# 通过Java+appium实现MyObservatory部分功能的UI自动化

## 步骤

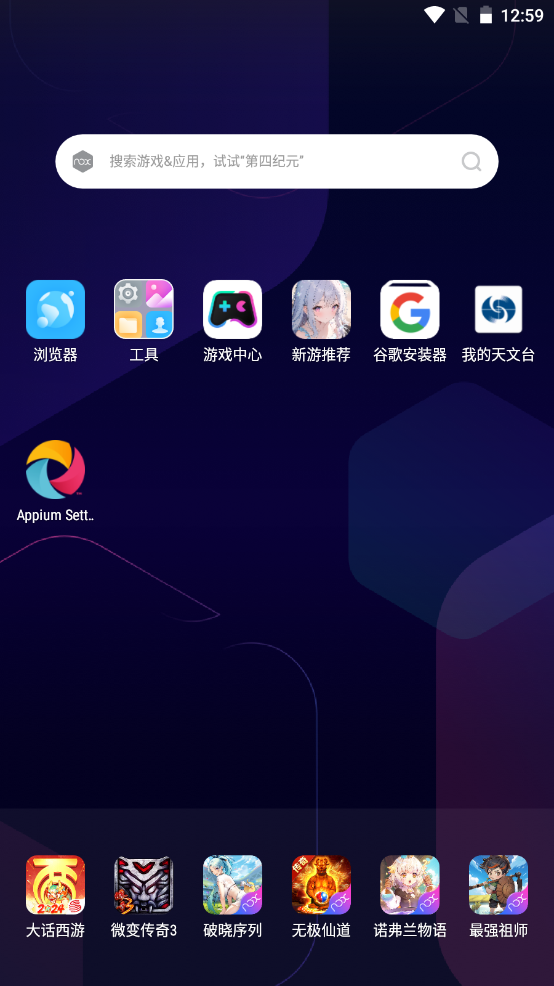
### 环境配置：

|  |  |  |
| --- | --- | --- |
| **步骤** | **作用** | **操作** |
| 1. 安装JDK | For: Appium服务器  For: 编写测试脚本 | --下载并安装JDK  --设置JAVA\_HOME环境变量 |
| 2. 安装Android SDK | Provide: AVD、  Tools(adb等) | --下载并安装Android Studio  --通过Studio安装Android SDK Platform、SDK Tools、SDK Build-Tools  --设置ANDROID\_HOME环境变量指向SDK的platform-tools和tools文件夹 |
| 3. 安装Appium Server | Appium Service | --使用Node npm 全局安装Appium，通过命令行启动Appium服务器  --或者安装Appium客户端,打开软件启动服务。 |
| 4. 安装Appium依赖 | Provide: API | --在Java项目的pom.xml文件中添加依赖  <dependency>  <groupId>io.appium</groupId>  <artifactId>java-client</artifactId>  <version> 7.3.0</version>  </dependency> |
| 5.连接Android设备 | Provide:测试运行环境 | 使用夜神模拟器创建并启动设备 |
| 6. 编写并运行测试脚本 | 执行自动化测试 | 使用Java和Appium客户端库编写测试脚本，执行测试 |

### 启用Appium环境

* 1. 连接Android设备：

打开夜神模拟器



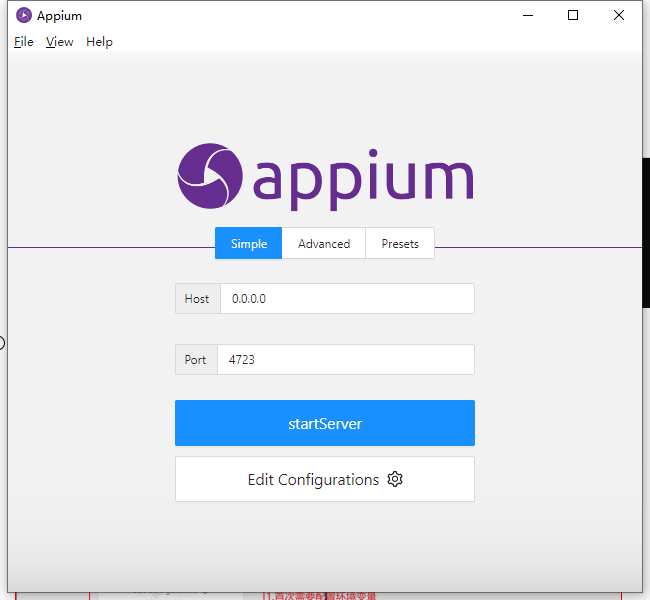
然后Windows通过 cmd命令：adb devices，识别到可用的设备ID。

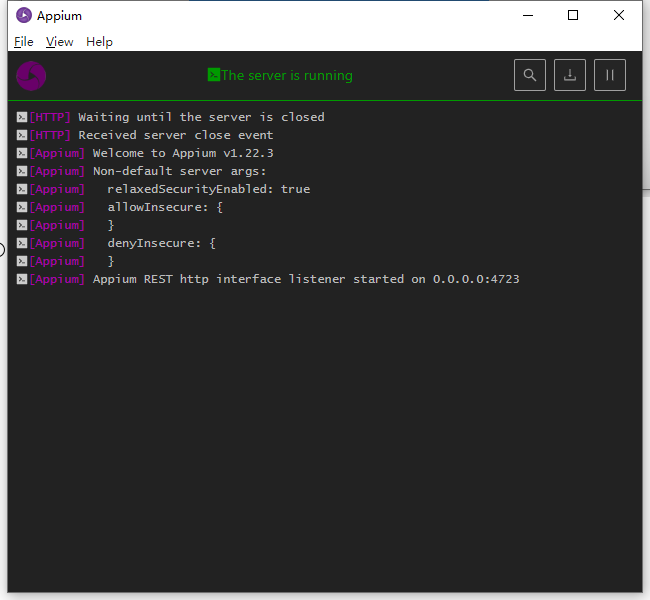
当adb命令无效时，确定安卓SDK正确下载，android-SDK\platform-tools是否添加到环境变量。



获取的设备ID用于Appium Inspector填写参数，连接该设备。

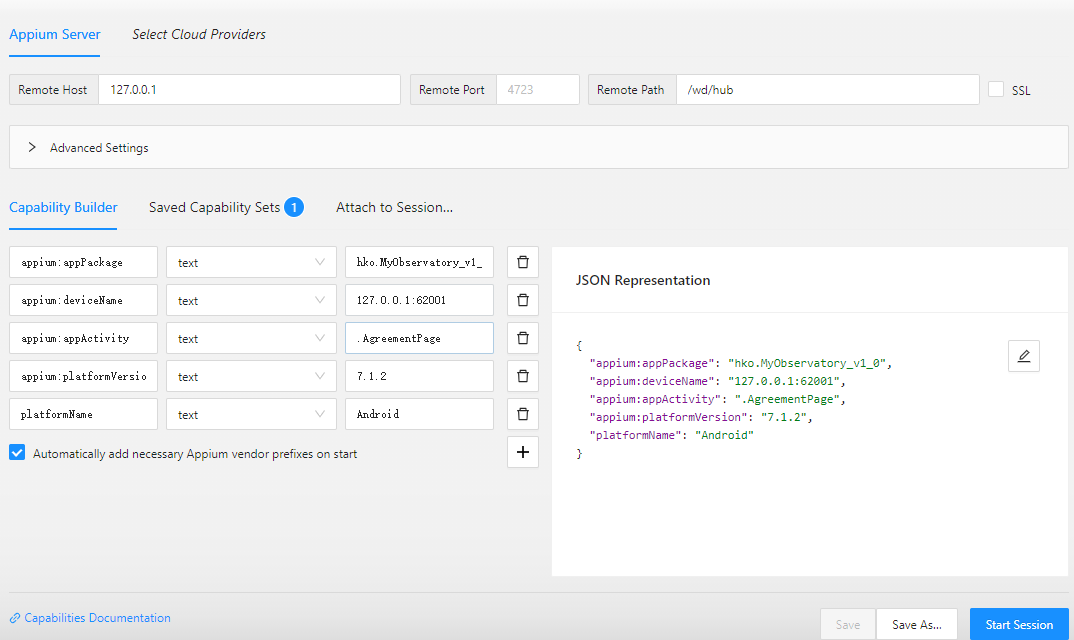
* 1. 启动Appium（服务）:





至此，Appium启动成功。

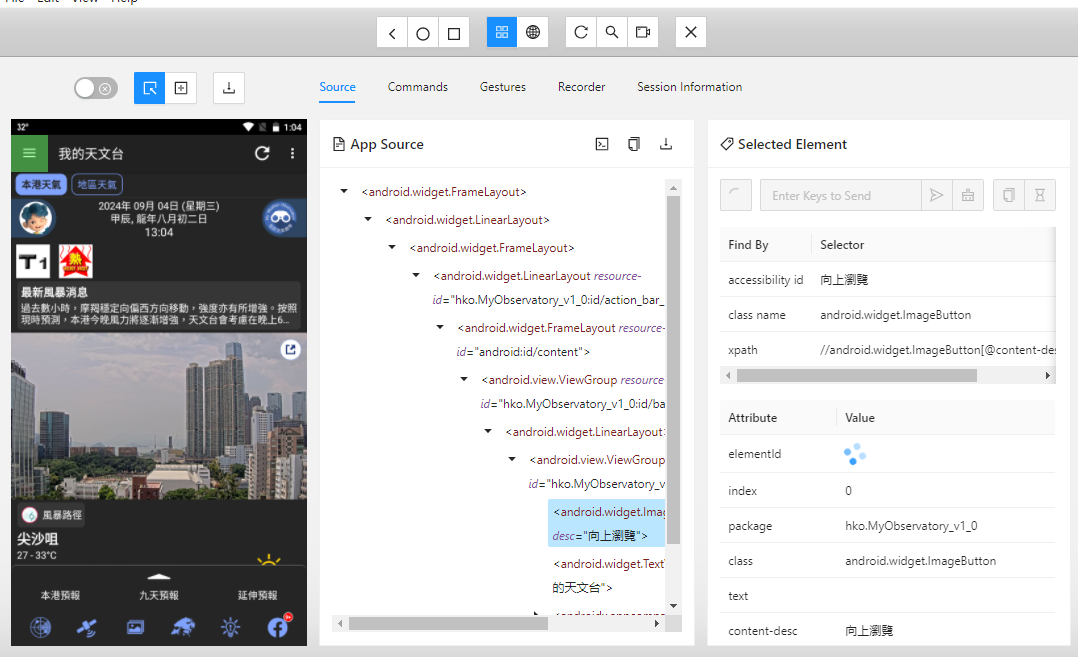
* 1. 启动Appium Inspector



JSON文件解释如下，如果粘贴不支持注释，要去掉注释。

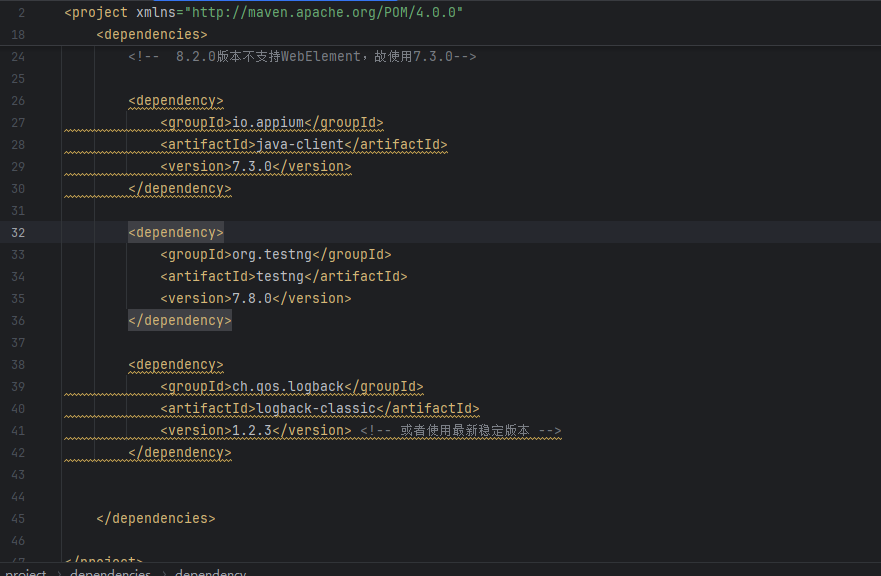
1. {
2. "platformName": "Android",  *//平台名称*
3. "appium:platformVersion": "7.1.2", *//平台版本，我这里是安卓7.1.2*
4. "appium:deviceName": "xguctgkfcuwcvonb", *//通过adb驱动获取的设备名称*
5. "appium:appPackage": "hko.myobservatory", *//包名，通过手机应用信息也可以找到*
6. "appium:appActivity": ".agreementPage" *//指定了应用内部的具体活动（某个界面）的名称。这里是登录页。可以文本格式打开apk找到Manifest获取*
7. }
   1. 会话冒烟

当第③步顺利完成，点击Start Session后，就可以正确的打开需要测试的app和界面了，如下图：

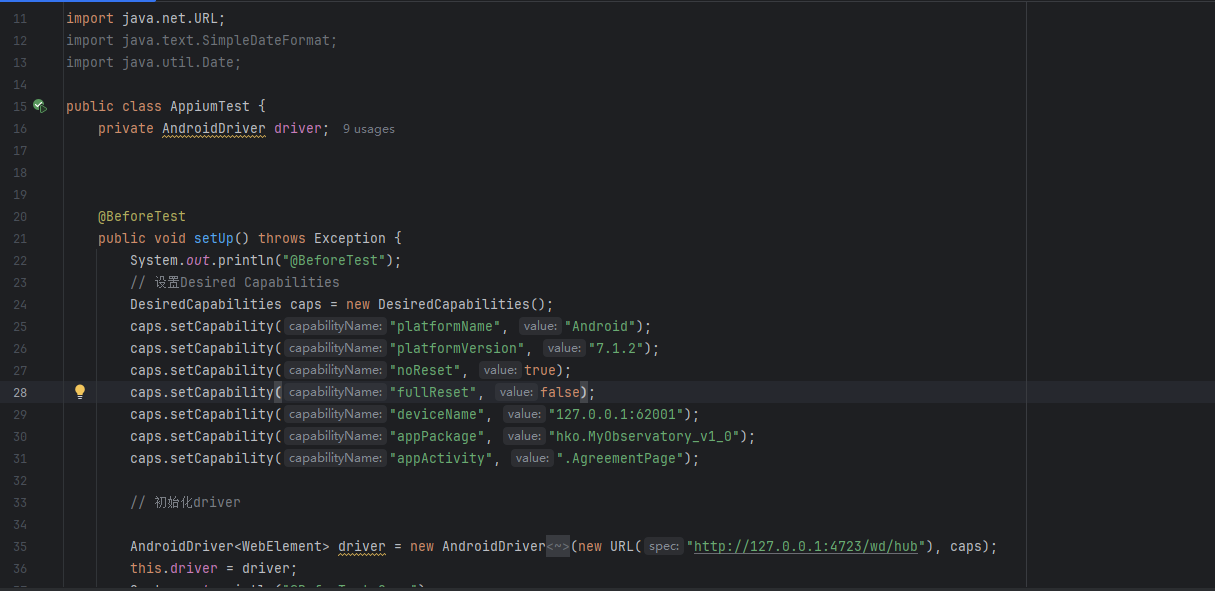


### 启用IDE环境

1. Maven参数配置：



编写测试类，实现登录测试：



执行如上测试类，实现测试。并得到测试结果如下：

