-7500"
  restart: always
  command: stf local --public-ip 192.168.164.164 --provider-min-port 7400 --provider-max-port 7500 --adb-host adbd
```

5.1.3 STF环境

■STF可以查看手机信息，集中管理被测设备。STF链接手机的方法有两种：
USB方式：

1505-A01

17:27 4G

门道科技

群主 门道科技-张老师
这种效果还是不错的

管理员 周丽
好的

吐槽 郭璇铃
好

14:55

吐槽 郭璇铃

控制面板 更多屏幕截图 自动化 高级 文件管理器 信息

物理设备 查找设备
位置 5ed72b98b1a6

SIM
信号 CHN-UNICOM,CMCC
网络 UNKNOWN
数字
IMEI 863170030892597
IMSI 460015677519119
集成电路卡识别码 89860116851066753524

CPU
名称 -

电池
健康 良好
电源来源 USB
状态 充电中
等级 45%
温度 32 °C
电压 3.863 v

硬件
制造商 360
产品 -
型号 1505-A01
串行 622123171691429664
释放 -

播放
大小 -
密度 XXHDPI
FPS 59.1300048828125
宽度 1080 px
高度 1920 px
ID 0
屏幕方向 0°
加密的 Yes
X DPI 403.4110107421875
Y DPI 403.0409851074219

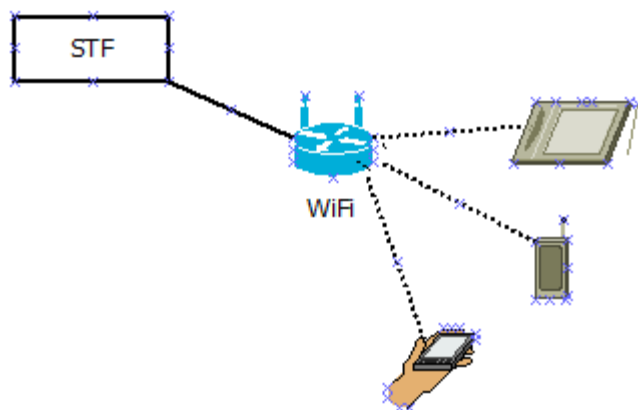
内存
随机存取存储器 -

网络
已连接 Yes
飞行模式 No
使用回退 No
漫游状态 No
类型 MOBILE
子类型 LTE

平台
操作系统 Android
版本 6.0
SDK 23
ABI arm64-v8a

5.1.3 STF环境

■Wifi方式:



手机端配置:

\> adb tcpip 5555

```
C:\>adb shell ip ad ls
25: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 74:ac:5f:af:06:e3 brd ff:ff:ff:ff:ff:ff
    inet 172.16.5.100/24 brd 172.16.5.255 scope global
        valid_lft forever preferred_lft forever
```

STF端配置:

adb connect 172.16.5.100:5555

5.1.4 STF使用

■ STF可以执行shell命令

以及管理APP

Shell脚本

清除

ps|

USER	PID	PPID	VSIZE	RSS	WCHAN	PC	NAME
root	1	0	13108	1848	SyS_epoll_	0000000000	S /init
root	2	0	0	0	kthreadd	0000000000	S kthreadd
root	3	2	0	0	smpboot_th	0000000000	S ksoftirqd/0
root	7	2	0	0	rcu_gp_kth	0000000000	S rcu_preempt
root	8	2	0	0	rcu_gp_kth	0000000000	S rcu_sched
root	9	2	0	0	rcu_gp_kth	0000000000	S rcu_bh
root	10	2	0	0	smpboot_th	0000000000	S migration/0
root	11	2	0	0	smpboot_th	0000000000	S migration/1
root	12	2	0	0	smpboot_th	0000000000	S ksoftirqd/1
root	14	2	0	0	worker_thr	0000000000	S kworker/1:0H
root	15	2	0	0	smpboot_th	0000000000	S migration/2
root	16	2	0	0	smpboot_th	0000000000	S ksoftirqd/2
root	18	2	0	0	worker_thr	0000000000	S kworker/2:0H
root	19	2	0	0	smpboot_th	0000000000	S migration/3
root	20	2	0	0	smpboot_th	0000000000	S ksoftirqd/3
root	21	2	0	0	worker_thr	0000000000	S kworker/3:0

上传APP

清除



拖放文件到这里以上传

应用程序

应用商店

设置

开发者

管理Apps

WiFi

联系我们

电话：0755-83221336/13928429246（微信同步）

邮箱：service@mtesting.net

官网：www.mtesting.net

学习社区：www.mtesting.cn

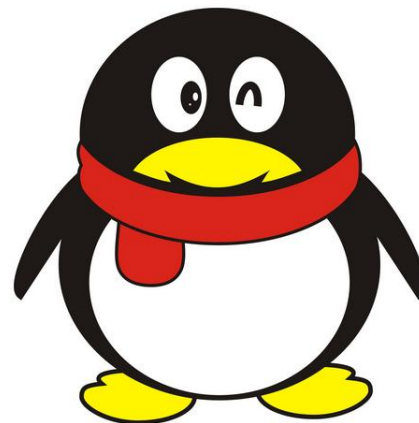
地址：深圳市福田区彩田南路深圳青年大厦五楼



手机扫描访问官网



手机扫描关注公众号



QQ群号:15233368

感谢您对我们的关注



Thanks&Best wishes for you!

多动脑 勤动手 定成功