

深入理解自动化测试工具 Appium

任 涛

(中国人民大学, 北京 100872)

摘要:在信息时代飞速发展的今天,移动软件已被应用到世界的各个领域,改变着人们的生活方式。人们的工作、学习、生活、娱乐等都与移动软件紧密相连,如何保证软件质量、如何减少问题缺陷给企业和用户带来的损失和隐患也就成为了软件开发过程中必须重视的问题。因此,软件测试的整个生命周期就显得尤为重要。笔者从软件测试中的自动化测试的角度出发,介绍 Appium 这款自动化测试工具。

关键词: Appium; 软件测试; 移动 APP

中图分类号: TP311.53 **文献标识码:** A **文章编号:** 1003-9767(2016)21-139-02

软件测试在整个软件生命周期中占有重要的地位,直接关系到软件的质量以及开发的总成本,随着软件开发技术的快速发展,软件的架构设计和开发技术得到了极大的提高。然而,测试的工作量却并未减少,依靠手工在软件开发后期不断重复地做回归测试,不仅费时耗力而且易出错。软件自动化测试技术的出现和发展,使得软件测试进入了一个崭新的时期,而自动化测试框架的引入及测试工具的开发则是提高自动化测试水平和效率的重要手段,将繁琐的测试自动化,提高了测试的效率,节约了人力成本,同时也弥补了手工测试的各种缺陷与不足。Appium 就是一款专门为移动软件设计的自动化测试工具,下文开始将从基础概念开始介绍。

1 Appium 简介

Appium 是由 Sauce Labs 提供支持的一款开源自动化测试框架,它是针对原生、混血和移动 Web 应用的一款高效的测试工具。它使用 Web Driver 协议来驱动 IOS 和 Android 应用。

Appium 的设计理念是测试原生应用,不要求用户引入额外的 SDK 或者重新编译应用,可直接使用打包好的 App 进行测试。另外,Appium 应能与测试人员喜欢的测试实践、测试框架、测试工具一起使用。

Appium 旨在使用户可以通过任何语言以及任何测试框架去自动化测试任何移动应用,并且可以通过测试代码访问后端的 API 和 DB。

2 Appium 的基本概念

Appium 是 IOS 和 Android 平台上针对原生、移动 Web 和混血应用的开源自动化测试工具。原生应用是使用 IOS 和 Android SDK 构建的应用。移动 Web 应用是通过移动浏览器(Appium 在 IOS 平台上支持 Safari,Android 平台上支持 Chrome 或者内置的 Browser 应用)访问的 Web 应用。混血应

用有一个壳包裹 WebView 一个原生控件用于与 Web 内容进行交互。通过一些工程(例如 PhoneGap),可以很方便地使用 Web 技术构建应用,并将其绑定到原生的壳中,从而创建一个混血应用。

重要的是,Appium 是跨平台的,可以在多个平台上(例如 IOS 和 Android)做测试,使用相同的 API。这样可以做到 IOS 和 Android 测试集的代码重用。

3 Appium 的设计理念

Appium 是根据以下四条原则描述的设计理念,来满足移动自动化的需求。

第一,测试人员不用为了自动化测试而去重新编译应用或者以任何方式修改;第二,编写自动化测试时,不会将测试人员限制在某种特定语言或者框架上;第三,移动自动化测试框架不应另起炉灶,搞一套自己独有的自动化 API;第四,移动自动化测试应是开源的。

关于 Appium 的设计,Appium 通过封装厂商提供的框架到统一的 API(即 WebDriver API)中,来满足第二个理论。WebDriver(又名 Selenium WebDriver)规范了一个客户端/服务器协议(被称为 JSON Wire Protocol)。基于该客户端/服务器架构,用任何语言编写的客户端都可以发送相应的 HTTP 请求到服务器。Appium 提供了用各种流行语言编写的客户端,这意味着,测试人员可以使用任何想要的测试运行器和测试框架。Appium 的客户端是简单的 HTTP 客户端,可以以任何方式集成在代码中。换句话说,Appium & WebDriver 客户端不是技术上的测试框架,它们是自动化测试库。测试人员可以以任何喜欢的方式来管理测试环境。

Appium 用相同的方式来满足第三个理念。WebDriver 意境成为 Web 浏览器自动化测试事实上的标准,并且是 W3C 工作草案。Appium 针对移动自动化测试,提供了额外的 API 方式用于扩展该协议。

作者简介:任涛(1988-),男,天津人,本科,工程师。研究方向:计算机软件技术。

4 Appium 的重要组成模块

4.1 客户端 / 服务器架构

Appium 的核心是一个暴露了 RestAPI 的 Web 服务。它从客户端接受连接、监听命令，并在一个移动设备上执行这些命令，接着响应一个 HTTP Response 表示命令执行的结果。事实上，Appium 的客户端 / 服务器架构开创了很多可能性：测试人员可以使用任何语言（包括 HTTP 客户端 API）来编写测试代码，但是如果使用 Appium 客户端会更容易。测试人员可以将服务器放在与测试运行机不同的机器上。测试人员可以写测试代码并依赖于云服务去接收和解释命令。

4.2 Session

自动化测试总是基于绘画内容来实现的。客户端与服务端初始化会话的方式，具体到不同的库有区别，但它们最终都会发送一个 Post/Session 请求到服务器，携带一个名为 Desired Capabilities 的 JSON 对象。在这一时刻，服务器将会启动自动化会话并响应一个 Session ID 用于发送后续的命令。

4.3 Desired Capabilities

Desired Capabilities 是一组发送到 Appium 服务器的键和值（即 Map 或者 Hash），告诉服务器测试人员期望建立的自动化会话类型。这里有各种类型的属性，可以用来改变服务器在测试期间的行为。例如，可以设置 Platform Name 属性的值为 IOS，以便告诉 Appium 我们期望 IOS 会话而非 Android 会话。或者可以设置 Safari Allow Popups 属性为 true，目的是在 Safari 自动化测试会话中，允许使用 JavaScript 打开新窗口。

4.4 Appium Server

Appium 是用 node.js 写的服务器。它可以通过源代码构件和安装或者直接通过 NPM 安装。

4.5 Appium Clients

Appium 提供了 Java、Ruby、Python、PHP、JavaScript 和 C# 语言的客户端库，都是基于 WebDriver 协议扩展的。当使用 Appium 时，测试人员可以使用它们来代替标准的 WebDriver 客户端。

4.6 Appium.app 和 Appium.exe

Appium 提供了 Appium server 的 GUI 封装下载。它们包含了运行 Appium Server 需要的所有东西，因此测试人员无须担心 Node。它们还携带了 Inspector，用于检查应用程序的层次结构，在编写测试代码时会很有用。

5 Appium 的原理框架

在 Android 客户端，Appium 基于 WebDriver 协议，并利用 bootstrap.jar，最后通过调用 UIAutomation 的命令，实

现 App UI 自动化测试的 Java 库。另外，由于 UIAutomator 对 WebView 的支持有限，Appium 引入了 ChromeDriver 来实现基于 WebView 的自动化。

在 IOS 客户端，Appium 同样使用 WebDriver 协议。与 Android 端测试框架不同的是，Appium IOS 封装来苹果公司的 Instruments 框架，主要用 Instrument 里的 UIAutomation（Apple 自动化测试框架），然后在设备中注入 bootstrap.js 进行监听。

6 选择 Appium 的原因

首先，测试人员不需要重新编译应用（App）或者以任何方式修改它，因为 Appium 在所有平台上使用标准的自动化 API。

然后，测试人员可以使用自己喜欢的任何开发工具，可以使用任何兼容 Web Driver 的语言，例如 Java、JavaScript、Node.js、Php、Python 等，并且基于 Selenium WebDriver API 和特定语言的客户端库来实现自动化测试。

最后，测试人员可以使用任何测试框架来驱动 Appium。

基于 Web Driver 协议，意味着将赌注押在一个统一、自由和开放的测试协议上，该协议已成为事实上的国际标准。

如果不能通过 Appium 直接使用苹果的 UIAutomation 库，测试人员只能使用 JavaScript 语言，并且只能通过 Instruments 应用进行测试。同样，对于 Google 的 UIAutomator，测试人员只能使用 Java 来编写测试代码。Appium 开辟了真正跨平台的本地移动自动化的可能性。

7 结 语

上文深入介绍了 Appium 这款自动化测试工具，其实它只是自动化测试的一个代表。回顾软件自动化测试的发展历程可以发现，自动化测试的每个阶段都伴随着软件测试人员对自动化测试技术的探索作出的努力，在自动化测试过程中选择合适的框架和测试工具不仅对自动化测试的结构起到了一定的帮助，还从方方面面提高了测试的效率。如何更好地进行软件测试工作，一款好的工具仅仅是个开始，我们需要不断完善理念，在实践中发现问题、分析问题，找到解决问题的更好办法。但是不可否认的是自动化测试技术将会不断进步，逐步代替手动测试。而如今日益白热化的软件市场对产品的质量提出了越来越严苛的要求，作为软件的质量把关者，监控者只有紧紧跟随自动化测试脚步，设计更为详细的测试用例，充分了解产品需求才能成为一个成功的测试工程师。

参考文献

- [1] 温素剑. 低成本实现 Android/iOS 自动化测试——基于 Appium 和 Test Perfect[M]. 北京: 电子工业出版社, 2016.
- [2] 邓佰臣, 张艳梅, 王俊华. 从菜鸟到测试架构师——一个测试工程师的成长日记[M]. 北京: 电子工业出版社, 2013.