

# Appium 错误排查与日志分析



扫码试看/订阅

《移动端自动化测试实战》视频课程

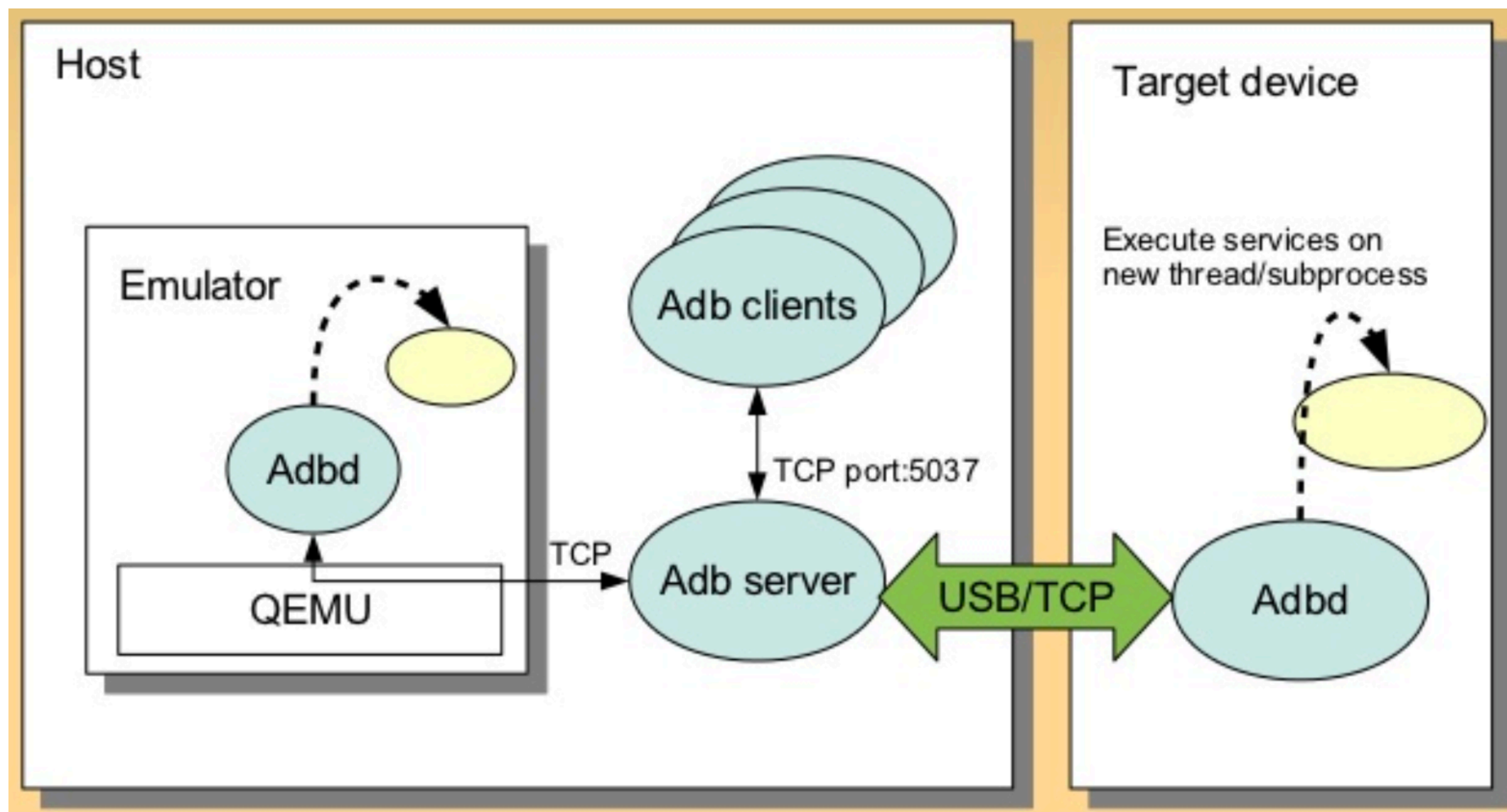
# Appium Android 自动化流程分析

- `appium -g <log file path>`
- shell 命令
- WebDriver 请求
- ChromeDriver 日志

# 调试分析方法

- Appium Log
  - 清晰记录了所有的请求和结果以及底层的一些错误堆栈
- 分析界面数据
  - 使用 getPageSource 获取界面的完整 DOM 结构
  - 利用 XPath 获取所有匹配的元素
- 脚本外调试
  - 借助于 curl 命令从脚本外探测界面数据

# adb 知识与排错



# adb 排错

- adb client: 普通的 adb 命令, 使用 shell 脚本代替
- adb server:
  - pc 上的一直开启的 server 进程
  - adb 命令通过 adb 协议与 adb server 通讯
  - 通过代理转发结合 capability

# Appium 原理与 JSONWP 协议分析

# WebDriver 协议与手工模拟

- WebDriver 协议基础知识要懂
- session\_id 获取
  - `session_id=`curl http://127.0.0.1:4723/wd/hub/sessions | awk -F\" '{print $8}'``
- element id 获取
  - `curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/elements --data-binary '{"using":"xpath","value":"//*[@class=\"android.widget.Toast\"]}' -H "Content-Type: application/json;charset=UTF-8"`
- 元素属性获取
  - `curl http://127.0.0.1:4723/wd/hub/session/$session_id/element/$element_id/attribute/text`
- 元素动作
  - `curl -X POST http://127.0.0.1:4723/wd/hub/session/$session_id/element/$element_id/click`



# Appium 源码分析

# 所有源代码

- appium server
  - Node.js 系列包
  - adb shell
  - ChromeDriver
- 底层引擎
  - Uiautomator Java
  - WDA
  - selenium
- appium client: Python、Java

# Appium 二次封装

# 自定义 Appium Server

- git clone
- npm install

# 重新编译 Uiautomator

- git clone
- Gradle

# 自动遍历测试方法与常见技术介绍

# 业务常见问题举例

- 功能问题
  - App 某界面崩溃
  - App 某接口有报错
  - 股票详情页中特定信息字段内容丢失或者数据异常
  - 微信分享不可用
- 兼容性问题：
  - 用户网络慢时发出请求后退出当前页面发生崩溃
  - 某些界面在 4.4 和 5.0 的系统上操作体验不同

# 测试痛点

- 快速迭代中传统的基于用例维护的自动化用例使用不当
  - 没有采用合理的分层测试体系，尝试用UI自动化覆盖所有测试场景
  - 没有采用 PageObject 模式导致自动化用例维护成本大
  - 对自动化框架掌握程度不够
- 测试内容太多导致手工测试无法充分覆盖
  - 界面字段正确性：比如股票相关数据变化，共数十个字段
  - 接口正确性：后端接口传输数据的变化和内容
  - 专项测试回归难度大：内存泄漏、健壮性测试、弱网等测试过程太多
  - 回归工作量大：不回归又会漏测



# 对测试手段的改进

- 分层测试：
  - 合理划分接口测试与UI测试的比例与业务边界
  - 自动化测试覆盖主要业务流程
- 手工测试：覆盖新功能测试，探索式测试
- 遍历测试手段
  - code less：用例维护成本降低到最低
  - automate：尽可能的自动化覆盖回归业务

# 自动遍历测试的需求

- 自动遍历的需求
  - 可控：可以定义遍历的路径
  - 可定制：可实现自动输入、自动滑动等基础行为
- 结果分析
  - 点击前后的截图对比
  - 结果的数据建模.
  - 新老版本 UI Diff
  - App 结构思维导图展示

# monkey 工具的缺点

- 不支持业务行为定制：
  - 经常点到外部的 app
  - 经常点击到注销和退出
  - 无法灵活控制

# 常见遍历工具与技术

- Google Android 原生 monkey
- 百度 SmartMonkey
- 腾讯 NewMonkey
- testerhome vigossjjj smart\_monkey
- 阿里 Macaca 的 NoSmoke
- 头条的 zhangzhao Maxim
- 霍格沃兹测试学院 seveniruby AppCrawler
- Google 的 App Crawler

# Monkey Maxim 遍历测试工具

# Android 官网对 monkey 工具的定义

## UI/Application Exerciser Monkey



The Monkey is a program that runs on your [emulator](#) or device and generates pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events. You can use the Monkey to stress-test applications that you are developing, in a random yet repeatable manner.



### Overview

---

The Monkey is a command-line tool that you can run on any emulator instance or on a device. It sends a pseudo-random stream of user events into the system which acts as a stress test on the application software you are developing.

The Monkey includes a number of options, but they break down into four primary categories:

- Basic configuration options, such as setting the number of events to attempt.
- Operational constraints, such as restricting the test to a single package.
- Event types and frequencies.
- Debugging options.

# monkey 与 Maxim

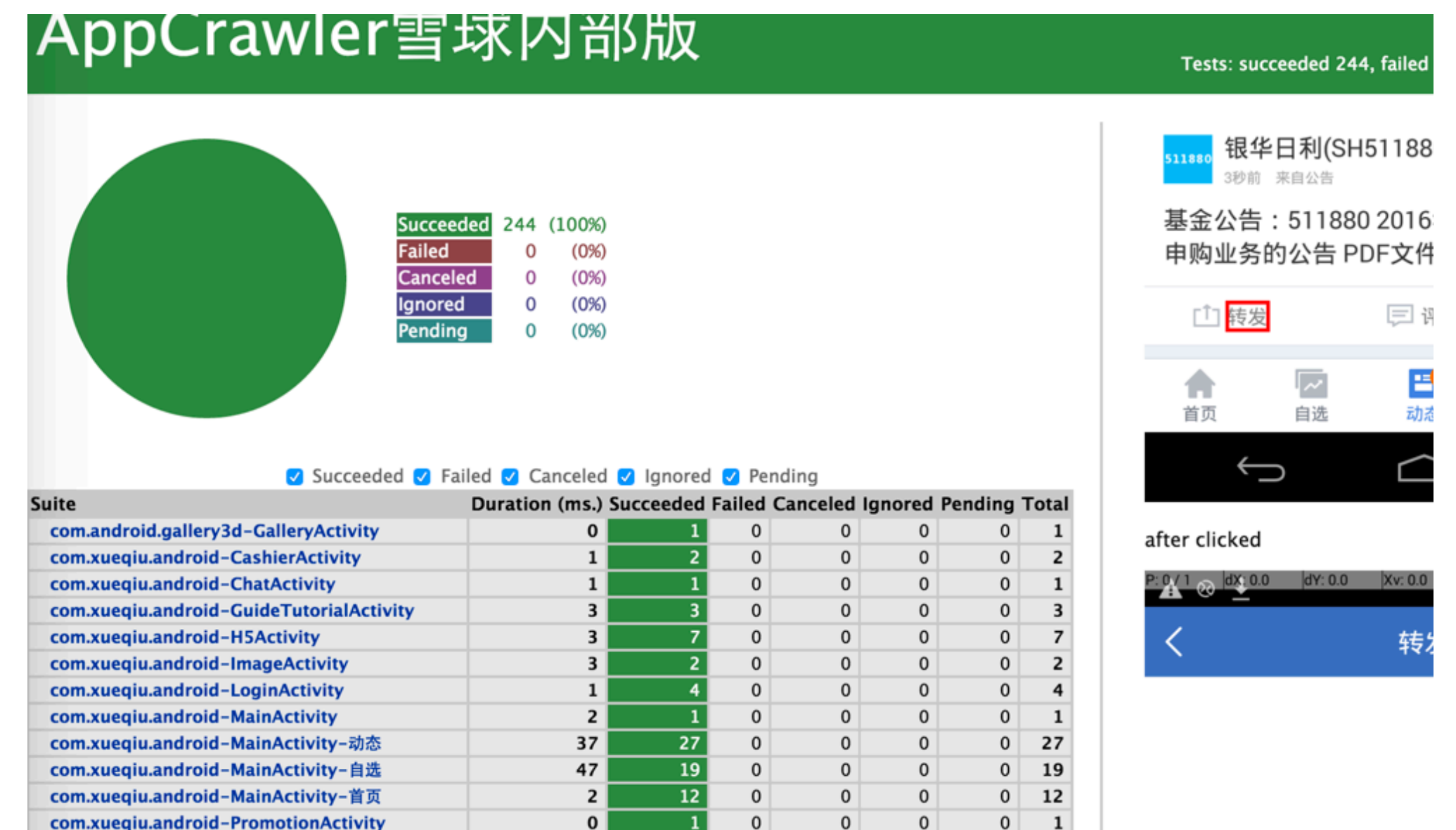
- monkey: 开源、Android 标准测试工具
- Maxim: 未开源, 改进版的 monkey, 提供预编译包

# AppCrawler 跨平台自动遍历测试



# 开源项目地址

- <https://github.com/seveniruby/AppCrawler>



# 与其他框架的关系

- AppCrawler 底层引擎
  - Appium
  - ADB
  - Macaca
  - selenium
- Appium 底层引擎
  - WDA
  - Uiautomator2

# AppCrawler 环境要求

- AppCrawler 2.4: java8 appium 1.8.x
- 下载方法: 百度网盘直接下载
- 从源代码编译: mvn 编译即可
- 后面提到的 AppCrawler 命令或者 `java -jar appcrawler.jar` 皆表示用 `jar -jar` 命令执行对应版本的 jar 包文件, 比如 `java -jar appcrawler-2.4.0-jar-with-dependencies.jar`

# quick start

- 启动 Appium
- 启动模拟器或者连接真机
- 开始自动遍历
- `appcrawler --capability "appPackage=com.xueqiu.android,appActivity=.view.WelcomeActivityAlias"`

# AppCrawler 基于配置文件的运行模式

# 生成样板配置示例

- `java -jar appcrawler.jar --demo`
- 会在当前目录下生成一个 `demo.yml`

# 执行方式

- 参数模式
  - `java -jar <appcrawler.jar路径> --demo`
  - `appcrawler --capability "appPackage=com.xueqiu.android,appActivity=.view.WelcomeActivityAlias"`
- 配置文件 [推荐方式]
  - `java -jar <appcrawler.jar路径> \`
  - `-c example.yml \`
  - `--capability "appPackage=com.xueqiu.android,appActivity=.view.WelcomeActivityAlias" \`
  - `-o /tmp/xueqiu/1`

# 执行参数与配置文件

- capability 设置：与 Appium 完全一致
- testcase：用于启动 App 后的基础测试用例
- selectedList：遍历范围设定
- triggerActions：特定条件触发执行动作的设置
- 执行参数比配置文件优先级别高



# capability设置示例

- appPackage: "com.xueqiu.android"
- appActivity: ".view.WelcomeActivityAlias"
- app: ""
- appium: "http://127.0.0.1:4723/wd/hub"
- noReset: true
- automationName: uiautomator2
- dontStopAppOnReset: true

# 自动化支持 TestCase

- TestCase 的完整形态
  - given: 所有的先决条件
  - when: 先决条件成立后的行为
  - then: 断言集合
- TestCase 的简写形态
  - XPath: 对应 when 里的 XPath
  - action: 对应 when 的 action

```
autoCrawl: true
testcase:
  name: demo1
  steps:
    - when:
      xpath: //*
      action: driver.swipe(0.5, 0.8, 0.5, 0.2)
    - when:
      xpath: //*
      action: driver.swipe(0.5, 0.2, 0.5, 0.8)
    - xpath: 自选
      action: click
      then:
        - //*[contains(@text, "港股")]
```

# 定位模式 XPath

- xpath
  - `//*[@resource-id='xxxx']`
  - `//*[contains(@text, '密码')]`
- 正则
  - `^确定$`
  - `^.*输入密码`
- 包含
  - 密码
  - 输入
  - 请

# 动作支持 action

- "" 只是截图记录
- back 后退
- backApp 回退到当前的 app 默认等价于 back 行为 可定制
- monkey 随机事件
- xxx() 执行代码
  - Thread.sleep(3000)
  - driver.swipe(0.9, 0.5, 0.1, 0.5)
- click
- longTap
- 非以上所有行为是输入 xx ddd

# 自动遍历支持

- selectedList: 需要被遍历的元素范围
- firstList: 优先被点击
- lastList: 最后被点击
- tagLimitMax: 同祖先（同类型）的元素最多点击多少次
- backButton: 当所有元素都被点击后默认后退控件定位
- blackList: 黑名单
- maxDepth: 6 遍历的最大深度

# 触发器

- triggerActions:
- 需要特定次数的触发动作
- 通常用于处理弹框
  - TestCase: 指定具体按钮
  - action: 动作
  - times: 规则的使用次数

```
triggerActions:  
- xpath: 请输入手机号  
  action: 15600534760  
  times: 1  
- xpath: //*[@password='true']  
  action: hys12345678  
  times: 1  
- xpath: button_next  
  action: click  
  times: 1  
autoGrow: true
```



扫码试看/订阅

《移动端自动化测试实战》视频课程