

AppCrawler工具

思寒

业务背景



雪球是什么

公司介绍

加入我们

雪球形象



买什么股票好，用雪球APP



 iOS下载

 Android下载

为什么做这个工具

- 快速迭代中自动化用例吃力
- 断言不可知. 颜色, 界面字段, 接口返回内容
- 颜色和尺寸的错乱只能靠人来保证?

设计理念

- 测试执行
 - 自动遍历
 - 自动化
- 测试分析
 - UI Diff [完善中]
 - 接口数据截取分析 [完成]
 - logcat与syslog收集 [完成]
 - 内存泄漏结合 [完成]
 - 代码流Diff [进行中]
 - 智能分析兼容性问题 [未开始]
 - 与云测平台结合 Testin 百度MTC 阿里MQC [进行中]

与Appium关系

- 升维 AppCrawler
- 同维 Macaca iTestin
- 降维 xctest Robotium ...


```
java -jar target/scala-2.11/appcrawler-1.3.0.jar
```

AppCrawler 1.3.0

app爬虫，用于自动遍历测试。支持Android和iOS，支持真机和模拟器

灵感来源：晓光 泉龙 杨榕 雪球测试团队出品

移动测试技术交流 <https://testerhome.com>

Usage: appcrawler [sbt] [options] <args>...

-a <value> | --app <value>

Android或者iOS的文件地址，可以是网络地址，赋值给appium的app选项

-c <value> | --conf <value>

配置文件地址

-p <value> | --platform <value>

平台类型android或者ios，默认会根据app后缀名自动判断

-t <value> | --maxTime <value>

最大运行时间。单位为秒。超过此值会退出。默认最长运行3个小时

-u <value> | --appium <value>

appium的url地址

-o <value> | --output <value>

遍历结果的保存目录。里面会存放遍历生成的截图，思维导图和日志

--capability k1=v1,k2=v2...

appium capability选项，这个参数会覆盖-c指定的配置模板参数，用于在模板配置之上的参数微调

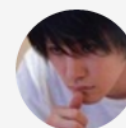
-v | --verbose

是否展示更多debug信息

--help

AppCrawler 自动遍历工具 1.2.1 版本

Appium · seveniruby · 于 3 月前发布 · 最后由 franky 于 2 天前回复 · 2456 次阅读



第四届线下沙龙回顾

3月份高鹏 徐世钊 mingway在阿里巴巴举办了第四届TesterHome线下沙龙
我在沙龙中分享了AppCrawler自动遍历工具的前世今生和一些feature.
当时因为在完善一些细节, 所以没有放出来试用版本. 今天算是给大家补上吧.

ChangeLog

增加了一个实验性的自动化测试框架. 放出来也是想让大家当小白鼠
增加了自动化测试的支持. 可以在自动化中进入自动遍历.
增加了Log插件. 支持读取Android的Logcat和iOS的syslog数据.
增加了TagLimit插件. 用于自动识别类似的控件减少遍历, 加快速度.
使用了标准的log4j风格的Log输出
增加了启动的默认划屏操作

下载地址

我把工具打包成了jar包.

百度网盘: <http://pan.baidu.com/s/1pLcjBkB>

新浪微盘: <http://vdisk.weibo.com/s/a1i55uMrHR9DZ>

开源时间还不确定. 心急的同学可以自行反编译. :)



43 个赞

收藏



共收到 **139** 条回复

参与回复

置顶 屏蔽 关闭



代码与文档内部公开在社区

Introduction

启动参数介绍

Android遍历

iOS遍历

遍历控制

动作触发器



自动化测试结合

兼容性测试

XPath表达式学习

▼ 插件

插件开发

代理插件

Log插件

JDIAAndroid自动调...

代理插件

自动获取app上每次点击对应的网络请求. 支持http和https

安装

目前是默认自带.

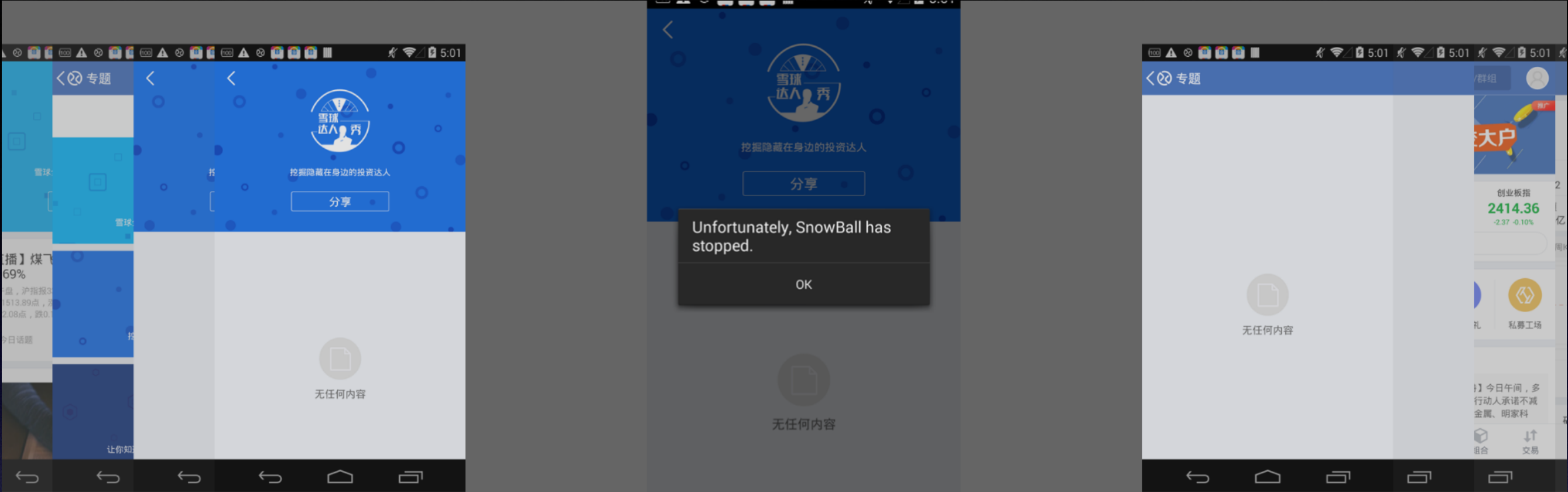
启用

在配置文件中加入插件



```
"pluginList" : [  
    "com.xueqiu.qa.appcrawler.plugin.TagLimitPlugin",  
    "com.xueqiu.qa.appcrawler.plugin.ProxyPlugin"  
],
```

代理插件默认开启7771端口.

配置你的Android或者iOS的设备的代理. 指向你当前运行appcrawl



74_MainActivity_-qj_专题-qi_-android.widget.TextView--无任何内容.jpg

名称	修改日期	大小	种类
 73_MainActivity_-qj_专题-qi_-android.widget.TextView--挖掘隐藏在身边的投资达人.jpg	今天 下午1:01	121 KB	JPEG 图像
 74_MainActivity_-qj_专题-qi_-android.widget.TextView--无任何内容.jpg	今天 下午1:01	135 KB	JPEG 图像
 75_MainActivity_-qj_专题-android.view.View-action_bar_overlay_layout-.jpg	今天 下午1:01	57 KB	JPEG 图像
 76_MainActivity_-qj_专题-android.view.View-action_bar-.jpg	今天 下午1:01	57 KB	JPEG 图像
 77_MainActivity_-qj_专题-android.widget.ImageView-home-.jpg	今天 下午1:01	461 KB	JPEG 图像
 78_MainActivity_-qj_专题--_-android.widget.TextView--上证指数.jpg	今天 下午1:01	200 KB	JPEG 图像
 79_qj_专题--_StockDetailActivity_-Back-Back-Back.jpg	今天 下午1:01	597 KB	JPEG 图像
 80_MainActivity_-android.widget.TextView--组合风云榜.jpg	今天 下午1:01	250 KB	JPEG 图像
 81_MainActivity_-jd_组合风云榜-android.widget.TextView--总排行榜.jpg	今天 下午1:01	317 KB	JPEG 图像
 82_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--总排行榜.jpg	今天 下午1:02	322 KB	JPEG 图像
 83_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--组合综合实力秀.jpg	今天 下午1:02	322 KB	JPEG 图像
 84_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--最近6个月.jpg	今天 下午1:02	322 KB	JPEG 图像
 85_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--最近3个月.jpg	今天 下午1:02	336 KB	JPEG 图像
 86_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--涨停.jpg	今天 下午1:02	197 KB	JPEG 图像

app功能遍历思维导图



技术点

- 跨平台 -> 基于Appium
- 简化 -> 基于Scala :)
- 自动执行 -> 爬虫思路
- 自定义 -> 插件化+XPath+控件定位的URI

与传统WebDriver的不同点

- findElementByXXX 根据标记直接定位
- see 先dump所有app的控件再通过xpath 正则等各种方式定位. 所以更灵活
- 每次操作都要重新刷新页面 拿到数据可以做更多的分析. 比如智能判断UI兼容性

自动化支持[实验性支持]

```
override def beforeEach(): Unit = {  
    config("appPackage", "com.xueqiu.android")  
    config("appActivity", ".view.WelcomeActivityAlias")  
    appium()  
}
```

```
test("自选") {  
    swipe("down")  
    see("自选").tap  
    see("雪球100").tap  
    swipe("down")  
    see("stock_current_price")("text").toDouble should be > 1000.0  
}
```


see

唯一的元素定位api.

see是引用了<阿凡达>电影里面一句台词"I See You".

它的作用是当你看到一个控件,你应该可以根据看见的东西就可以定位它,并获取到这个控件的属性,无须借助其他工具或者使用findElementByXXX之类的函数.

比如有个Button,名字是"登录",它的id是account,定位它可以通过如下多种方式的任何一种

- see("登录")
- see("登")
- see("录")
- see("account")
- see("acc")
- see("//UIButton[@id="account"]")
- see("screen_name")("text")
- see("screen_name").nodes.head("text")
- see("action_bar_title")("text") 文本
- see("action_bar_title")("tag") 类型
- see("action_bar_title")("selected") 是否选中

如果当前界面中有了有歧义的空间,比如其他一个名字为"登录"的输入框.那么上述定位方法中定位中两个控件的定位方法会失败,你需要自己调整即可.

这就是关于元素定位你只需要用see这个方法即可.

动作

目前只封装了3个动作. tap send swipe.

```
see("输入手机号").send("13067754297")  
see("password").send("xueqiu4297")  
see("button_next").tap()
```

支持链式调用. 当然不推荐日常使用

```
//对三次连续出现的tip控件点击三次.  
see("tip").tap().tap().tap()  
see("输入手机号").send("13067754297").see("password").send("xueqiu4297")
```


XPath的充分利用

- 把当前app的界面dump为xml结构
- 利用xpath定位界面和控件
- 自动点击+自动输入[危险]
- 利用规则来引导输入或者遍历顺序

目前的使用方式和收益

- 手工测试+自动遍历
- Android 一小时1000张
- iOS一小时500张
- 结合LeakCanary发现大部分的内存泄漏
- 发现一些低级的崩溃

为什么开源

- 雪球是一家小而美的公司
- 希望用最小的投入换取最大的收益
- 有限的精力只能放在关注业务上 28原则
- 感谢开源 回馈开源
- 已经提前开源在社区内部并开放了1.2.1测试工具下载
- 已经获得领导批准 将在下周一正式开源到GitHub

提醒

- 自动遍历获得了从安全大牛到小白各阶段工程师的欢迎.
- 不要过于依赖自动遍历 仍要专注于业务测试
- 没有银弹