

戏说APP自动化



本节内容

- 为什么要做APP自动化
- APP自动化与WEB自动化的比较



APP自动化



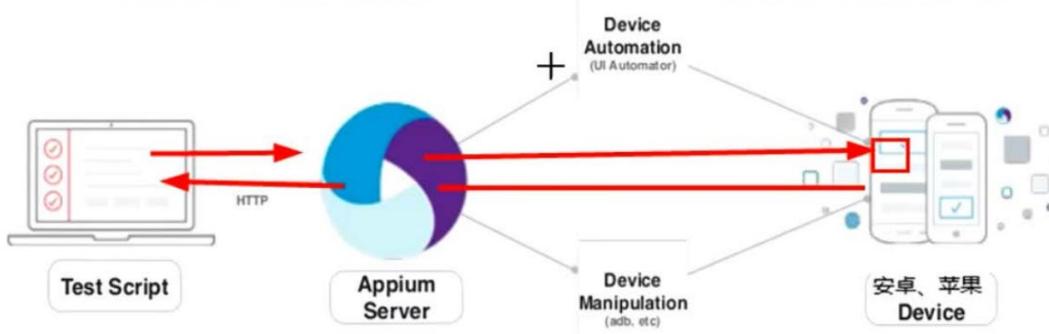
为什么要做APP自动化

- 用户对App产品的质量和用户体验要求越来越高
- 互联网公司对移动端产品投入的重视越来越高
- 互联网产品迭代快
- 繁琐的回归测试

APP自动化与WEB自动化

	WEB	APP
启动方式	浏览器启动，可以多线程	呼出APP启动，只能单线程
客户端信息	通过启动webdriver不同的浏览器类	需要制定desired_caps内容，包含设备信息
元素定位	name, id, class_name, css, xpath等	不支持css_selector和linktext
元素操作	可以支持通过js实现	不支持js

移动自动化设计



APP自动化工具演进史



本节内容

- APP自动化测试工具的前世今生



移动互联网”进化“



APP自动化与WEB自动化

	Instrumentation	Robotium
优点	Android 自带的一个测试框架，是很多其它测试框架的基础	基于Instrumentation框架开发的一个更强的框架. 对常用的操作进行了易用性的封装. 用于开发功能性、系统和验收测试场景 容易在最短的时间内编写测试脚本，易用性高
缺点	Instrumentation不支持跨应用，封装不完善	不能处理flash和web组件 不支持iOS设备

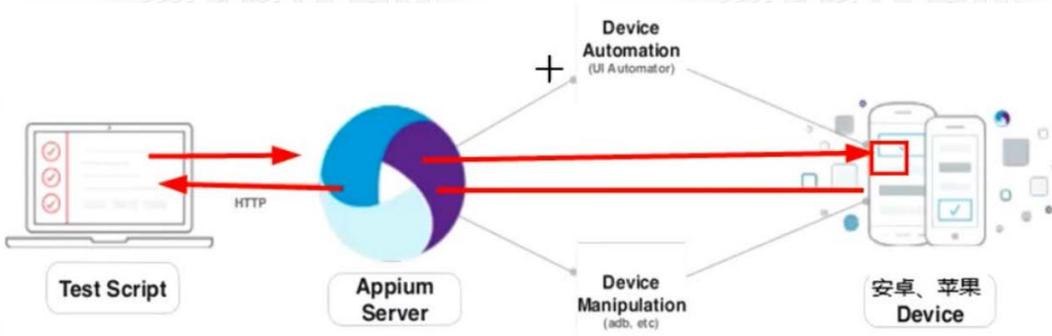
APP自动化与WEB自动化

	UIAutomator	Espresso
优点	谷歌提供的测试框架。这是一个包含API的Java库，用来创建功能性UI测试，还有运行测试的执行引擎。给JUnit测试案例特权。库由谷歌社区支持和维护	Google的开源自动化测试框架。规模更小、更简洁，API更加精确，编写测试代码简单，容易快速上手
缺点	仅支持android 4.1及以上，完美支持JAVA，对其他语言支持不好	基于Instrumentation的，不能跨App

APP自动化与WEB自动化

	Appium	Calabash
优点	开源的、跨平台的自动化测试工具。通过驱动苹果的UIAutomation和Android的UiAutomator框架来实现的双平台支持，同时绑定了Selenium WebDriver用于老的Android平台测试。开发者可以使用WebDriver兼容的任何语言编写测试脚本	适用于iOS和Android开发者的跨平台app测试框架，可用来测试屏幕截图、手势和实际功能代码。Calabash开源免费并支持Cucumber语言
缺点	--	测试耗费时间，默认先安装app；除了ruby对其他语言支持不好

APPIUM原理



测试框架appium环境配置



本节内容

□APPIUM环境配置



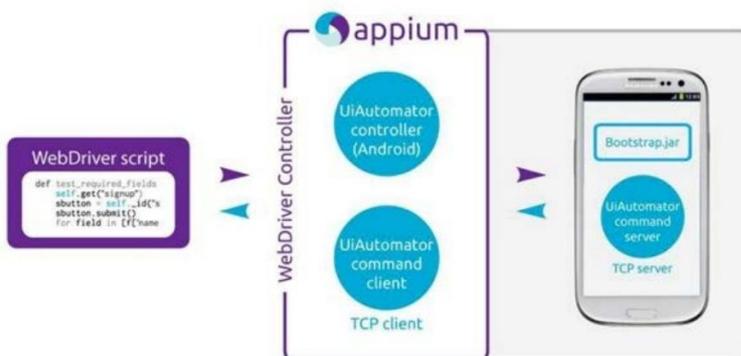
Android环境搭建

- JAVA环境安装
- Android SDK环境搭建

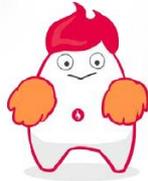
Appium环境搭建

- nodejs安装配置
- Appium安装
- Appium-Python-Client

Appium原理



appium快速上手

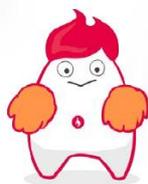


本节内容

□通过APPIUM完成第一个自动化脚本（计算器）



appium Capability详解



本节内容

□ 什么是Capability

□ Capability配置讲解



什么是Capability

- 本次测试是启动浏览器还是启动移动app?
- 是android还是ios?
- android的版本是?
- app的配置是?

Capability配置详解

```
PATH = lambda p: os.path.abspath(os.path.join(os.path.dirname(__file__), p))
capabilities = {}
capabilities['platformName'] = 'Android' #设置平台
capabilities['platformVersion'] = '4.1' #系统版本
capabilities['deviceName'] = '127.0.0.1:62001' #设备id
capabilities['autoLaunch'] = 'true' #是否自动启动
capabilities['app'] = PATH('../apps/mukewang.apk') #安装包路径, 放在该py文件的目录下
capabilities['appPackage'] = 'cn.com.open.mooc' #包名
capabilities['appActivity'] = 'cn.com.open.mooc.aindex.splash.MCSplashActivity' #启动的activity
self.driver = webdriver.Remote('http://localhost:4723/wd/hub', capabilities)
```

appium定位手机元素



本节内容

□常用元素定位方法（ID、ClassName、Xpath）

□层级定位

□内嵌H5定位

□滑动定位

□uiautomator text定位



appium模拟手机操作



本节内容

□ 键盘操作

□ 触摸动作

□ 其他操作



实战appium自动化测试项目



本节内容

□ 慕课网APP自动化实战操作



app自动化总结与展望

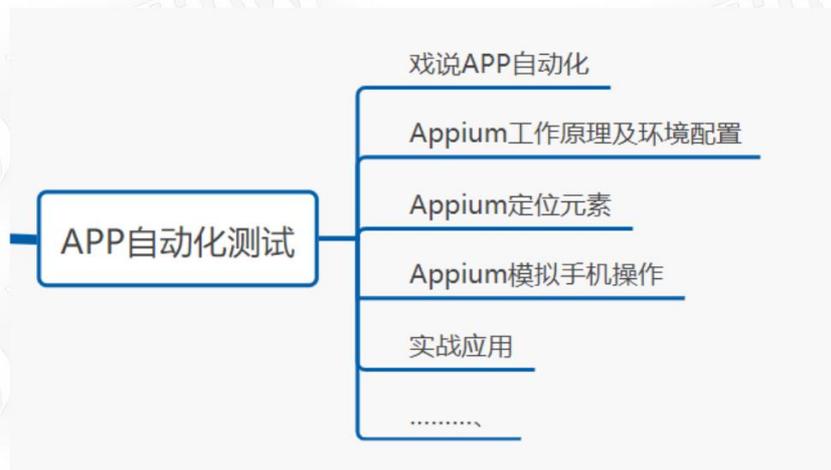


本节内容

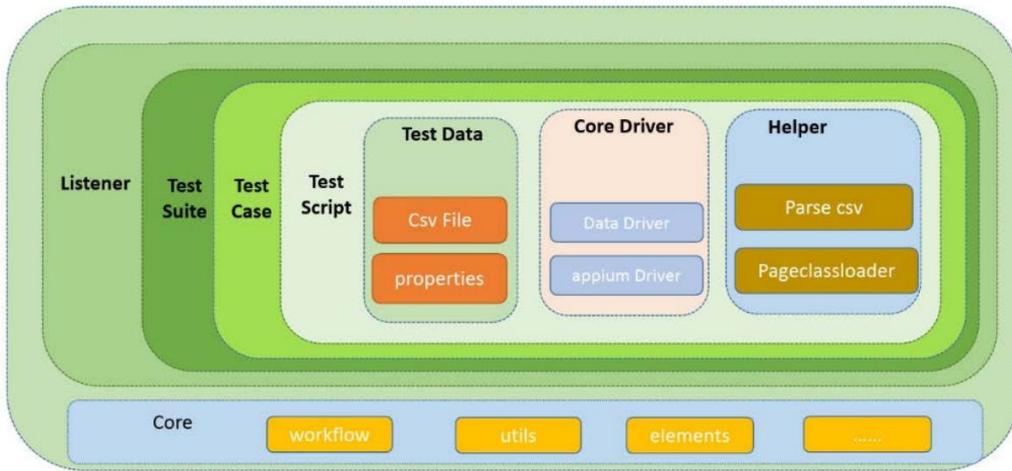
- app自动化总结与展望



APP自动化总结



APP自动化框架设计思路



APP自动化展望

- AI+自然语言
- 分布式自动化
- 从自动化到无人化