浅谈如何用APPIUM做移动端自动化测试

Testerhome 广州站第一期

目录

官方自动化测试工具

- Android
  - 单元测试： AndroidJUnitRunner
  - 稳定性测试： Monkey
  - UI 自动化测试：
    - Espresso(base on instrumentation)
    - UIAutomator(cross-app)

官方自动化测试工具

- iOS
- 单元测试：**XCTest**
- **UI 自动化测试：UIAutomation**

APPIUM 的介绍

- 什么是 Appuim
- Appium 的定位
- Appium 的哲学
- Appium 与 Selenium
- Appium的结构

什么是 APPIUM

- Appium is an open source test automation framework for use with native, hybrid and mobile web apps.
- Appium 是一个开源的自动化测试框架，用于对原生应用，混合应用和 web 应用进行测试
- It drives iOS and Android apps using the WebDriver protocol.
- 它通过 WebDriver protocol 驱动 iOS 和 Android 应用

APPIUM 的定位

- UI 自动化测试
- 现有测试工具的再封装
- android 和 iOS 使用统一的 API 编写测试

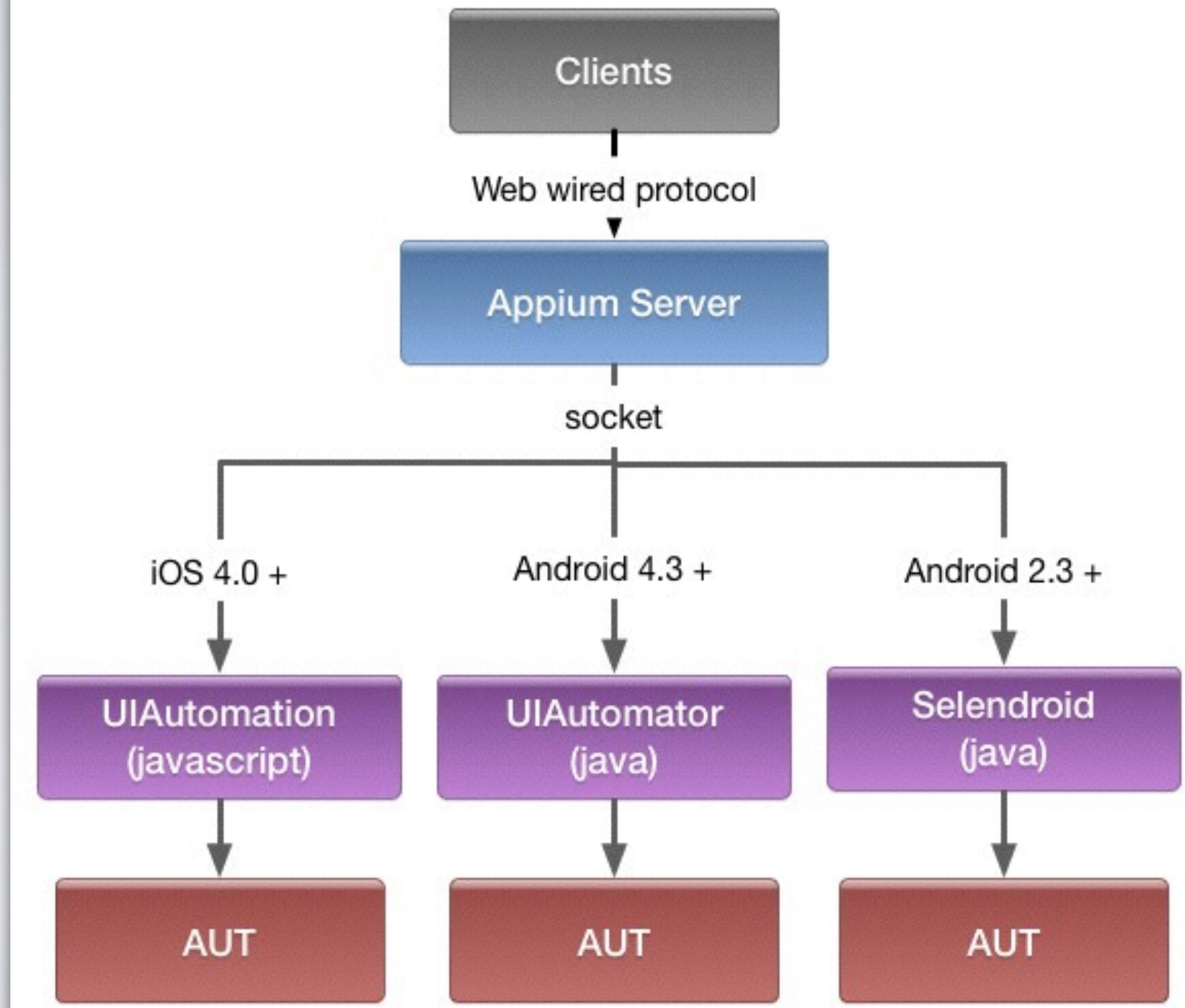
APPIUM 的哲学

- 你不应该为了自动化而重新编译/修改你的应用——使用 UIAutomator，UIAutomation 这种官方提供的测试框架，从而不用在应用中加入额外接口
- 你不应该被限定于使用某种特定语言/框架来执行测试——采用 WebDriver 协议与 client 进行通讯，client 可以使用多种语言
- 一个移动自动化测试框架不应该重复造轮子——执行方面采用官方的测试工具，API 方面使用 Webdriver API
- 一个移动自动化框架应该在精神上和实际上开源——Appium 目前开源

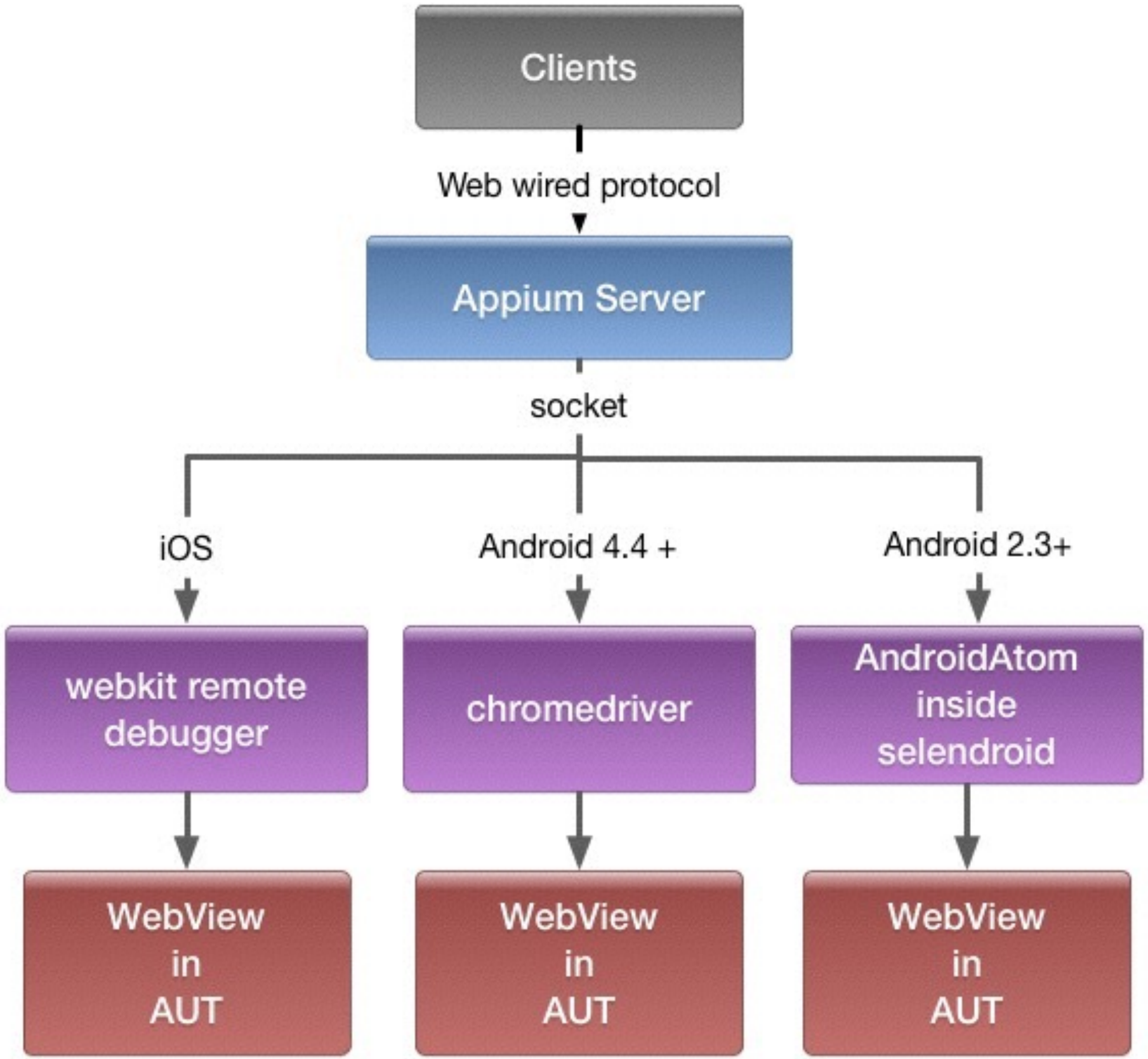
APPIUM 与 SELENIUM

- Selenium2 又叫 Selenium Webdriver
- Appium Clients 扩展了 Selenium 的 WebDriver API
- Appium Server 实现了 Selenium 中存在的大部分方法，它属于 Selenium 的第三方 webdriver

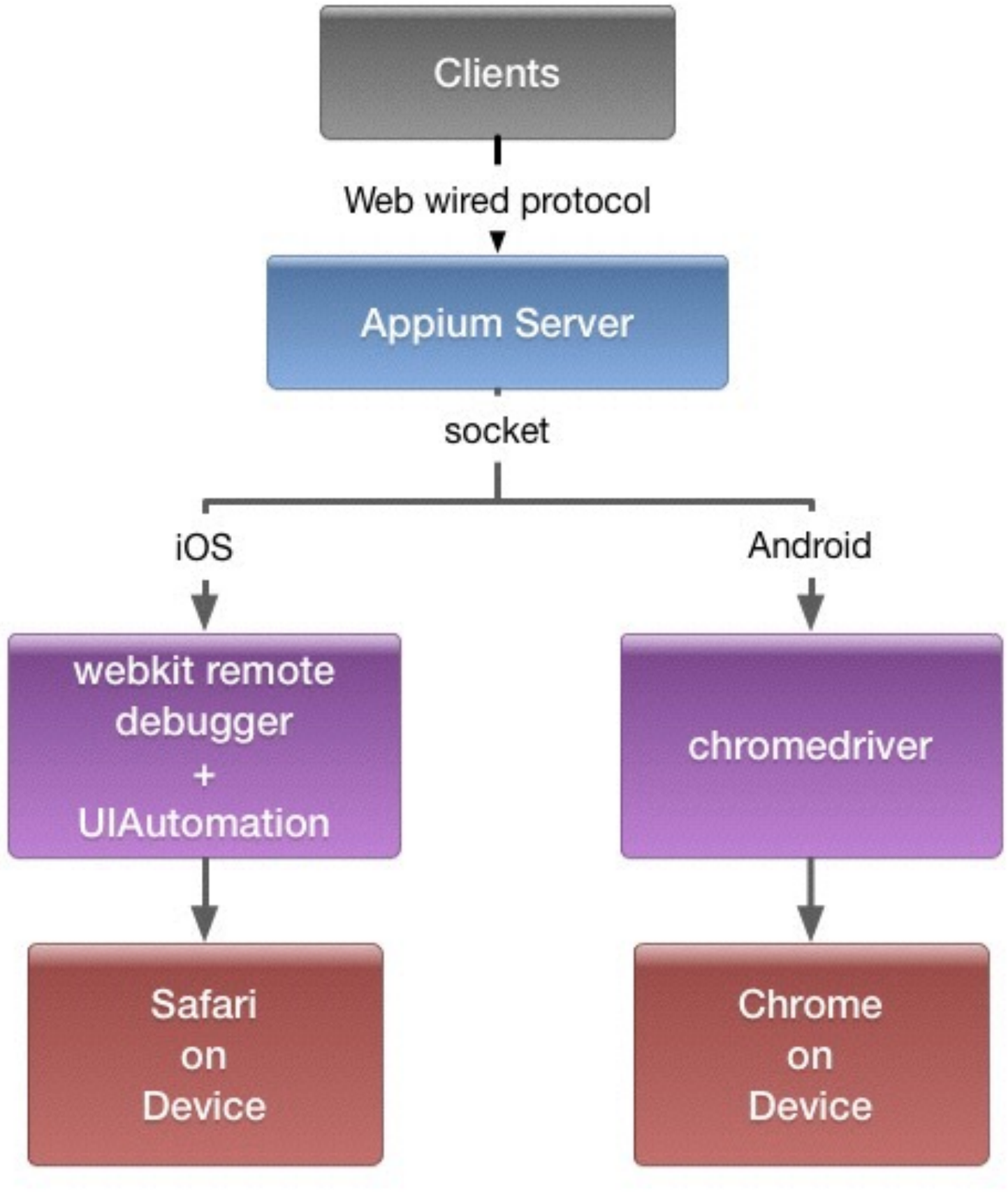
Appium Topology Native



Appium Topology : Hybrid Webview



Appium Topology: Web



APPIUM 的使用

- 官方文档及示例代码
- 编写/执行脚本
- 实际使用中的坑
- 查错方法

官方文档及示例代码

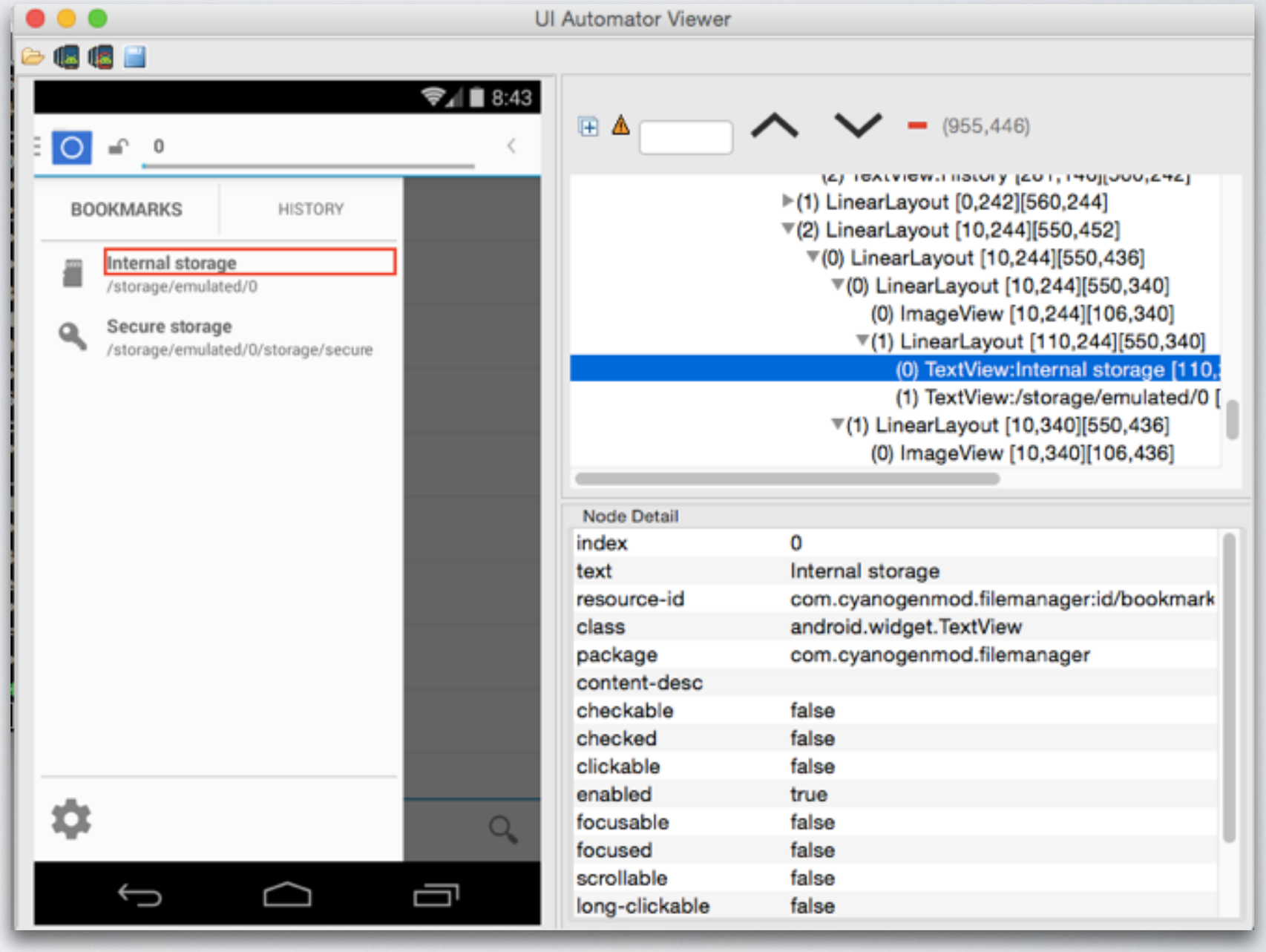
- 英文版： http://appium.io/slate/en/master/
- 中文版： http://appium.io/slate/cn/master/
- 示例代码： https://github.com/appium/sample-code/

编写脚本

- 元素查找
- native/hybrid/web app 的设置
- WebDriver API 的使用

元素查找

Android native app (4.3+)



元素查找

Android

hybrid app

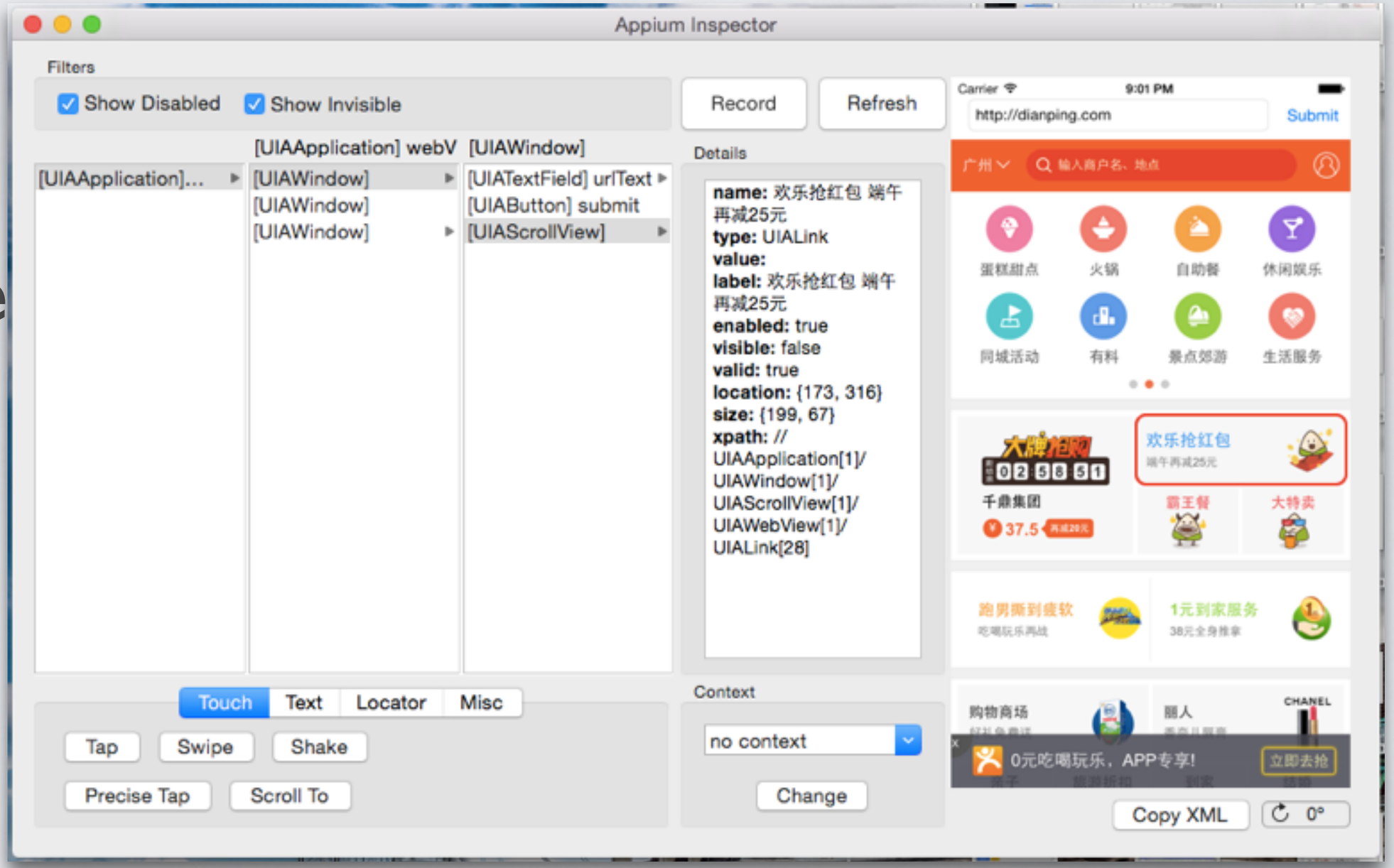
/web app(4.4+)

元素查找

iOS native

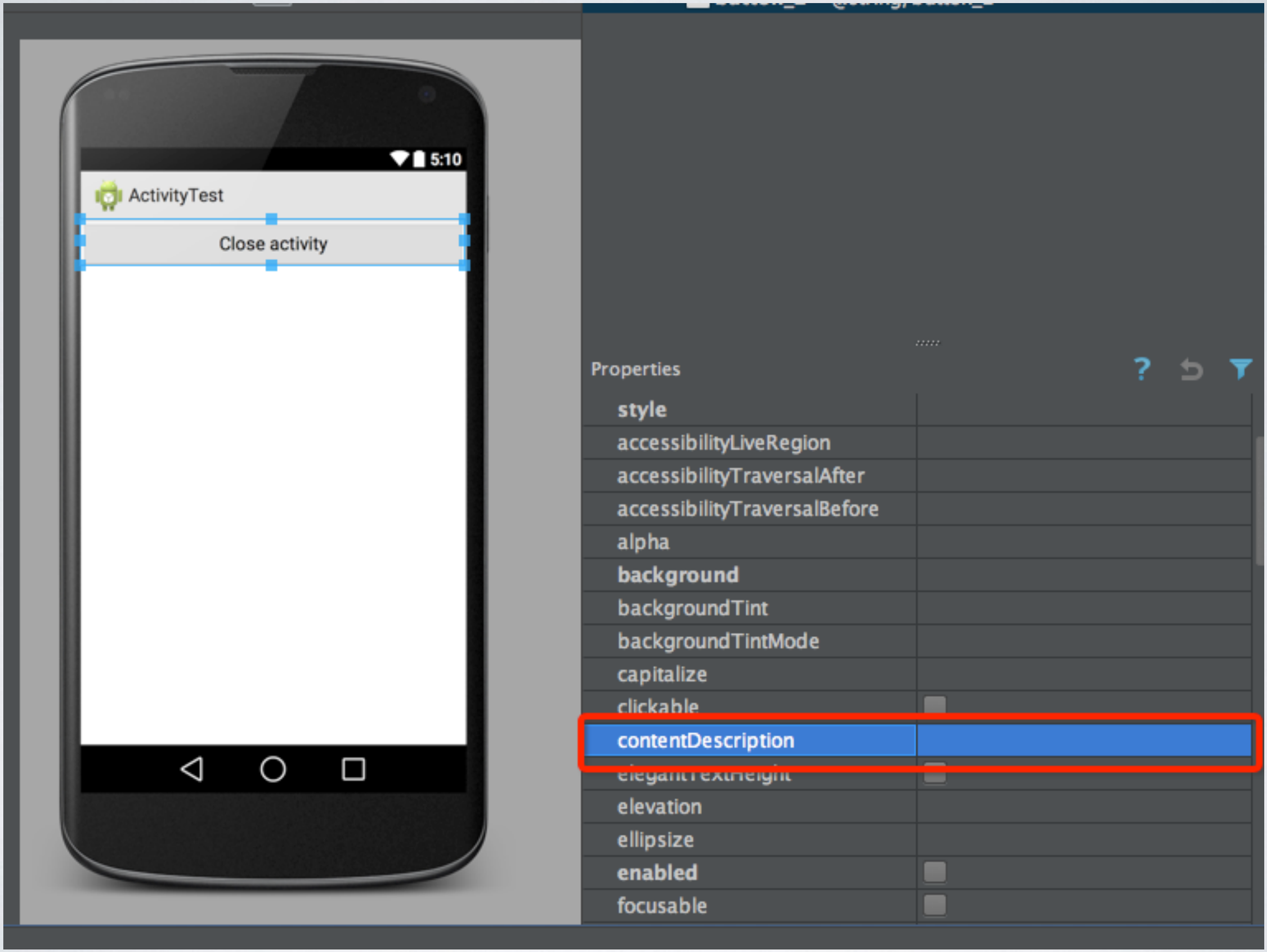
/hybrid

/web app



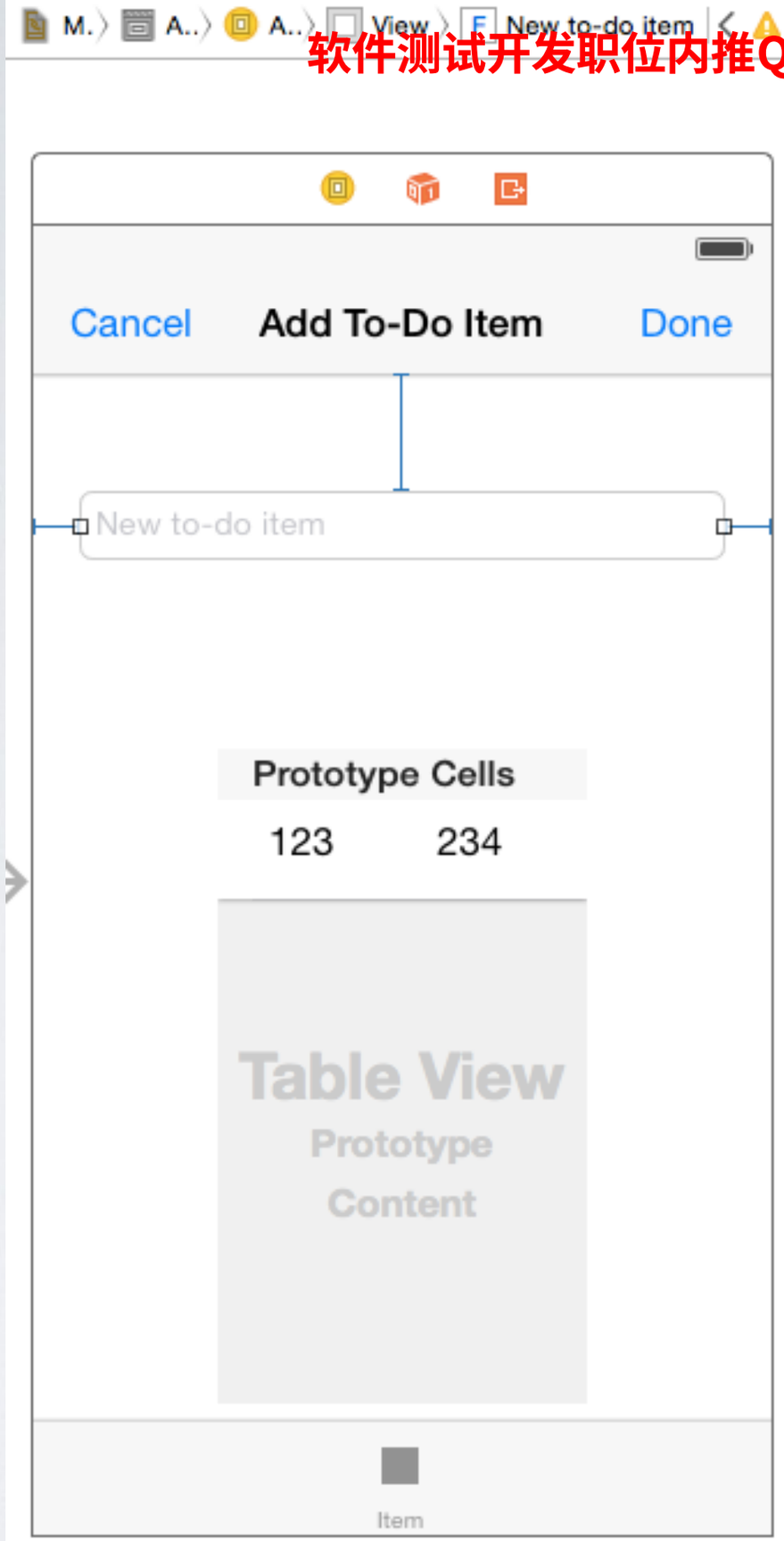
元素查找

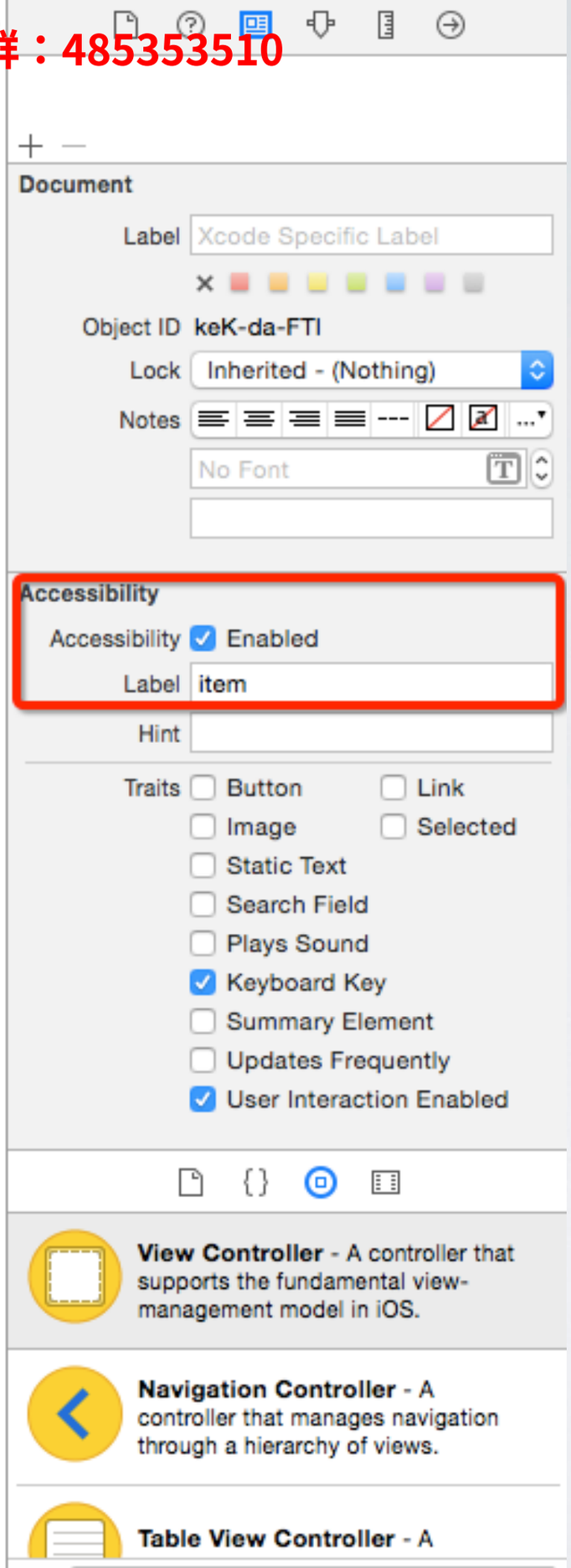
- Android 上建议使用 content-description，这个属性大部分情况下与功能无关，添加后不会影响功能
- 查找时使用 find_element_by_accessibility_id (python)



元素查找

- iOS 上的元素若没有 Enable Accessibilty 将无法在 UIAutomation 中获取
- 建议使用 Accessibility 中的 Label 进行定位，可通过 find_element_by_accessibility_id 查找（python）





不同类型 APP 的设置

- Native

```
desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '4.4'
desired_caps['deviceName'] = 'e4d42545'
desired_caps['unicodeKeyboard'] = 'true'
desired_caps['resetKeyboard'] = 'true'
desired_caps['app'] = '/Users/hengjiechen/Develop/app/Dianping_dianping-web_7.1.1.apk'
self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

不同类型 APP 的设置

- Hybrid

```
desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '4.4'
desired_caps['deviceName'] = 'e4d42545'
desired_caps['unicodeKeyboard'] = 'true'
desired_caps['resetKeyboard'] = 'true'
desired_caps['app'] = '/Users/hengjiechen/Develop/app/webview-debug-unaligned.apk'
self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

```
for context in self.driver.contexts:
    if "WEBVIEW" in context:
        webview_context = context

self.driver.switch_to.context(webview_context)
```

不同类型 APP 的设置

- Web app

```
desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '4.4'
desired_caps['deviceName'] = '750ACKS3LSE4'
desired_caps['browserName'] = "Chrome"
self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

不同类型 APP 的设置

- 在 webView/browser 中大部分 Appium 新增的方法无效

WEBDRIVER API 的使用

- 基础： https://github.com/easonhan007/webdriver_guide

- Appium 新增：

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

WEBDRIVER API 的使用

- 主要新增部分：
  - context 切换（针对 Hybrid app）
  - 使用 UIAutomation 或 UIAutomator API 查找元素
  - 手势及多点触控
  - 应用管理相关（置于后台、安装、卸载等）
  - 设备相关（锁屏、摇晃、获取文件）

执行脚本

- 打开 Appium server
- 保证设备已连接
  - adb devices
  - instrument -s devices
- 运行脚本

实际使用的坑

- 找不到元素，但在查看元素时找得到
- 主要出现在使用 xpath 的时候，通过 driver.page_source 获取当前界面 xml 文件（个别 app 的界面树连元素顺序都会更新），并对比看 xpath 是否能真的定位到元素节点，并考虑通过非 xpath 方式定位
- 在 iOS 上找到元素了，但对其进行 click 操作会出现 cannot be tap 的问题
- 目前没找到确切原因，推测是该元素对象还没完全初始化完成就被获取了。解决办法是获取元素的 location 然后点坐标（有时候坐标是一个类似 5666345e-32 这样的数，必须多次获取）

实际使用的坑

- 对于左右滑动的 scrollView 没有 API 直接控制
- 如果是 android 且界面内容可以通过 content-desc 或 text 定位，可用 UIAutomator 的 UiScrollable 进行元素定位，它会自动滑动。若需要通用，只能通过自己再次封装进行。基本思路：

1. 查看当前界面是否存在要找的元素
2. 若当前界面没有，滑动到最左面，可以通过对比一个不同 view 值会不一样的元素在滑动前后的值以确定是否到最左面
3. 逐个滑动并查看是否有要找的元素，直至滑动到最右面

实际使用的坑

- iOS 上无论元素是否显示都会出现在控件树，导致 xpath 定位困难

- 这个是从 iOS 7 开始出现的。可通过查看元素的 isDisplay() 方法确认元素是否显示在屏幕中。若要筛选则可通过在 xpath 中加上 visible="true" 进行筛选。

查错方法

- 查官方文档
- 根据 appium log 关键内容搜索

APPIUM LOG

- Appium communication log
  - Appium Client 与 Appium Server 通讯的具体内容

```
2015-06-20 03:00:51:228 - info: <-- GET /wd/hub/status 200 9.710 ms - 104 {"status":
0,"value":{"build":
{"version":"1.3.4","revision":"c8c79a85fbd6870cd6fc3d66d038a115ebe22efe"}}}

2015-06-20 03:00:55:902 - info: --> POST /wd/hub/session {"desiredCapabilities":
{"deviceName":"750ACKS3LSE4","unicodeKeyboard":"true","browserName":"Chrome","resetK
eyboard":"true","platformVersion":"4.4","platformName":"Android"}}
```

APPIUM LOG

- Appium server log
  - 指示当前 appium 内部的工作状态
  - 说明 appium 正在调用的系统命令

```
2015-06-20 03:00:55:909 - info: [debug] Creating new appium session
bd75db3f-7a75-43f6-938d-e5a12f0fef4f
2015-06-20 03:00:55:910 - info: [debug] Using fast reset? true
2015-06-20 03:00:55:910 - info: [debug] Preparing device for session
2015-06-20 03:00:55:911 - info: [debug] Not checking whether app is present since we
are assuming it's already on the device
2015-06-20 03:00:55:911 - info: [debug] Checking whether adb is present
2015-06-20 03:00:55:912 - info: [debug] Using adb from /Applications/adt-bundle-mac-
x86_64-20140702/sdk/platform-tools/adb
2015-06-20 03:00:55:913 - info: Retrieving device
2015-06-20 03:00:55:913 - info: [debug] Trying to find a connected android device
2015-06-20 03:00:55:913 - info: [debug] Getting connected devices...
2015-06-20 03:00:55:914 - info: [debug] executing cmd: /Applications/adt-bundle-mac-
x86_64-20140702/sdk/platform-tools/adb devices
```

APPIUM LOG

- UIAutomator log（Android）
  - bootstrap server log
  - adb shell standard output

```
2015-06-20 03:01:29:692 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
numtests=1

2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
stream=
2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT]
io.appium.android.bootstrap.Bootstrap:
2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
id=UiAutomatorTestRunner
2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
test=testRunServer
2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
class=io.appium.android.bootstrap.Bootstrap
2015-06-20 03:01:29:693 - info: [debug] [UIAUTOMATOR STDOUT] INSTRUMENTATION_STATUS:
current=1

2015-06-20 03:01:29:695 - info: [debug] [UIAUTOMATOR STDOUT]
INSTRUMENTATION_STATUS_CODE: 1

2015-06-20 03:01:29:728 - info: [debug] [UIAUTOMATOR STDOUT] tcp port:4724
2015-06-20 03:01:29:731 - info: [debug] [BOOTSTRAP] [debug] Socket opened on port
4724
2015-06-20 03:01:29:733 - info: [debug] [BOOTSTRAP] [debug] Appium Socket Server
Ready
2015-06-20 03:01:29:733 - info: [debug] [BOOTSTRAP] [debug] Loading json...
2015-06-20 03:01:29:733 - info: [debug] Pushing command to appium work queue:
["getDataDir",{}]
```

APPIUM LOG

- Chromedriver log（Android）
  - 指示 chromedriver 启动信息

```
2015-06-20 03:01:29:876 - info: [debug] [CHROMEDRIVER] Starting ChromeDriver
(v2.10.267517) on port 9515
Only local connections are allowed.
```

APPIUM LOG

- UIAutomation log（iOS）
  - 指示 UIAutomation 具体执行状态

```
2015-06-20 03:48:36:526 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug: Got
new command 10 from instruments: au.tapById('2')
2015-06-20 03:48:36:533 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug:
evaluating au.tapById('2')
2015-06-20 03:48:36:534 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug:
UIAButton.tap()

2015-06-20 03:48:37:079 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug:
evaluation finished

2015-06-20 03:48:37:083 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug:
responding with:
2015-06-20 03:48:37:087 - info: [debug] [INST] 2015-06-20 03:48:36 +0000 Debug:
Running system command #11: /Applications/Appium.app/Contents/Resources/node/bin/
node /Applications/Appium.app/Contents/Resources/node_modules/appium/node_modules/
appium-uiauto/bin/command-proxy-client.js /tmp/instruments_sock 2,{"status":
0,"value":""}...
```

APPIUM LOG

服务器参数	默认值	描述	举例
-g, --log	null	同时把 log 保存到指定文件	--log /path/to/appium.log
--log-level	debug	log 级别。默认使用 debug	--log-level debug
--log-timestamp	FALSE	在 log 中加入时间戳信息	

总结

- 官方测试工具数量有限，且不同工具使用方法差异大
- Appium 将其封装起来，使用统一的 API
- Appium 仍在发展中