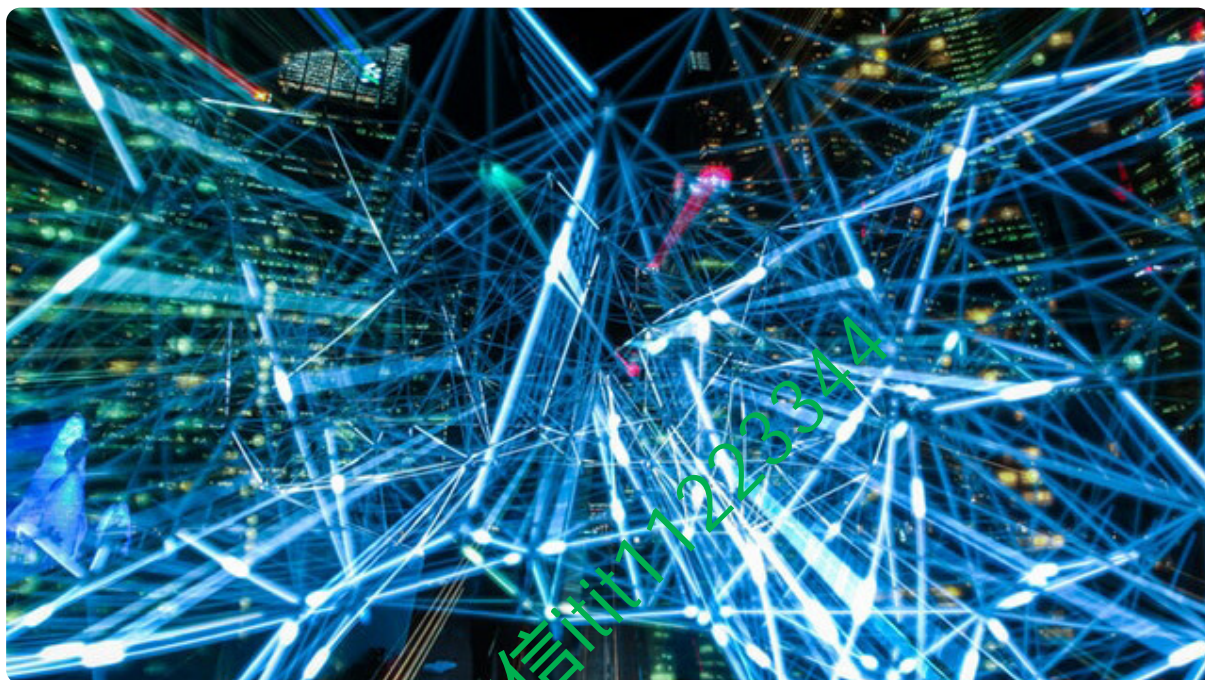


22 探索移动自动化的世界

更新时间: 2019-10-09 09:04:25



“理想必须要人们去实现它，它不但需要决心和勇敢而且需要知识。——吴玉章”

每次谈起 APP 测试，很多小伙伴就不由自主的联想到 APP 自动化测试，似乎只有自动化测试，才是一个领域里最“至高无上”的技术巅峰。然而，我在开篇就要给大家一记重锤：对于 APP 测试来说，掌握上节内容中那些弱网、中断等测试手段和方法是相比自动化更重要的技能。

这也不是说我们不需要考虑自动化测试了，在 WEB、接口、APP 中我自己是最晚才接触 APP 自动化测试的，但是发现其实从自动化的角度上别无二致。自动化测试的原理和设计思路其实在前边 WEB 和接口已经聊过不少了，所以相信大家也理解了自动化的思想，唯一不明确的就是工具的选择。

“风落”的 APP 自动化之旅

接下来，又到了我“现身说法”的时候了。我刚接触 APP 自动化的时候，由于还在一家小公司，以探索尝试为主，所以先拿来“祭旗”的就是 Android APP（PS：穷苦的少年风落一直在用的就是 Android 手机，彼时也还没有获得持有 IPHONE 的老婆的青睐），本着以身正法的信念，加入了刚刚出现的 APP 自动化大军。

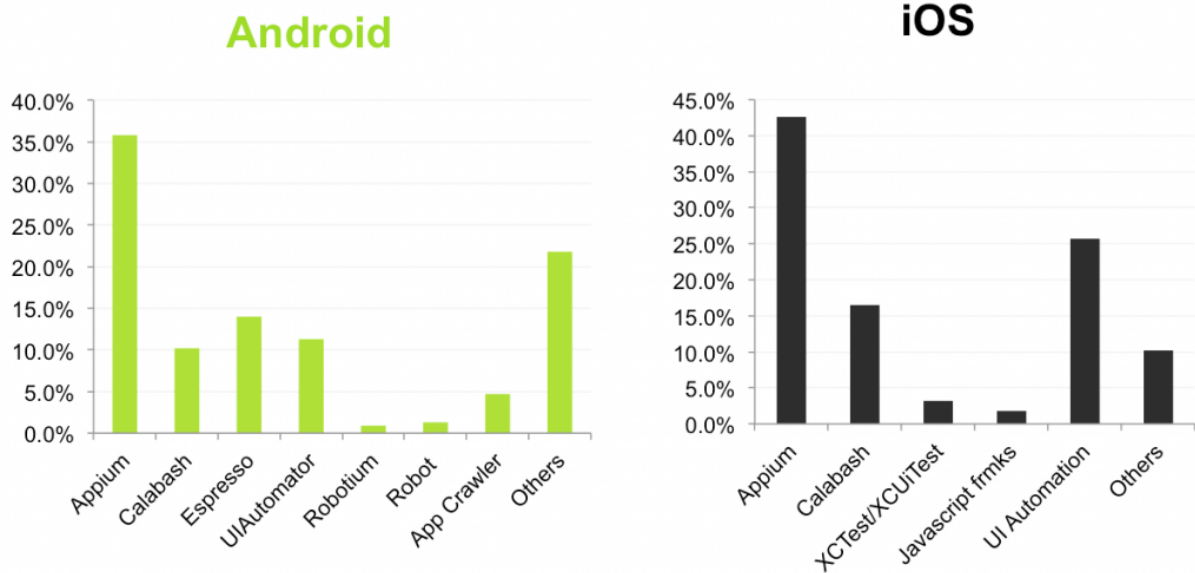
最早用的框架自然不用说，自然是 Google 最早提供的 Instrumentation 自动化测试工具类，这算是在当时非常出色的一项技术，当你看到通过代码能够操作各种按键、屏幕操作、滚动等等事件的时候，成就感还是略微超过 WEB 自动化的。但是虽然我自己用起来感觉还可以，但是推广起来却很难，因为进行 Instrumentation 脚本编写还是要对 Android 相关知识有不俗的了解，加上要配置一个 Android 的 xml 文件，实在是有些复杂。

于是需要探索更新的框架，浮出水面的就是 Robotium 和 UIAutomator 了。Robotium 虽然是基于 Instrumentation 开发的，但是对常用的操作进行了二次封装，所以脚本编写上也容易了很多，同时速度更快。但是同样的，仍然摆脱不了 Instrumentation 的一些劣势。相比之下 Google 提供的 UIAutomator 框架打破了之前大家自动化一定要 Instrumentation 的束缚，重新定义了 APP 自动化测试。但是它也有自身的缺点，就是库只能支持 JAVA 使用，且对 Android 版本很有要求，老版本的 Android 系统无法支持。当然，在当时，UIAutomator 已经成为了新型 APP 自动化测试工具的代表。

但是很快的，一代 APP 自动化测试框架的主角诞生了：Appium。我也是在简单研究测试后，很快投入了 Appium 的怀抱。它的优势非常明显：

- 开源
- 跨平台，支持 Android、iOS、Firefox OS
- 不需要源代码，不需要重新编译 App，可以基于现有的执行
- 扩展了 WebDriver，对于测试人员来说几乎无需切换
- 支持多种主流语言编写

从原来上来说，在 Android 端 4.2 以上是用 UiAutomator 作为底层，Android 2.3 ~ 4.1 用的是 Instrumentation，而在 ISO 则是封装 UIAutomation 作为底层，这样拥有了所有框架的优点，很快的成为了 APP 自动化测试的不二之选。



从上边统计图片中也能看到，Appium 在 APP 自动化测试领域无疑是最佳也是最普遍的选择。所以，后面我将框架再进行 iOS 上的扩展仍然没有考虑过其他底层，复用着 APPIUM，直到今天。

所以对于后来也有很高评价和口碑的工具，诸如 Android 端的 Espresso、Calabash 以及 iOS 端的 XCTest、Frank 都没有再继续深入使用。尤其是 Calabash 和 Frank，行为驱动的模式也从某种程度上改变了自动化的方式，所以大家有兴趣的可以研究一下。

就像我们的 Web 框架、接口框架一样，APP 自动化也有一些开源的框架和集成平台，例如 AppTester、TestBird 等等，目前市面上主流的框架平台也都是基于 Appium 进行的二次封装。

当然，有很多像我一样喜欢重复造轮子的童鞋呢，也可以基于 Appium 来开发搭建属于自己的框架，完全可以跟自己的 Web 或接口框架融为一体。

更新的移动自动化技术

随着互联网业务的发展，很多项目里 H5 和小程序的占比越来越多，同时由于互联网 H5 和小程序相对于 WEB 和 APP 来说，迭代速度更快，所以自动化测试也更加迫在眉睫。

尝试使用了一下 Appium，对于 H5 还是可以部分支持的，只是在页面切换稍微有一点问题，对于微信小程序就比较困难了。所以恰到好处的，腾讯开源了 FAutoTest (FAT) 框架，重点针对微信 H5 和小程序，使用起来还是相对快捷的，但是有它的限制所在，只对微信相关的兼容比较好，对于支付宝小程序、H5 如果想要支持，还是需要修改一部分源代码的（这是当然的，毕竟

自动化前瞻

到这里，我们对于自动化测试的内容就告一段落了，尽管自动化已经发展了很多年，但是我认为还处于未完成成熟的阶段。目前关注的主要在于“模拟”和“自动”，我个人觉得未来的方向有可能如下几个：

1. **行为驱动优化**：目前的行为驱动还是略显不够合理，更多的框架是硬向行为驱动上靠近，也许未来会做更多的行为驱动上的优化和改进，把行为驱动进一步发扬光大。当然，现在的 cucumber 如果使用习惯的话，还是比较出色的。
2. **流量回放自动化**：刚刚说了，现在自动化关注的是模拟。但是模拟的足够么？我觉得还不够，因为模拟的更多还是基于我们自己编写的用例，那是不是有办法可以把客户的行为动作模拟下来呢？这就是流量回放的意义，可以把生产环境的流量完全复制到自动化测试环境。

现在的阿里已经有了这样的数据平台，叫 DOOM，但是还没有能够完整推广和开源开来，所以现在的流量回放更多是从 TCP 请求上下手，这样不利于我们针对生产数据的清洗，这也有可能是未来自动化测试的发展方式之一。

3. **AI 人工智能测试**：其实现在已经有很多团队开始基于 AI、基于神经网络和图像识别开展完全无需编程的自动化测试了，但是效果还比较有局限性。但是阿尔法狗已经出现了，下一个阿尔法 Tester 会很远么？



21 求同存异：论移动测试与传统测试

23 换个角度看世界-从为什么要做性能测试谈起



精选留言 1

欢迎在这里发表留言，作者筛选后可公开显示

土豆稀饭

顶一个，更新速度好快呀，赞赞赞

千学不如一看，千看不如一练

一手微信itit1223344