



霍格沃兹测试学院

1. Appium与WebDriver

- [什么是Appium](#)
- [什么是WebDriver](#)
- [两者有什么联系](#)
- [更进一步](#)

什么是Appium

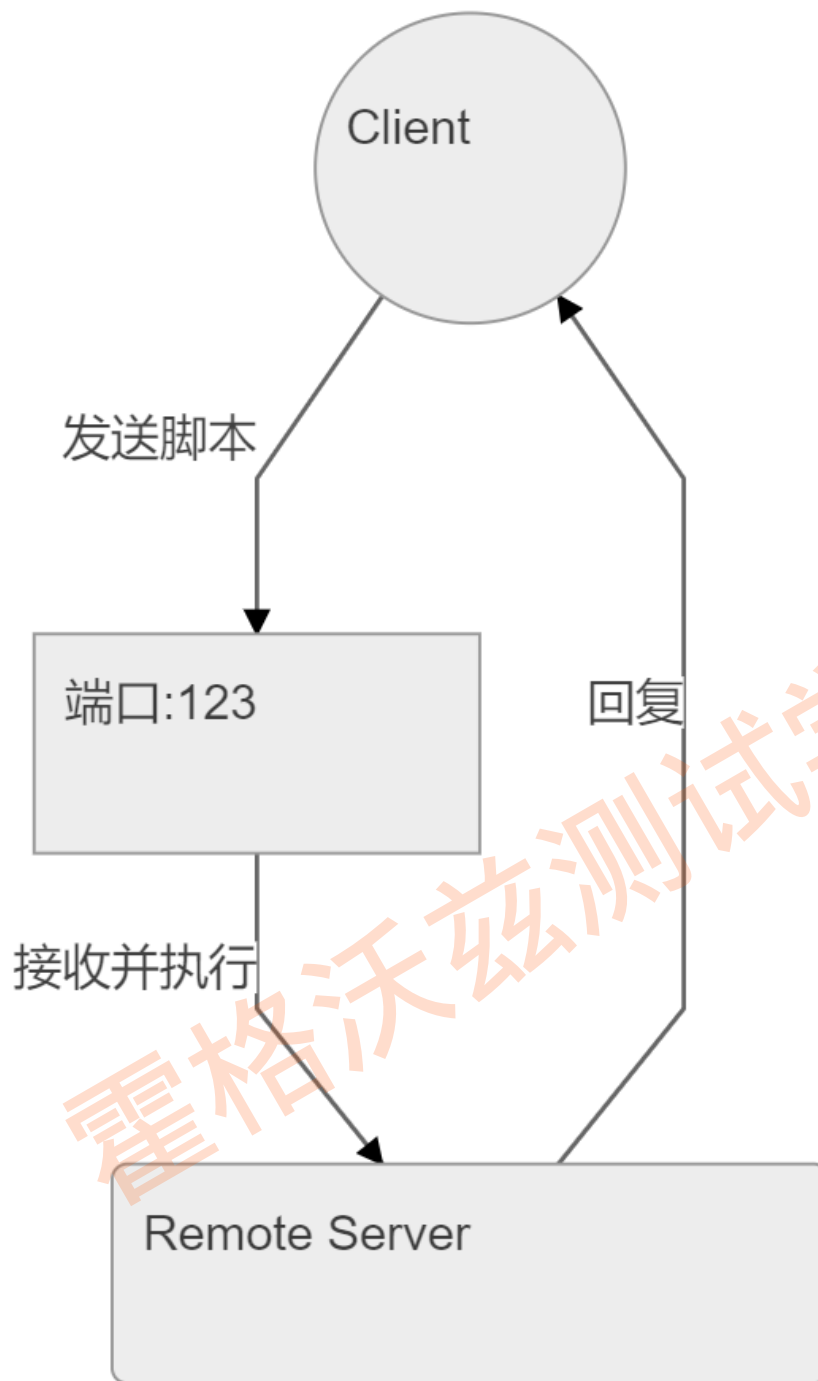
这是一个开源的**自动测试**框架，可以用它来**操控app**，测试app**功能**的完美性。当然，它的作用远不止这些。如果你高兴，可以用它来收能量，签到，甚至是游戏自动化。正式学习之前，我希望你能明白下面Appium的工作原理。

什么是WebDriver

WebDriver的**设计模式**是这样的： – Server端：即Remote Server（远程服务器），可以是任意的浏览器，当脚本启动浏览器时，该浏览器就是Remote Server，它的职责是等待Client发送请求并做出响应。 – Client端：简单来说就是我们的测试代码，测试代码中的一些行为是以HTTP请求的方式发送给被测试浏览器——Remote Server，Remote Server接受请求，执行相应操作，并在Response中返回执行状态、返回值等信息。

WebDriver**工作流程**是这样的： 1. WebDriver启动目标**浏览器**，并绑定至指定**端口xxxx**，启动的浏览器实例将作为WebDriver的Remote Server。 2. Client端通过CommandExcuter发送HttpRequest

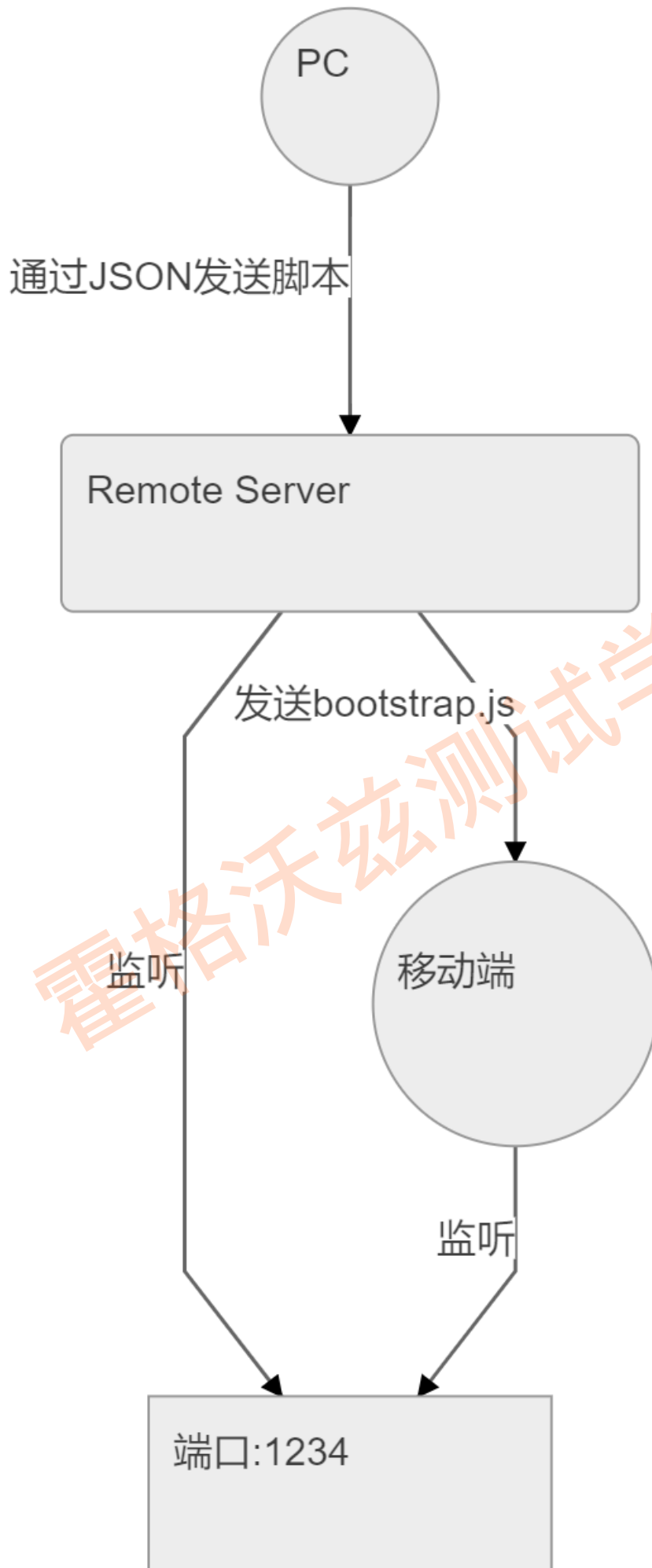
给Remote Server的侦听端口xxxx。 3. Remote Server需要依赖原生的浏览器组件（比如：chromedriver.exe）来转化浏览器的native调用。



两者有什么联系

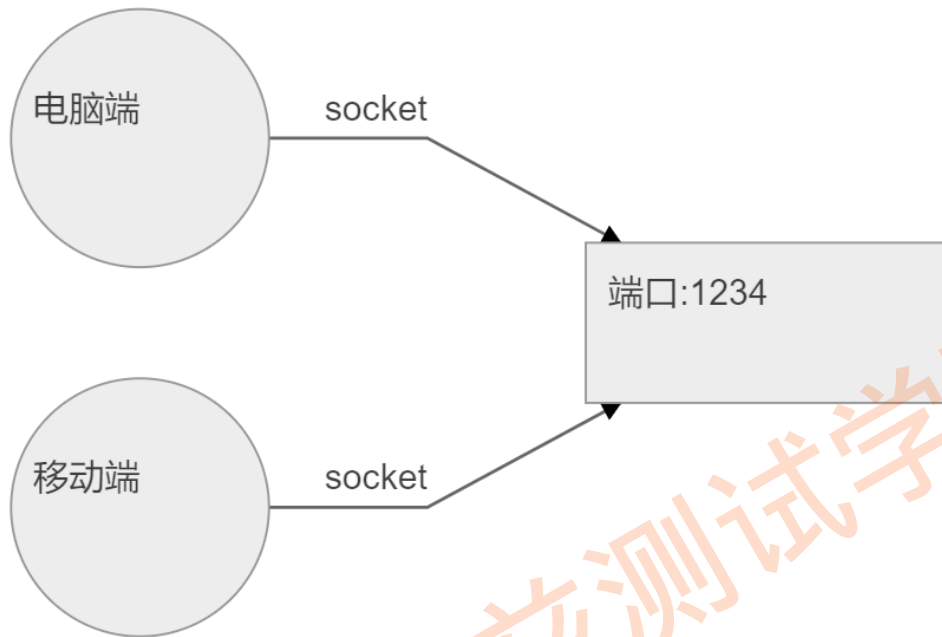
Appium基于**WebDriver模式**，对比WebDriver，Appium的执行过程如下：

1. **PC端**的测试人员将测试脚本通过**json**格式传递给**Remote Server**
2. Remoserver启动了一个**监听端口:1234**。同时向手机端adb push 一个**bootstrap.jar/bootstrap.js**的脚本，手机端通过该脚本同时监听端口1234。



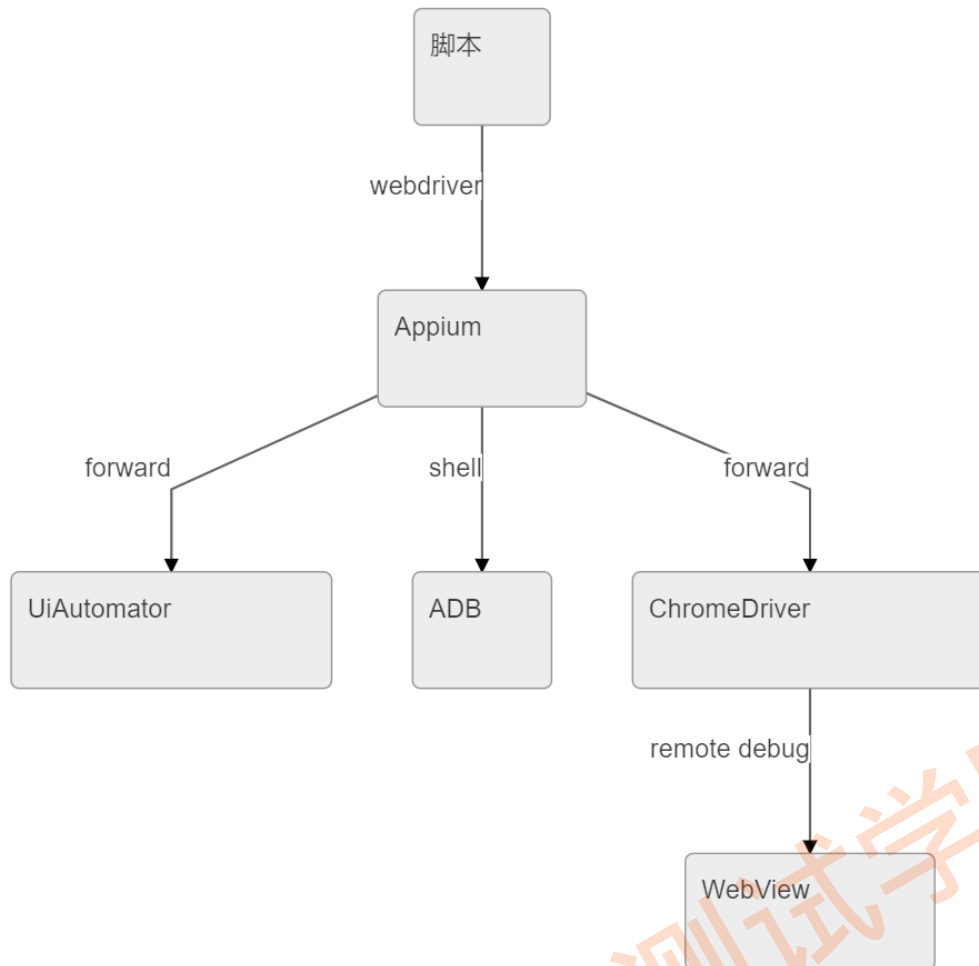
3. PC和移动端就通过这个端口实现了**通信和交互**，基于socket通信（一个封装了TCP/IP协议的接口）。
4. 手机端通过该端口传输的**命令**执行APP。
5. 执行完操作后通过端口**返回**给PC端，PC端根据返回结果做校验，同时也知道了操作是否执行成功。

bootstrap里面封装了安卓和苹果的自动化测试框架UIautomato

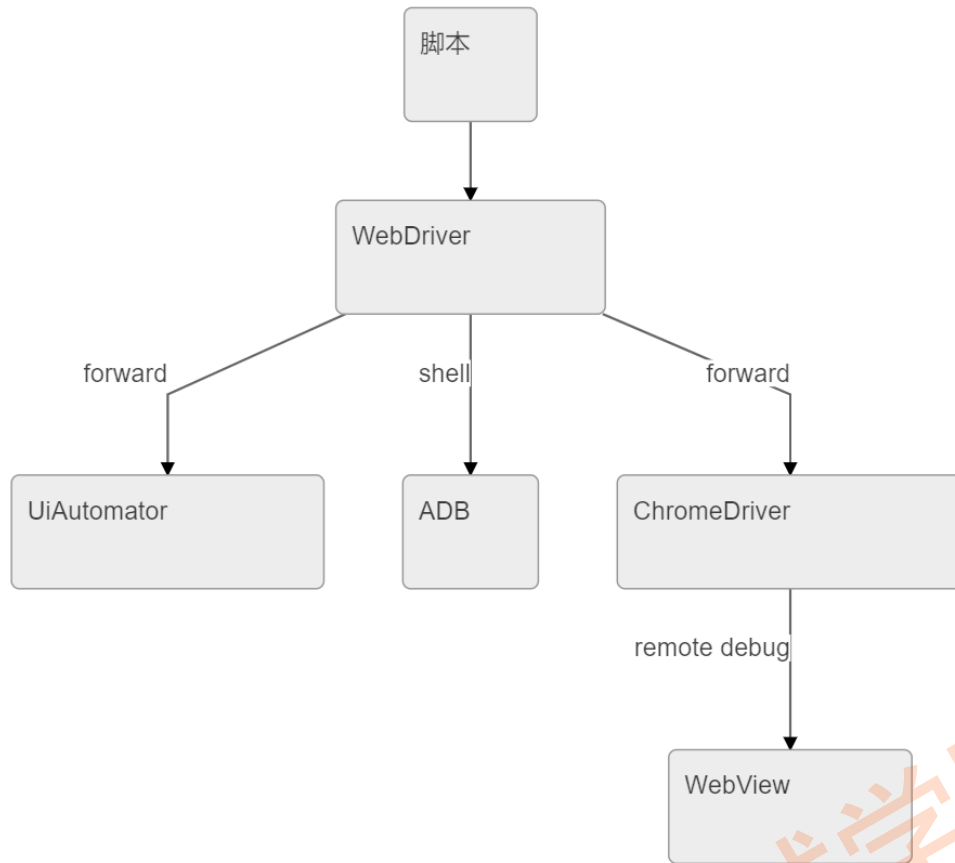


更进一步

下面的结构图清晰的表明了Appium的执行过程，其实它只是一个**中转站**：脚本传递给Appium后，Appium调用UiAutomator，ADB和ChromeDriver分别进行移动设备的UI自动化，电脑端与移动设备交互和WebView类控件的操作。



可以说，没有了Appium，上面的过程依旧成立。



WebDriver跳过Appium获取element_id的代码如下:

```
curl -H "Content-Type: application/json;charset=UTF-8" \
http://127.0.0.1:4723/wd/hub/session/$session_id/elements --data-binary '{"using":"xpath","value":"//*[@class=\\"android.widget.Toast\\""]}'
```

相关链接

霍格沃兹测试学院官网首页:<https://testing-studio.com>

Appium官网:<http://appium.io/>

Appium Client相关文档:<https://github.com/appium/appium/blob/master/docs/en/about-appium/appium-clients.md>

brew官网:<https://brew.sh/>

node官网:<https://nodejs.org/zh-cn/>

淘宝源NPM官网:<https://npm.taobao.org/>

霍格沃兹测试学院

霍格沃兹测试学院