

# Appium手机自动化测试

TesterHome出品(公开版)



# Automation for Apps

Appium is an open source test automation framework for use with native and hybrid mobile apps. It drives iOS and Android apps using the WebDriver JSON wire protocol.

[Download Appium](#)

[View on GitHub](#)

[Downloads](#)

[Documentation](#)

[Examples](#)

[Contributions](#)

build passing

# Appium维护情况

## Appium

Appium is an open source, cross-platform test automation tool for native, hybrid and mobile web apps.

npm module 0.11.4 build passing

### Supported Platforms

- iOS
- Android
- FirefoxOS

- 由SauceLabs支持，一家业界领先的测试产品公司，有成熟的企业测试解决方案
- Saucelabs有基于appium的云产品
- 在Google举办的GATC2013大会上有过详细介绍

# 目前mobile自动化的方案

| iOS              | Android          |
|------------------|------------------|
| calabash-ios     | calabash-android |
| Frank            | MonkeyTalk       |
| UIAutomation     | Robotium         |
| ios-driver       | UiAutomator      |
| KeepItFunctional | selendroid       |



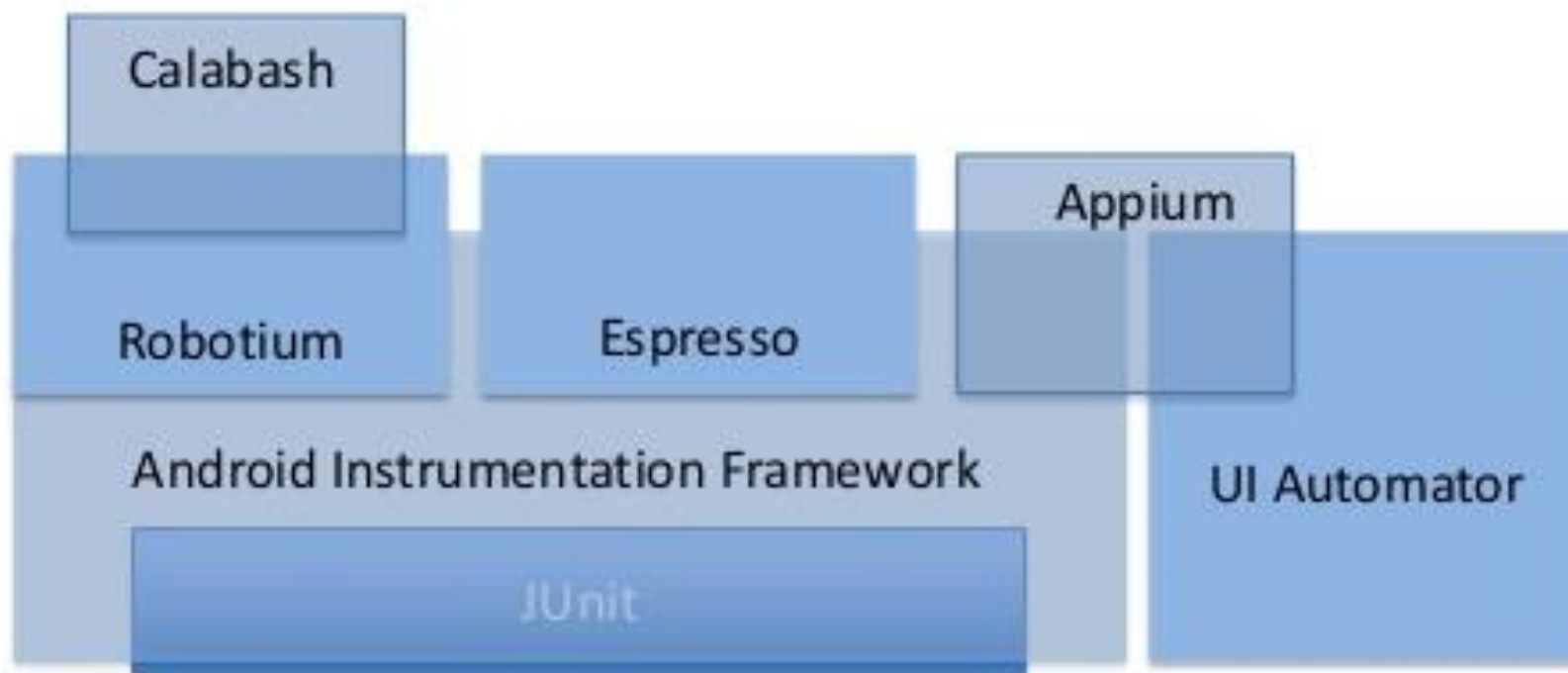
# 业界top框架综合对比

|                         | Robotium              | uiautomator              | Espresso            | Appium                 | Calabash         |
|-------------------------|-----------------------|--------------------------|---------------------|------------------------|------------------|
| Android                 | Yes                   | Yes                      | Yes                 | Yes                    | Yes              |
| iOS                     | No                    | No                       | No                  | Yes                    | Yes              |
| Mobile web              | Yes<br>(Android)      | Limited to x.y<br>clicks | No                  | Yes<br>(Android & iOS) | Yes<br>(Android) |
| Scripting<br>Language   | Java                  | Java                     | Java                | Almost any             | Ruby             |
| Test creation<br>tools  | Testdroid<br>Recorder | UI Automator<br>viewer   | Hierarchy<br>Viewer | Appium.app             | CLI              |
| Supported<br>API levels | All                   | 16 =>                    | 8, 10, 15-19        | All                    | All              |
| Community               | Contributors          | Google                   | Google              | Active                 | Pretty quiet     |

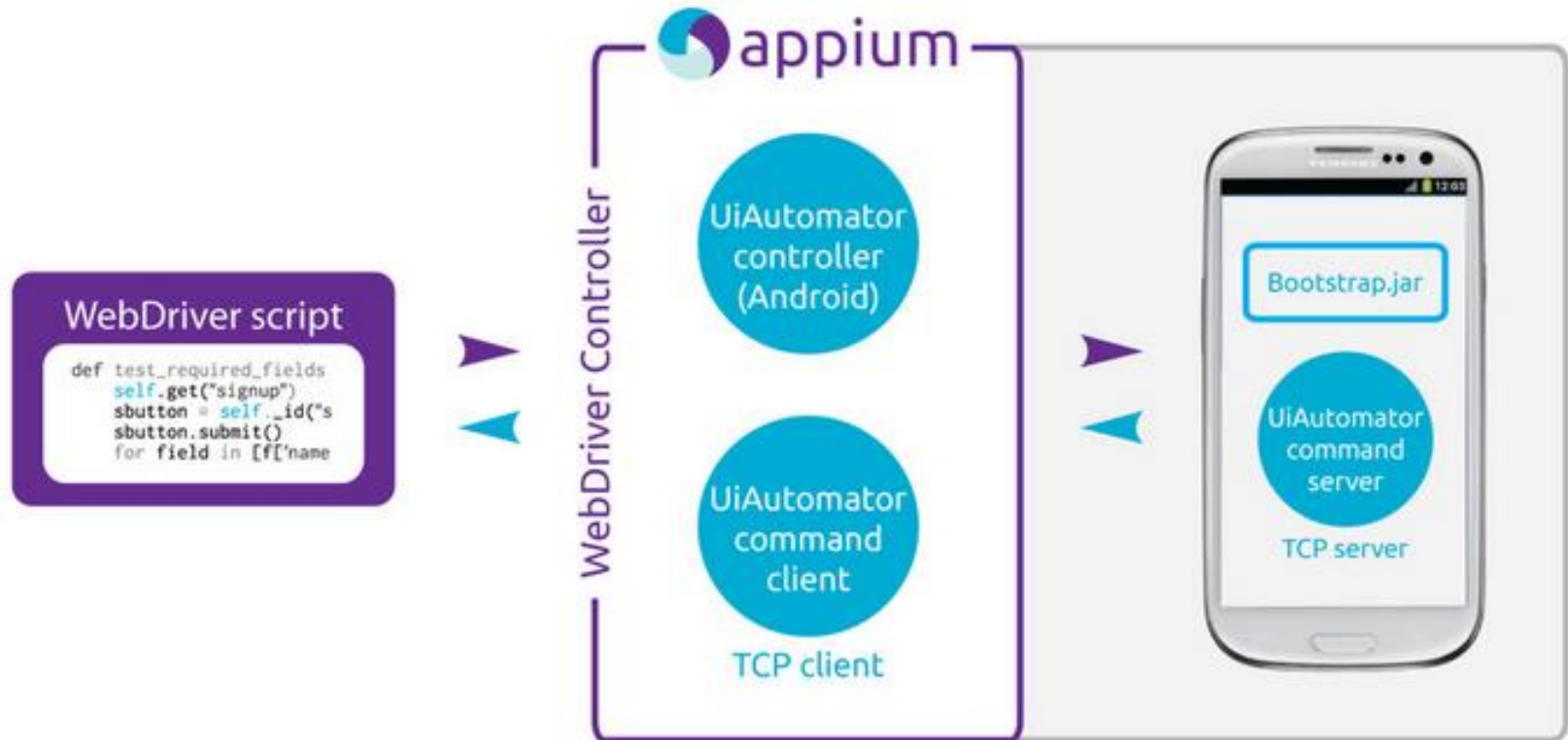
# Appium的优点

- 跨架构, native hybrid webview
- 跨设备, android ios firefofos
- 跨语言, java python ruby nodejs php
- 跨app, 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

# 基于Android的测试框架

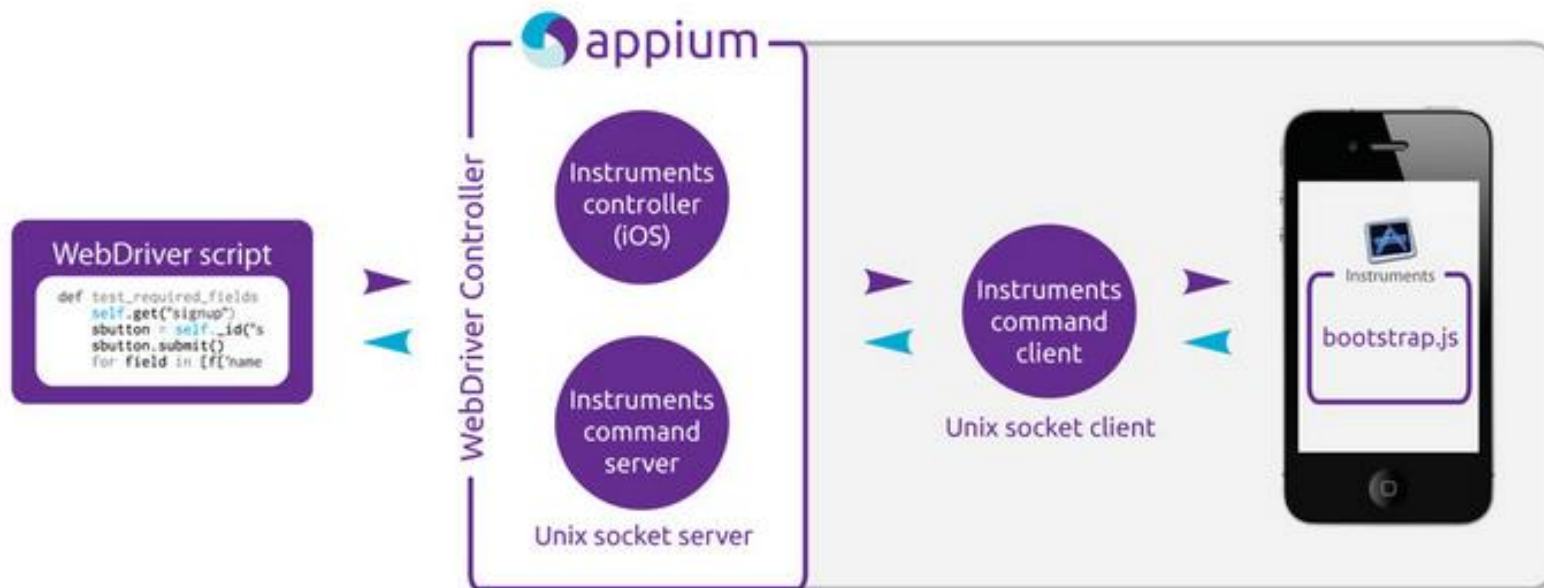


# Appium在Android上的架构





# Appium在iOS上的架构



# Appium架构

- Appium在4.1以上使用uiautomator
- 4.1以下使用selendroid
- Android上使用了instrumentation和uiautomator两套技术
- iOS使用uiautomation
- 同时还支持firefox, 并可扩展其他平台

# Appium的使用

- 安装nodejs
- 使用npm安装appium
- Appium启动

```
$ appium
info: Welcome to Appium v0.11.3 (REV 4a0cb4cc9e974d8d584ae98fef41eee0c067c081)
info: Appium REST http interface listener started on 0.0.0.0:4723
info - socket.io started
```

# Appium的启动参数

- Appium -help 可以看到一些可选参数
  - 通过参数可以配置如下内容
  - 监听ip和端口
  - 是否使用selendroid
  - 配置android和ios相关的参数
  - 启动app
- 
- 默认开启4723端口接受webdriver请求
  - 默认开启4724用于和android通讯

# Remote webdriver调用

```
1 import os
2 from selenium import webdriver
3
4 # Returns abs path relative to this file and not cwd
5 PATH = lambda p: os.path.abspath(
6     os.path.join(os.path.dirname(__file__), p)
7 )
8
9 desired_caps = {}
10 desired_caps['device'] = 'Android'
11 desired_caps['browserName'] = ''
12 desired_caps['version'] = '4.2'
13 desired_caps['app'] = PATH('../..../sample-code/apps/ContactManager/ContactManager.apk')
14 desired_caps['app-package'] = 'com.example.android.contactmanager'
15 desired_caps['app-activity'] = '.ContactManager'
16
17 driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
18
19 el = driver.find_element_by_name("Add Contact")
20 el.click()
21
22 textfields = driver.find_elements_by_tag_name("textfield")
23 textfields[0].send_keys("My Name")
24 textfields[2].send_keys("someone@somewhere.com")
25
26 driver.find_element_by_name("Save").click()
27
28 driver.quit()
```

# 视频演示

- [http://v.youku.com/v\\_show/id\\_XNjM0ODIxNTUy.html](http://v.youku.com/v_show/id_XNjM0ODIxNTUy.html)

# Appium的加载流程

- 调用android adb完成基本的系统操作
- 向android上部署bootstrap.jar包并启动
- Forward android的端口到pc机器上
- Pc上监听端口接受请求，使用webdriver协议
- 分析命令并转通过forward的端口发给bootstrap.jar
- Bootstrap.jar接受请求并把命令发给uiautomator或者插桩体系

# 具体流程举例

- Webdriver的脚本执行 `Element.send_keys`
- 发送到remote webdriver  
`/wd/hub/session/xxx/keys`
- pc处理请求并转发给android和iphone
- Bootstrap.jar调用uiautomator
- Uiautomator调用settext方法



# Appium设计理念

- 使用webdriver协议与用户交互
- XPath抽象定位
- 底层调用各自平台的自动化技术
- 类似selenium的架构

# Webdriver协议

- 使用webdriver协议，成熟标准，应用广泛
- 它即将成为w3c业界标准
- 跨语言, 不限制测试框架
- Remote WebDriver方式支持云测和集群模式
- PC端运行可以支持各种runtimes，不受限手机环境

# Appium支持主流语言







appium / appium

branch: master appium / sample-code / examples / +

Updated Sauce Java samples to use Sauce Java helper libraries, which ...

 **rossrowe** authored 20 hours ago

--

|  |  |
|--|--|
|  dotnet   | update all examples to use correct desired caps                          |
|  java     | Updated Sauce Java samples to use Sauce Java helper libraries, which ... |
|  node   | update all examples to use correct desired caps                          |
|  php    | update all examples to use correct desired caps                          |
|  python | update all examples to use correct desired caps                          |
|  ruby   | add chinese docs and a ruby xunit testcase sample for android            |

# webdriver的capabilities配置

- App: 本地路径或者网络下载地址
- browserName: 兼容selenium, 并不使用
- Device: ios android firefoxos
- Version: Android API version, iOS Version
- Launch: 是否让appium自动启动app

# Capabilities的配置例子

- Android2.3
- Android4.2
- iOS6.1

```
{  
  device: 'Android',  
  browserName: '',  
  version: '4.2',  
  app: myApp,  
  'app-package': myAppPackage,  
  'app-activity': myAppActivity  
}
```

```
{  
  device: 'Selendroid',  
  browserName: '',  
  version: '2.3',  
  app: myApp,  
  'app-package': myAppPackage,  
  'app-activity': myAppActivity  
}
```

```
{  
  device: 'iPhone Simulator',  
  browserName: '',  
  version: '6.1',  
  app: myApp  
}
```

# 演示代码

```
desired_caps = {}
desired_caps['device'] = 'Android'
desired_caps['browserName'] = ''
desired_caps['version'] = '4.3'
desired_caps['app'] = PATH('../../sample-code/apps/ApiDemos/bin/ApiDemos-debug.apk')
desired_caps['app-package'] = 'com.example.android.apis'
desired_caps['app-activity'] = '.ApiDemos'

driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

```
def capabilities
  {
    'browserName' => 'android',
    'platform' => 'linux',
    'version' => '4.1',
    'app-activity'=> '.Settings',
    'app-package'=> 'com.android.settings'
  }
end

def init(data={})
  server_url = 'http://127.0.0.1:4723/wd/hub'
  driver = Selenium::WebDriver.for(:remote, :desired_capabilities => capabilities.merge(data),
  driver.manage.timeouts.implicit_wait = 20 # seconds
  driver
end
```

# 元素定位

- Android和iOS的原生控件以及webview控件映射为特定的命名, 比如把所有平台的按钮映射为button元素
- 把当前的结构树映射为html文档
- 通过xpath来选择元素
  
- 三种定位方式
- Name: id, label text
- Tag\_name: component type
- Xpath
  
- 定位元素建议采用PageObject模式, 使用XPath
- 不要在测试用例中使用XPath

# Xpath支持语法列表

- `//button`
- `//button[last()]`
- `//button[1]`
- `/button`
- `Button`
- `//button/text/webview`
- `//button[1]/text/webview[3]`
- `//button[@name='hi there']`
- `//button[@other_attr='hi there']`
- `//list/button[@name="hi there"]`
- `//button[contains(@other_attr, "hi")]`
- `//*[contains(@text, 'agree')]`
- `//*[@text='agree']`
- `//frame//button`



# Xpath映射关系查询

```
map.put("autocomplete", "autocompleteTextView");
map.put("button", "Button");
map.put("breadcrumbs", "FragmentBreadCrumbs");
map.put("calendar", "CalendarView");
map.put("checkbox", "CheckBox");
map.put("checked", "CheckedTextView");
map.put("chronometer", "Chronometer");
map.put("compound", "CompoundButton");
map.put("datepicker", "DatePicker");
map.put("dialerfilter", "DialerFilter");
map.put("digitalclock", "DigitalClock");
map.put("drawer", "SlidingDrawer");
map.put("expandable", "ExpandableListView");
map.put("extract", "ExtractEditText");
map.put("fragmenttabhost", "FragmentTabHost");
map.put("frame", "FrameLayout");
```

.....

# Xpath获取

- Appium inspector
- uiautomatorview
- android也可以使用selendroid的inspector

### UI Automator Viewer

- (2) EditText:请输入搜索关键字 [199,66][600,138]
      - (2) ImageButton {搜索} [616,66][696,138]

#### Node Detail

|              |                            |
|--------------|----------------------------|
| index        | 2                          |
| text         |                            |
| class        | android.widget.ImageButton |
| package      | com.taobao.taobao          |
| content-desc | 搜索                         |
| checkable    | false                      |
| checked      | false                      |
| clickable    | true                       |
| enabled      | true                       |
| focusable    | true                       |
| focused      | false                      |

# 元素交互

- Webdriver已经支持
  - 点击
  - 输入
  - 按键事件
- Mobile相关:
  - 拖动
  - 划动
  - 长按

# 点击与输入

WD.js:

```
driver.elementByName('Go', function(err, el) {  
  el.tap(function(err) {  
    driver.quit();  
  });  
});
```

Java:

```
driver.findElement(By.tagName("button")).sendKeys("Hi");  
  
WebElement element = findElement(By.tagName("button"));  
element.sendKeys("Hi");  
  
List<WebElement> elems = findElements(By.tagName("button"));  
elems.get(0).sendKeys("Hi");
```

Ruby:

```
@driver.find_element(:name, 'Go').click
```

Python:

```
driver.find_element_by_name('Go').click()
```

# 长按操作代码

Java:

```
WebElement element = wd.findElement(By.name("API Demo"));
JavascriptExecutor js = (JavascriptExecutor) wd;
HashMap<String, String> tapObject = new HashMap<String, String>();
tapObject.put("element", ((RemoteWebElement) element).getId());
js.executeScript("mobile: longClick", tapObject);
```

Python:

```
element = wd.find_element_by_xpath("your_element_xpath")
wd.execute_script("mobile: longClick",{ "touchCount": 1, "x": 0, "y": 300, "element":element.id })
```

# 通过execute\_script执行mobile方法

- Python:

```
driver.execute_script("mobile: tap", {"touchCount": "1", "x": "0.9", "y": "0.8", "element": element.id})
```

- Ruby:

```
@driver.execute_script 'mobile: tap', :x => 150, :y => 30
```

- Ruby:

```
b = @driver.find_element :name, 'Sign In'  
@driver.execute_script 'mobile: tap', :element => b.ref
```

- C#:

```
Dictionary<String, Double> coords = new Dictionary<string, double>();  
coords.Add("x", 12);  
coords.Add("y", 12);  
driver.ExecuteScript("mobile: tap", coords);
```

# 一个完整的例子

```
def test_settings
  #flick the screen until find the About phone item
  while @driver.find_elements(:xpath, '//text[@text="About phone"]').count==0
    begin
      @driver.execute_script 'mobile: flick', :startY=>0.9, :endY=>0.1
    rescue
    end
  end
  about=@driver.find_element(:xpath, '//text[@text="About phone"]')
  about.click
  #parent select, locate the container
  version_setting=@driver.find_element(:xpath, '//list/linear[4]/relative')
  #child select
  version_value=version_setting.find_element(:xpath, '//text[2]')
  #check the version, should be 4.1.2 or other version string
  assert_not_equal nil, version_value.text=~/[0-9\.]/
end
```



# Appium的灵活扩展

- Appium对webdriver协议进行了扩充
  - 通过利用execute\_script执行mobile操作
  - 通过自身服务增加了一些处理命令，比如安装，卸载，加载app
  - 也扩展支持一些平台专有的指令
- 
- Webdriver即将成为W3C标准
  - Appium在推动webdrive增强mobile特性

# 位置模拟演示

- 只支持android模拟器

```
driver=Mobile.new("app-package"=> "com.tencent.mobileqq", "app-activity"=> "com.
tencent.open.agent.AgentActivity").driver
driver.execute_script 'mobile: setLocation', :latitude=>40.0445, :longitude=>118
.3184
qq(driver)
driver.quit
```

# 原生,混合,web

- 三种主流的app设计模式
- 原生:
- 使用开发特定平台专用app
  
- 混合:
- Phonegap 百度轻应用
  
- Web:
- 使用html5技术开发
- jquery mobile, sencha touch

# 混合模式的自动化

- iOS

```
handles = driver.window_handles()
driver.window(handles[0])
driver.get("http://google.com")
driver.find_element_by_id("q").send_keys("my search")
driver.find_element_by_css("input[type=submit]").click()
driver.execute_script("mobile: leaveWebView")
```

- Android

```
driver.find_element_by_id('buttonStartWebview').click()
driver.window("WEBVIEW")
driver.find_element_by_css("input[type=text]").send_keys("hi")
driver.find_element_by_css("input[type=submit]").click()
driver.window("NATIVE")
```

# 移动测试环境规划

- 私有移动测试云环境
- 针对android ios等各个主要版本的测试环境
- 模拟器为主, 应付日常测试需求
  
- 公共云服务
- 百度MTC Testin Saucelabs
- 真机为主, 可以测试更多场景. 测试覆盖度高

# 移动云测试

- Saucelabs提供了appium的在线云环境
- 提供虚拟机
- 提供运行时的截图和录像
- 脚本执行非常简单
  
- Testin和MTC也各具特色

# 移动测试持续集成

- Appium运行于PC端
- 支持webdriver协议和Grid模式
- 可以实现无缝的持续集成
- 与底层单元测试互补, 实现分层自动化

# Appium对RF的支持

- Appium支持webdriver协议
- 可以使用RFT来设计测试用例
- 需要对appium目前的版本打补丁
- 已经提需求给官方, 后续会更好的支持
- 我目前提供的版本已经支持appium



# Robot的测试用例演示

- 打开淘宝, 输入小米手机并搜索

|   |                         |                                     |                                       |
|---|-------------------------|-------------------------------------|---------------------------------------|
| 1 | open app                | com.taobao.taobao                   | com.taobao.tao.homepage.MainActivity3 |
| 2 | Capture Page Screenshot |                                     |                                       |
| 3 | Input Text              | xpath=//textfield[@text='请输入搜索关键字'] | xiaomi \ shouji \                     |
| 4 | Capture Page Screenshot |                                     |                                       |
| 5 | Click Element           | xpath=//frame/linear/imagebutton[2] |                                       |
| 6 | Capture Page Screenshot |                                     |                                       |
| 7 |                         |                                     |                                       |

# Robot Test Report

Generated  
 20131120 22:53:51 GMT +08:00  
 5 minutes 54 seconds ago

## Summary Information

**Status:** [All tests passed](#)  
**Start Time:** 20131120 22:53:23.463  
**End Time:** 20131120 22:53:51.313  
**Elapsed Time:** 00:00:27.850  
**Log File:** [log.html](#)

## Test Statistics

| Total Statistics               | Total | Pass | Fail | Elapsed  | Pass / Fail   |
|--------------------------------|-------|------|------|----------|---|
| <a href="#">Critical Tests</a> | 1     | 1    | 0    | 00:00:26 | <div style="width: 100%; height: 10px; background-color: green;"></div> |
| <a href="#">All Tests</a>      | 1     | 1    | 0    | 00:00:26 | <div style="width: 100%; height: 10px; background-color: green;"></div> |

| Statistics by Tag | Total | Pass | Fail | Elapsed | Pass / Fail   |
|-------------------|-------|------|------|---------|---|
| No Tags           |       |      |      |         | <div style="width: 0%; height: 10px; background-color: green;"></div> |

| Statistics by Suite          | Total | Pass | Fail | Elapsed  | Pass / Fail   |
|------------------------------|-------|------|------|----------|---|
| <a href="#">Robot</a>        | 1     | 1    | 0    | 00:00:28 | <div style="width: 100%; height: 10px; background-color: green;"></div> |
| <a href="#">Robot.Appium</a> | 1     | 1    | 0    | 00:00:28 | <div style="width: 100%; height: 10px; background-color: green;"></div> |

[-] **TEST CASE: taobao**

**Full Name:** Robot.Appium.taobao

**Start / End / Elapsed:** 20131120 22:53:25.269 / 20131120 22:53:51.294 / 00:00:26.025

**Status:** **PASS** (critical)

[+] **KEYWORD:** open app com.taobao.taobao, com.taobao.tao.homepage.MainActivity3

[+] **KEYWORD:** Selenium2Library.Capture Page Screenshot

[+] **KEYWORD:** Selenium2Library.Input Text xpath=//textfield[@text="请输入搜索关键字"], xiaomi \ shouji \

[+] **KEYWORD:** Selenium2Library.Capture Page Screenshot

[+] **KEYWORD:** Selenium2Library.Click Element xpath=//frame/linear/imagebutton[2]

[+] **KEYWORD:** Selenium2Library.Capture Page Screenshot

[+] **TEARDOWN:** quit

KEYWORD: open app com.taobao.taobao, com.taobao.tao.homepage.MainActivity3

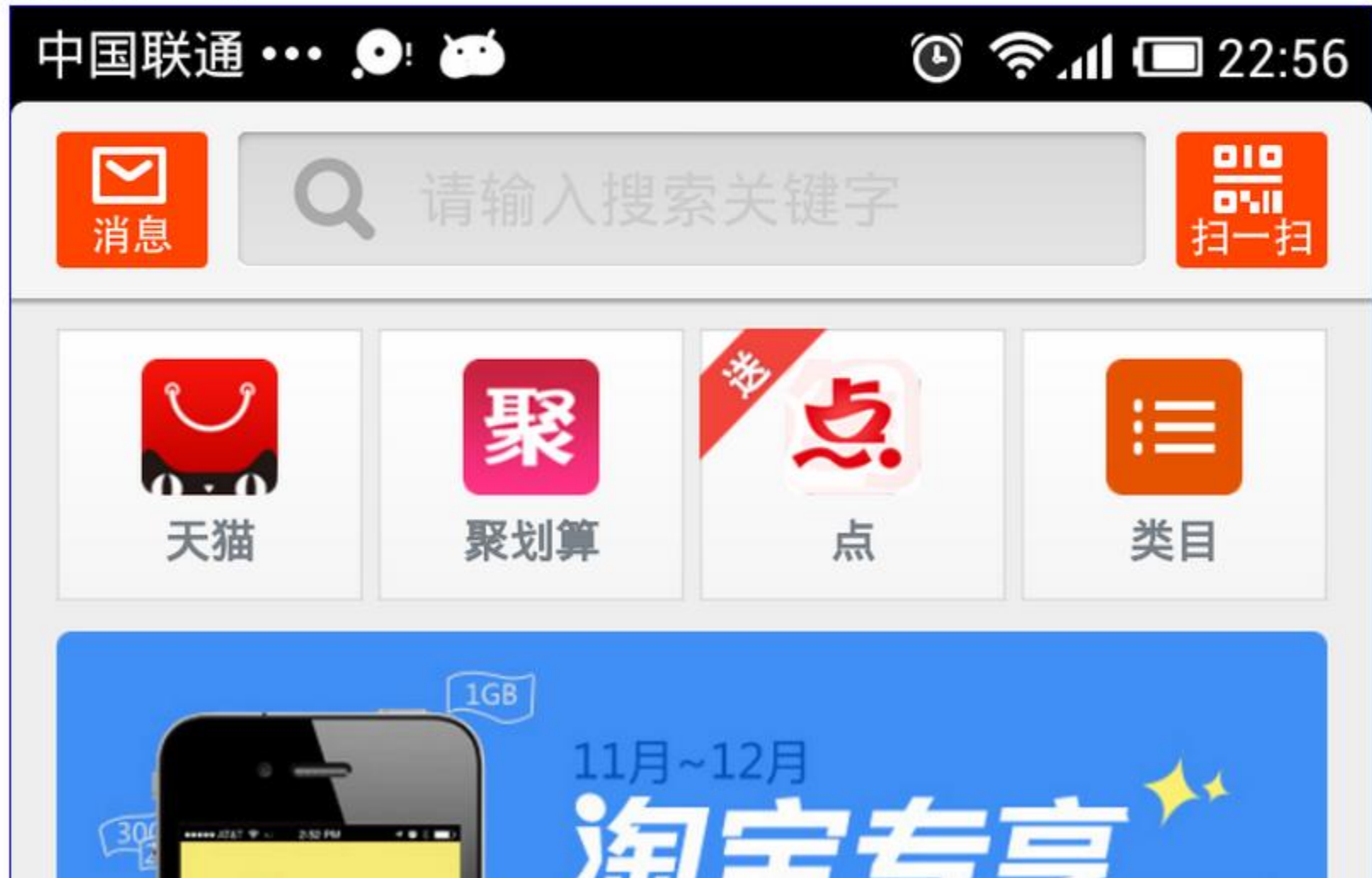
KEYWORD: Selenium2Library.Capture Page Screenshot

Documentation: Takes a screenshot of the current page and embeds it into the log.

Start / End / Elapsed: 20131120 22:53:34.196 / 20131120 22:53:35.995 / 00:00:01.799

22:53:34.200 DEBUG GET http://127.0.0.1:4723/wd/hub/session/07b9a682-28d4-4db7-ac59-81fbba9070cd/screenshot {"sessionId": "07b9a6

22:53:35.992 INFO



KEYWORD: Selenium2Library.Input Text xpath=//textfield[@text='请输入搜索关键字'], xiaomi \ shouji \

KEYWORD: Selenium2Library.Capture Page Screenshot

Selenium2Library.Input Text

Documentation: Takes a screenshot of the current page and embeds it into the log.

Start / End / Elapsed: 20131120 22:53:44.496 / 20131120 22:53:45.529 / 00:00:01.033

22:53:44.505 DEBUG GET http://127.0.0.1:4723/wd/hub/session/07b9a682-28d4-4db7-ac59-81fbba9070cd/screenshot {"sessionId": "07b9a682-28d4-4db7-ac59-81fbba9070cd"}

22:53:45.526 INFO



KEYWORD: Selenium2Library.Capture Page Screenshot

Documentation: Takes a screenshot of the current page and embeds it into the log.

Start / End / Elapsed: 20131120 23:20:11.080 / 20131120 23:20:12.981 / 00:00:01.901

23:20:11.088 DEBUG GET http://127.0.0.1:4723/wd/hub/session/5f1d160b-6bd7-416d-b8ba-c0520e2b8e8b/screenshot {"sessionId": "c0520e2b8e8b"}

23:20:12.978 INFO



# Appium应用注意事项

- Xpath的支持有问题，并不是完全的xpath功能
- 一些app比如qq不允许简单的am start，需要提升权限，比如让appium扩展支持su
- Appium长时间运行不太稳定，建议在执行测试前重启appium

# 展望appium

- 目前还未到正式版，很多地方需要完善，但是代码清晰，而且还有saucelabs的工程师维护，代码质量有保证
- 架构非常优秀，扩展性很好，不会限制用户使用特定的语言
- 已经成为mobile自动化事实上的标准
- Saucelabs云服务已经支持appium，MTC TESTIN后续也可能跟进，使appium成为标准的云端解决方案



# 与appium互动里程碑

- Done
  - 翻译了appium的文档
  - 补充了测试用例类型
  - 提交发现的issue
- Doing
  - 改进appium
- 目标:
  - 和saucelabs一起维护appium
  - 技术与商业合作
- @恒温 @monkey @国文

# 更多技术爆料敬请期待

- Appium自动遍历
- Appium录制回放技术
- Android调试分析技术
- Android应用逆向分析与破解