随着互联网的迅速发展，为了满足用户的需求，产品迭代速度也越来越快，持续集成（CI）和持续交付（CD）都旨在缩短开发周期、提高软件交付效率以及实现全流程的自动化测试。

对于测试人员来说，使用自动化的手段去完成一些重复性高的回归测试工作、或者性能测试工作，用更多的精力去探索发现更复杂的业务逻辑的问题显得尤为重要。

对于客户端产品 UI 界面的功能测试，Appium 是一个非常好的选择，它支持 Android、iOS 系统的原生应用，网页应用以及混合应用，同时也支持多语言，比如 Java、Python、Ruby、JS 等。可以使用 Appium 完成回归测试，冒烟测试等测试阶段的工作。

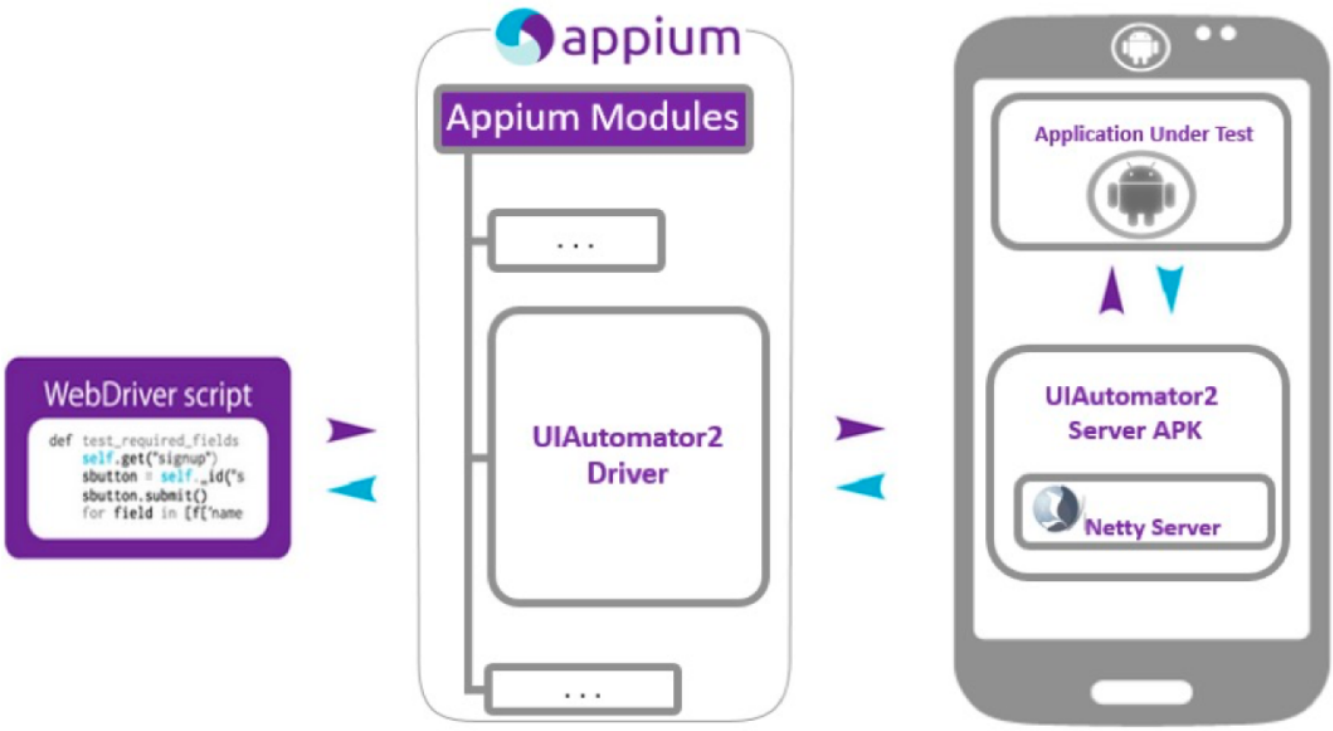
Appium架构

Appium 设计哲学

- 不需要为了自动化而重新编译或修改被测应用
- 不应该让移动端自动化测试限定在某种语言或者某个具体的框架
- 不要为了移动端的自动化测试而重新造轮子
- 移动端自动化测试应该是开源的

Appium 架构

Appium 架构图如下：



Appium 的核心是一个 Web 服务器，提供了一套 REST 的接口，接收到客户端的连接，监听到命令，在移动设备上执行这些命令，将执行结果放在 HTTP 响应中返还给客户端。

事实上，这种客户端/服务端的架构给予了许多的可能性。可以使用任何实现了该客户端的语言来写测试代码，可以把服务端放在不同的机器上，可以只写测试代码，然后使用服务来执行命令。

对于 Android、iOS 底层使用了不同的工作引擎驱动实现自动化测试。Appium 引擎列表：

Platform	Driver	Platform Versions	Appium Version	Driver Version
iOS	XCUITest	9.3+	1.6.0+	All
	UIAutomation	8.0 to 9.3	All	All
Android	Espresso	?+	1.9.0+	All
	UiAutomator2	?+	1.6.0+	All
	UiAutomator	4.3+	All	All
Mac	Mac	?+	1.6.4+	All
Windows	Windows	10+	1.6.0+	All

Appium 支持的语言

Appium 支持如下语言编写测试用例：

Language	Support	Documentation
Java	All	seleniumhq.github.io
Python	All	selenium-python.readthedocs.io
Javascript (WebdriverIO)	All	
Javascript (WD)	All	github.com
Ruby	All	www.rubydoc.info
PHP	All	github.com
C#	All	github.com

Appium 环境安装

Appium Windows 版本只支持 Android 系统，Appium Mac 版同时支持 Android 系统 和 iOS 系统。这里只介绍 MacOS 系统的安装。

Appium 环境依赖

软件列表：

```
1. Java 1.82. Android SDK3. Appium Desktop
```

其中 Java 推荐使用 1.8 版本。Android SDK 是 Android 系统的开发工具包，里面有很多自动化测试常用的工具。Appium Desktop 提供了服务与录制功能。

Appium 客户端安装（Python版本）

如果想要在代码中能够相关包，需要安装第三方库：

```
pip install Appium-Python-Client
```

Appium 客户端安装（Java版本）

当使用 Maven 或 Gradle 等构建工具时，会自动加载依赖项。

```
<properties>        ...        <!-- 尽可能使用最新版本 -->
<appium.version>7.3.0</appium.version>        ...    </properties>    <dependencies>
    ...        <dependency>            <groupId>io.appium</groupId>
<artifactId>java-client</artifactId>            <version>${appium.version}</version>
    </dependency>        ...    </dependencies>
```

Appuim的原理

可概述为以下的内容：

第一次连接创建一个session对话

1、客户端运行脚本，向Appium Server端post一条HTTP请求，内容为json数据

2、Appium Server端接收到请求后，解析出JSON数据并发送到手机端

3、手机端接收到对应的请求后，

4、通过BootStrap.jar翻译成UIAutomator能执行的命令，然后通过UIAutomator处理并操作APP完成测试。