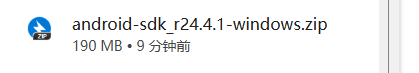
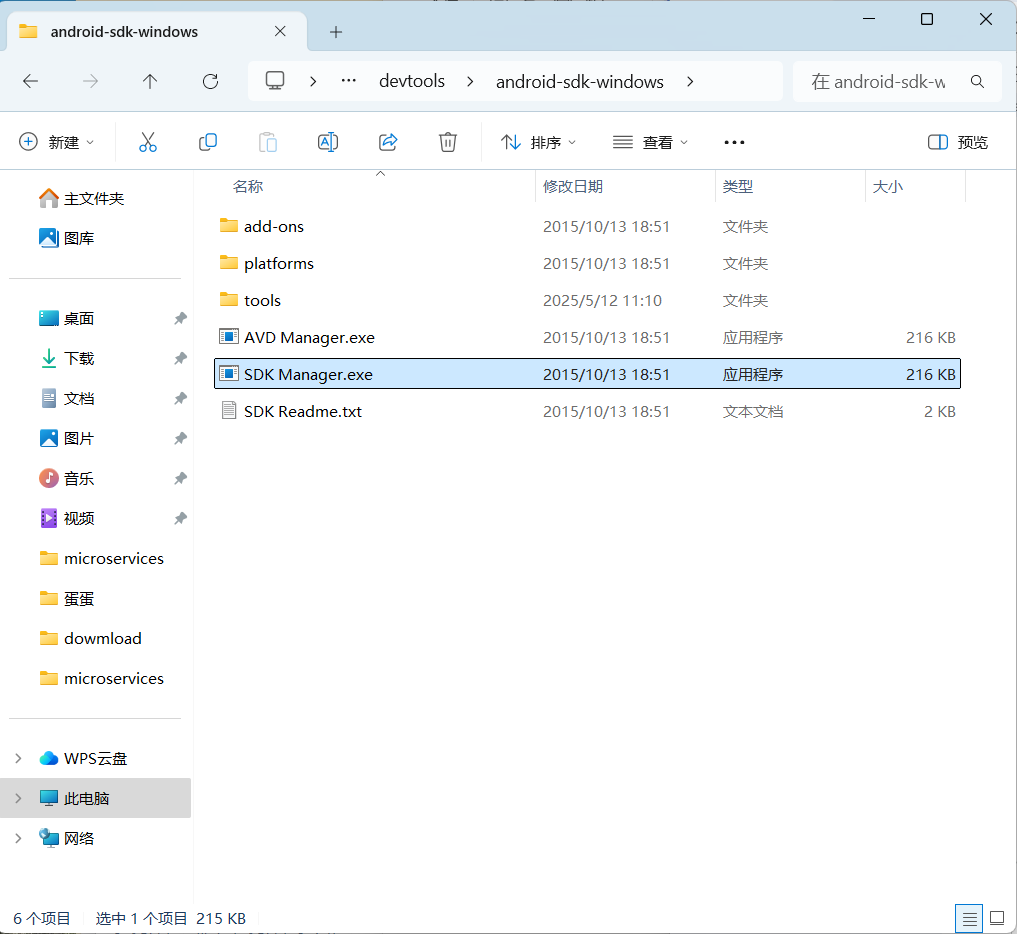
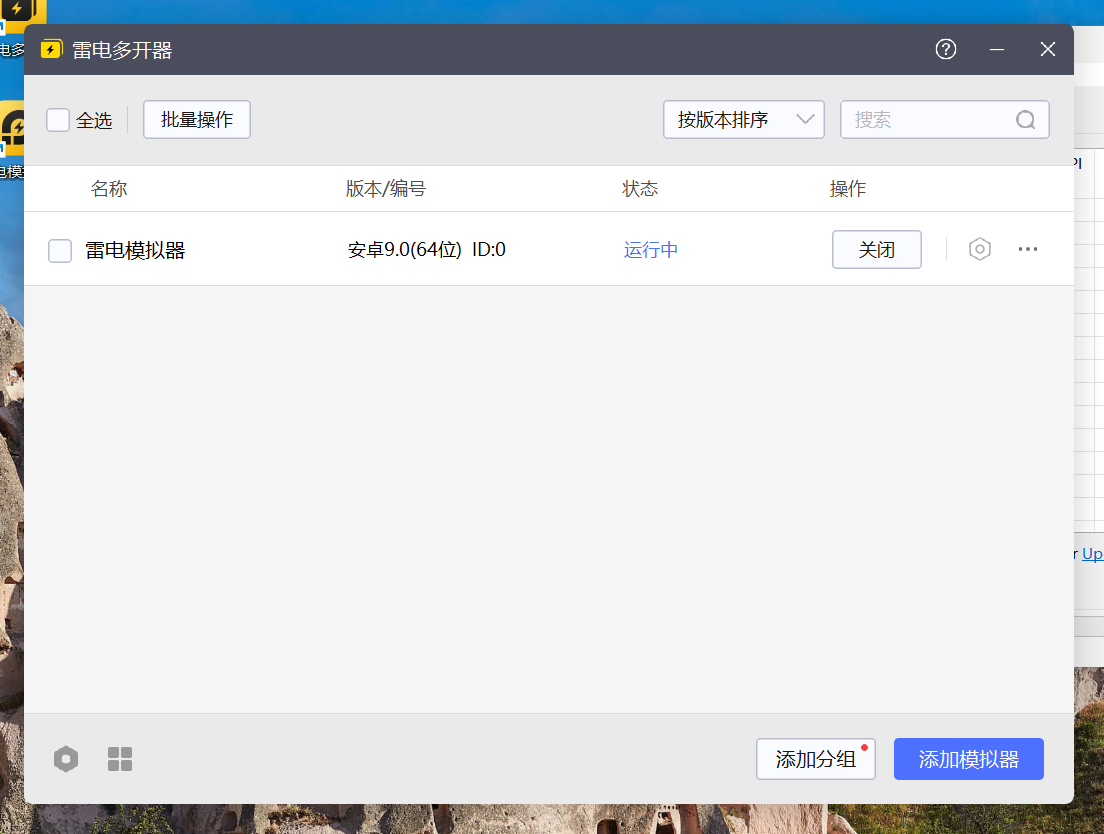
下载android sdk工具



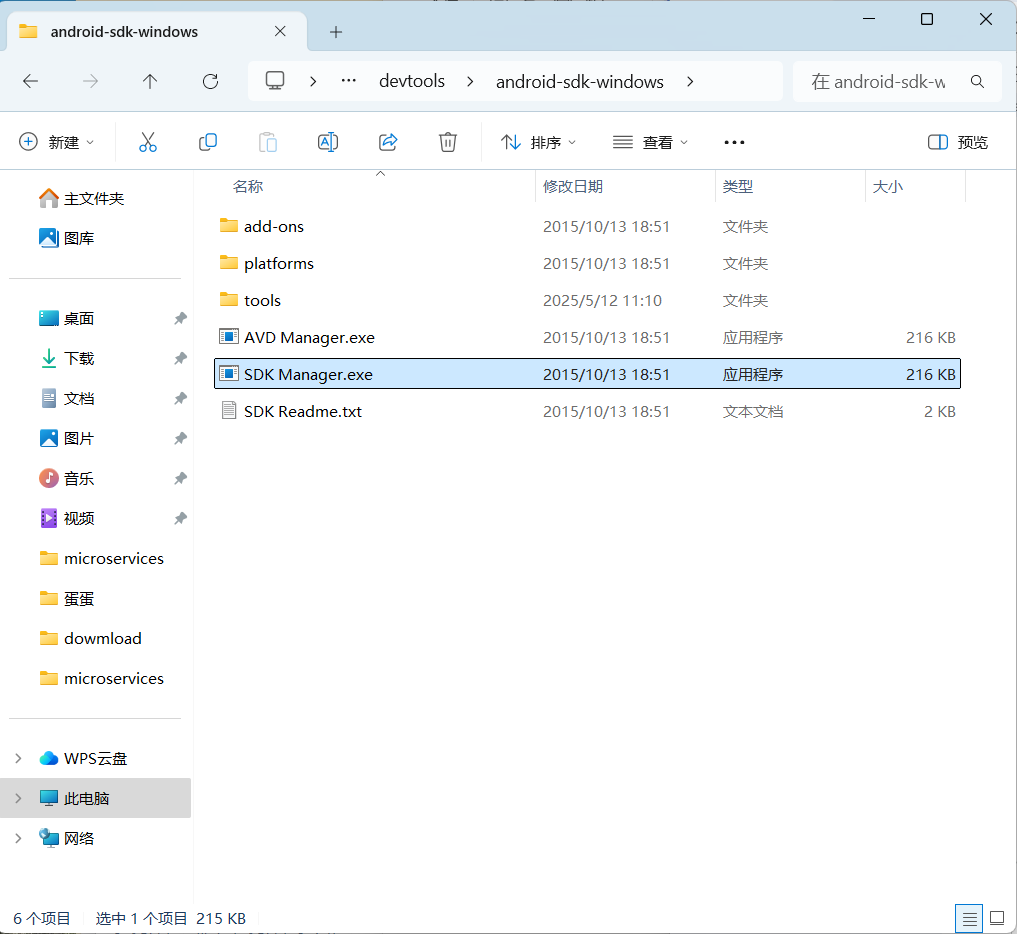
解压



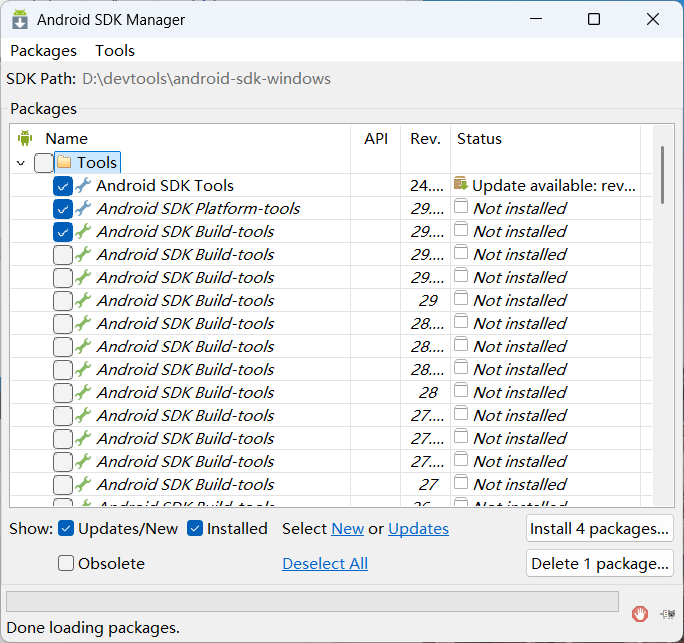
安装雷电模拟器



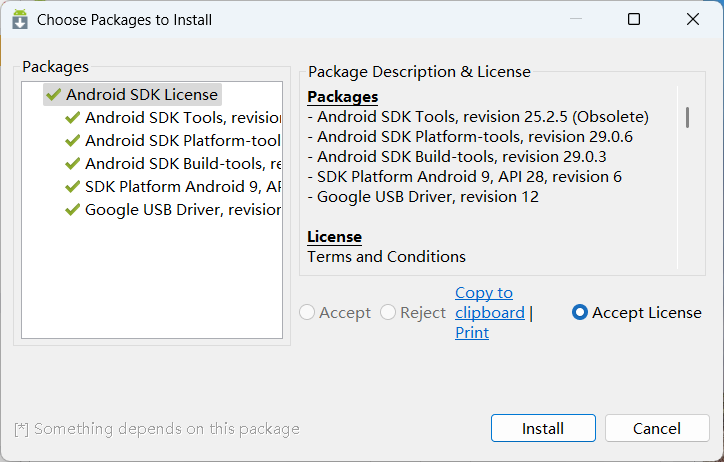
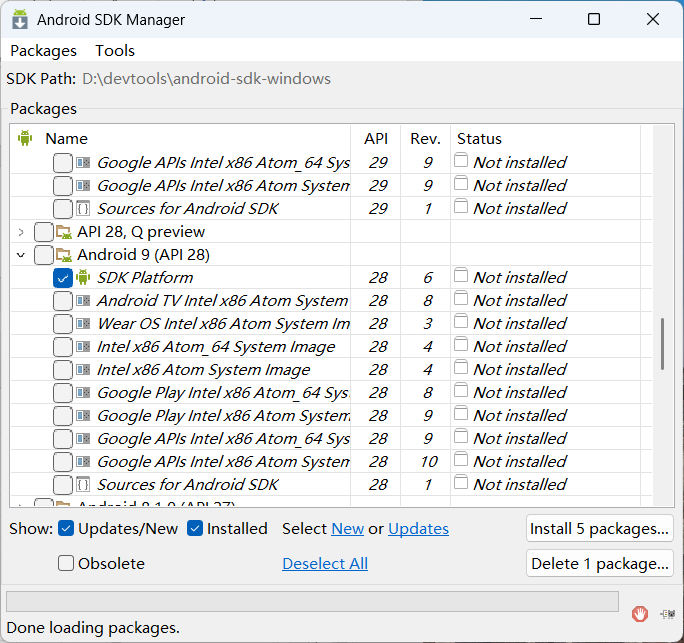
看到是安装9，后面会填入json



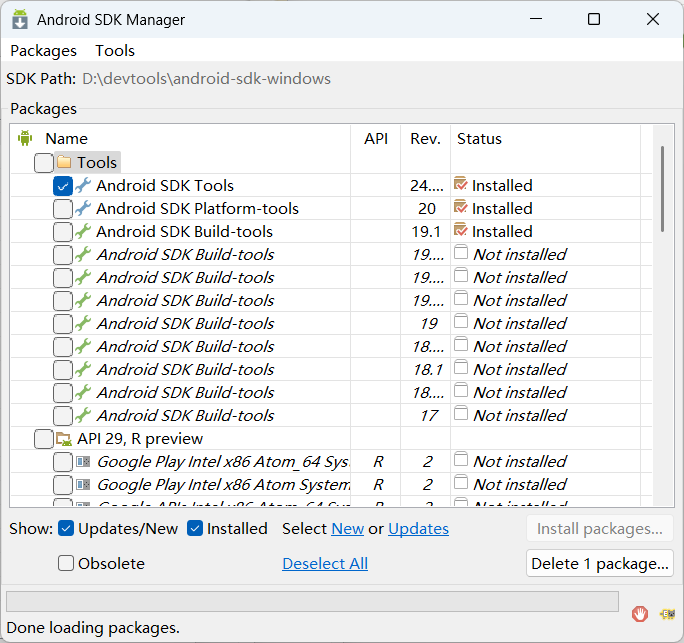
点击sDK Manager.exe



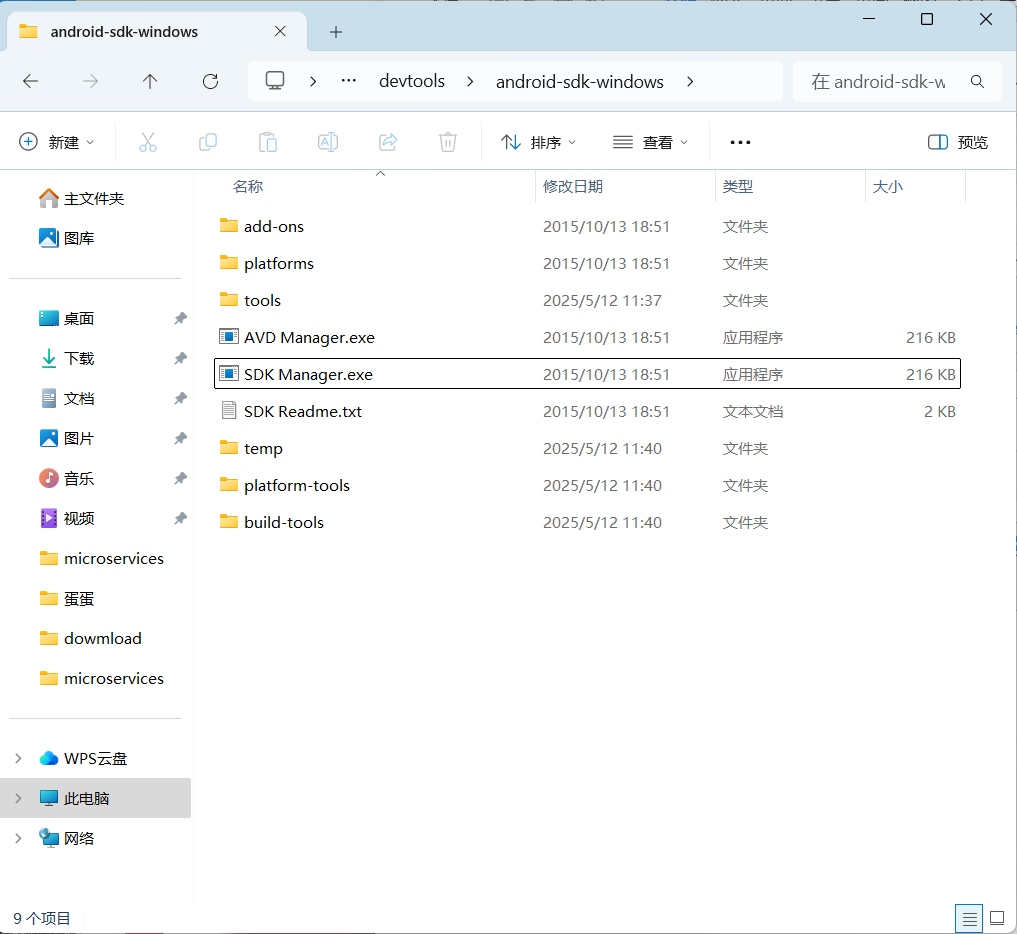
安装需要的包



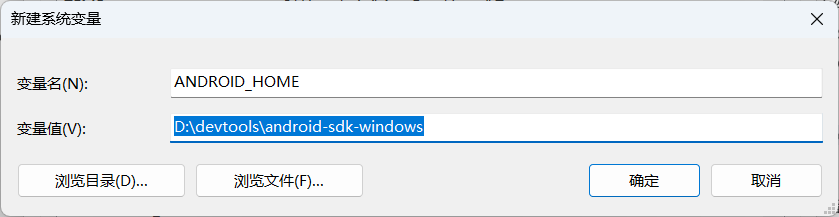
安装成功

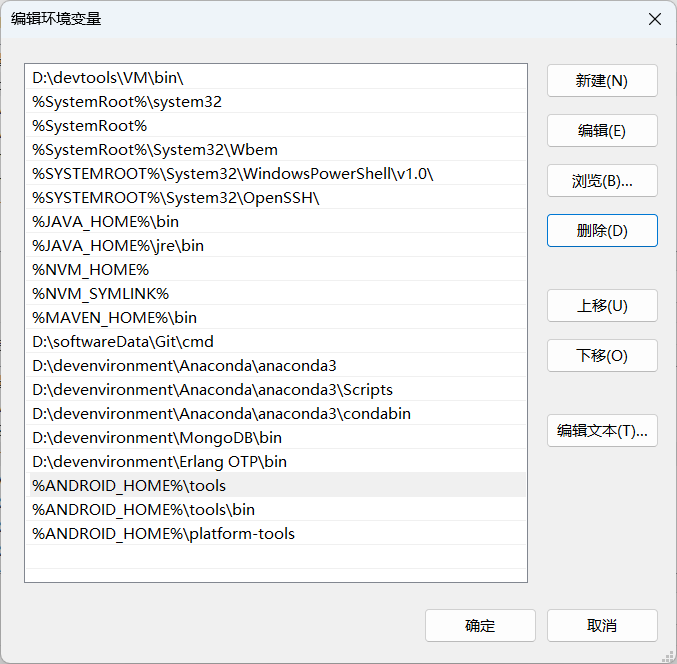


这个就是成功后的文件夹

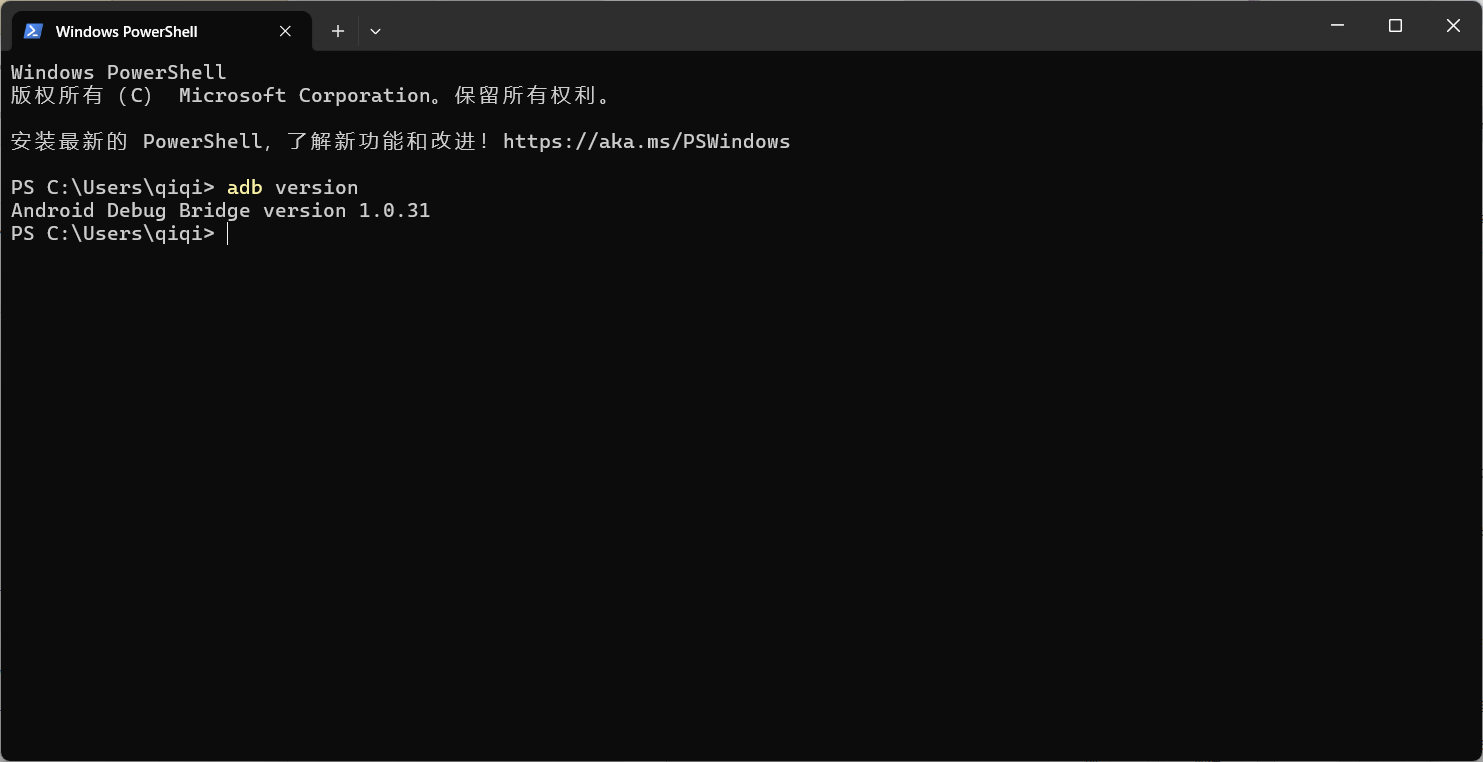


配置环境变量



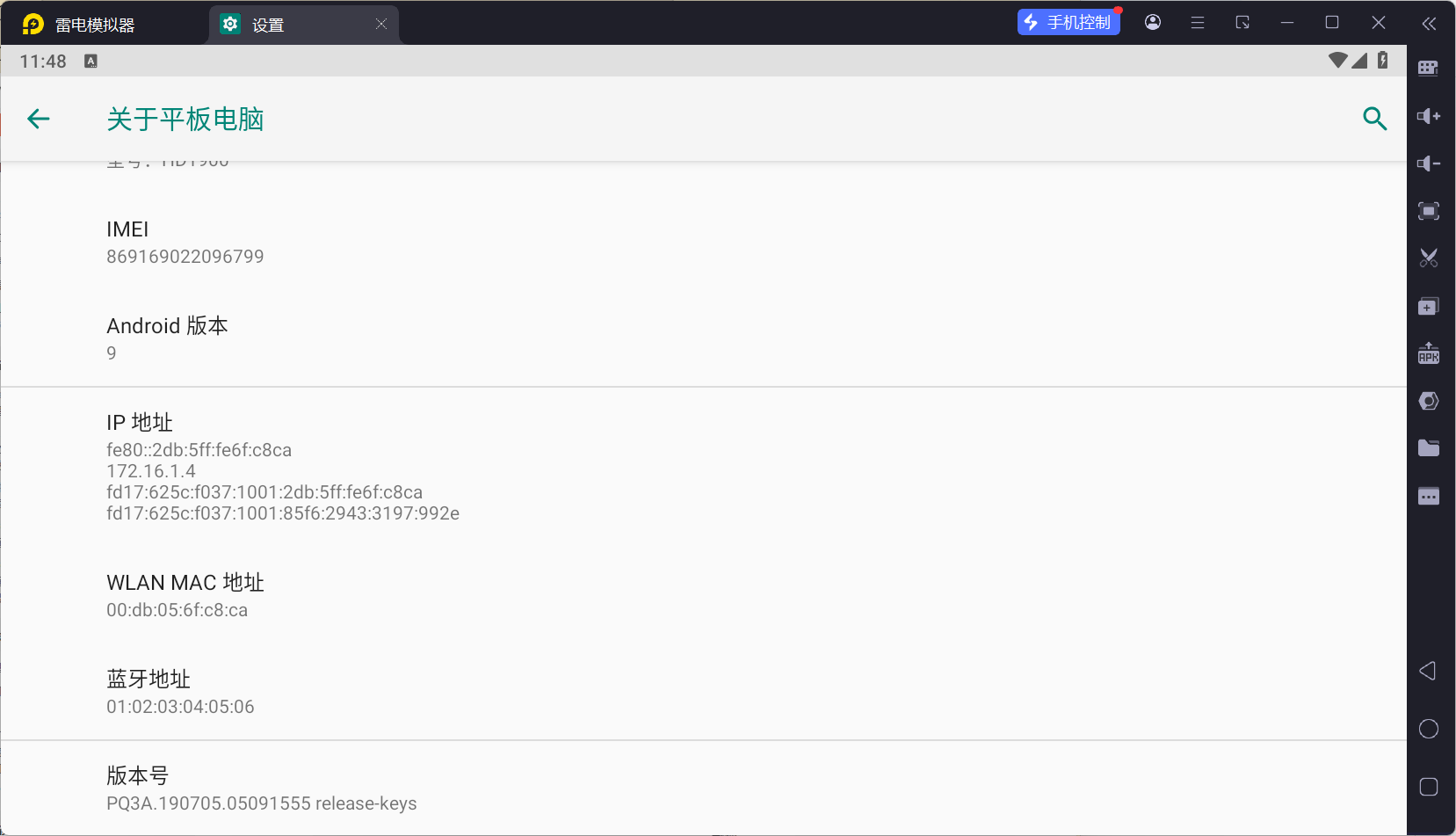


配置成功



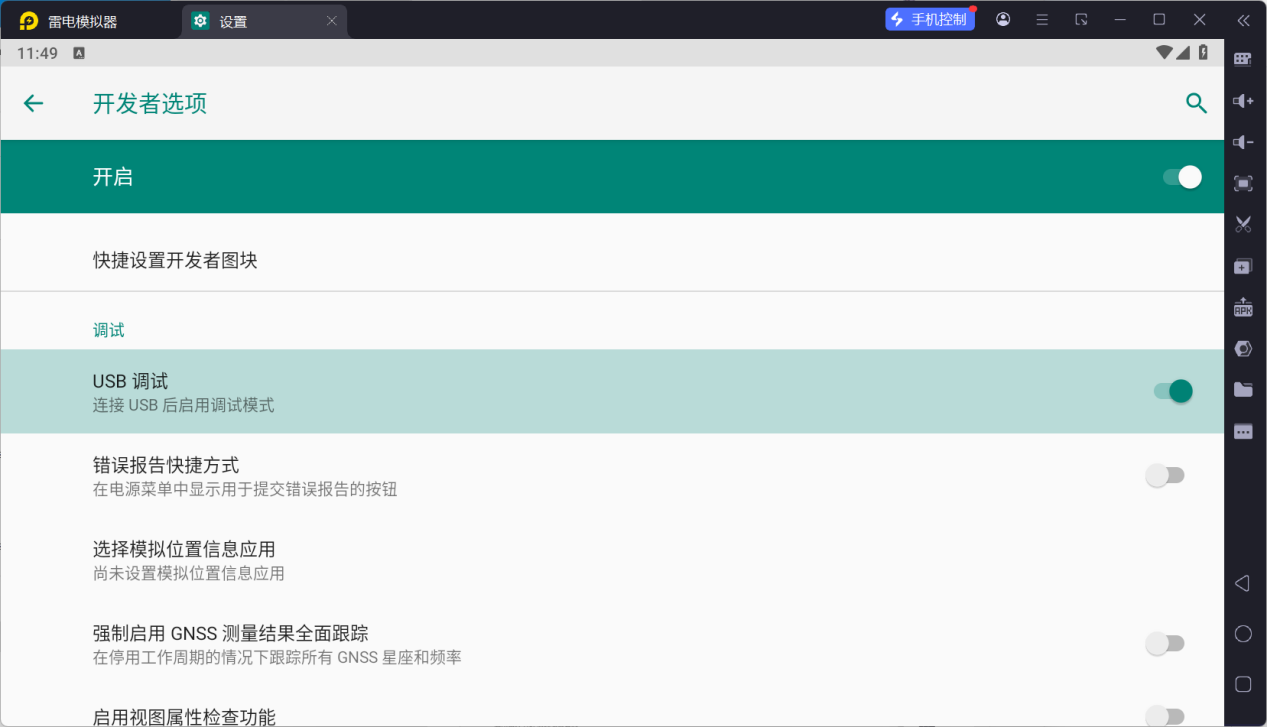
打开雷电，进入开发者模式

连续点击版本号即可进入





打开usb调试



终端执行

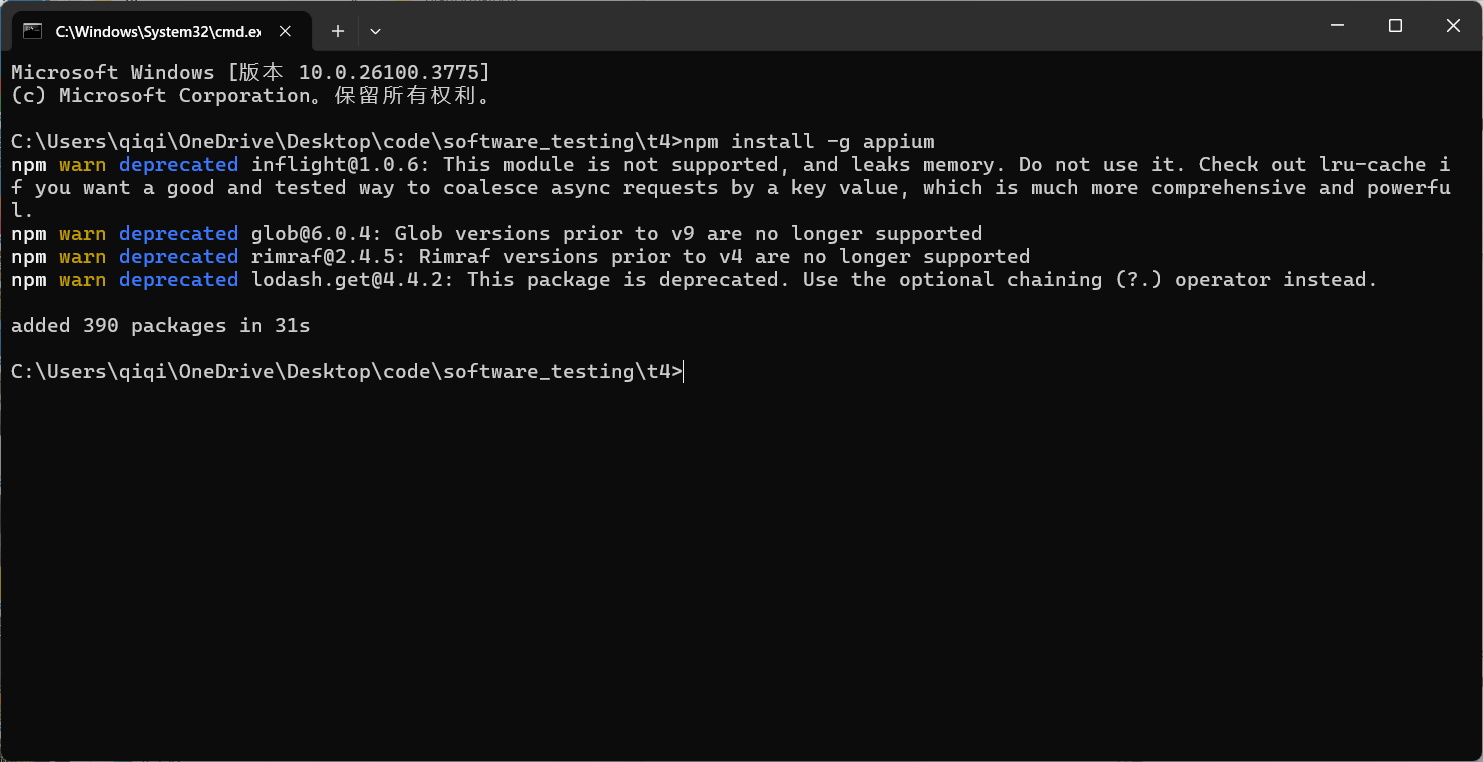
adb devices

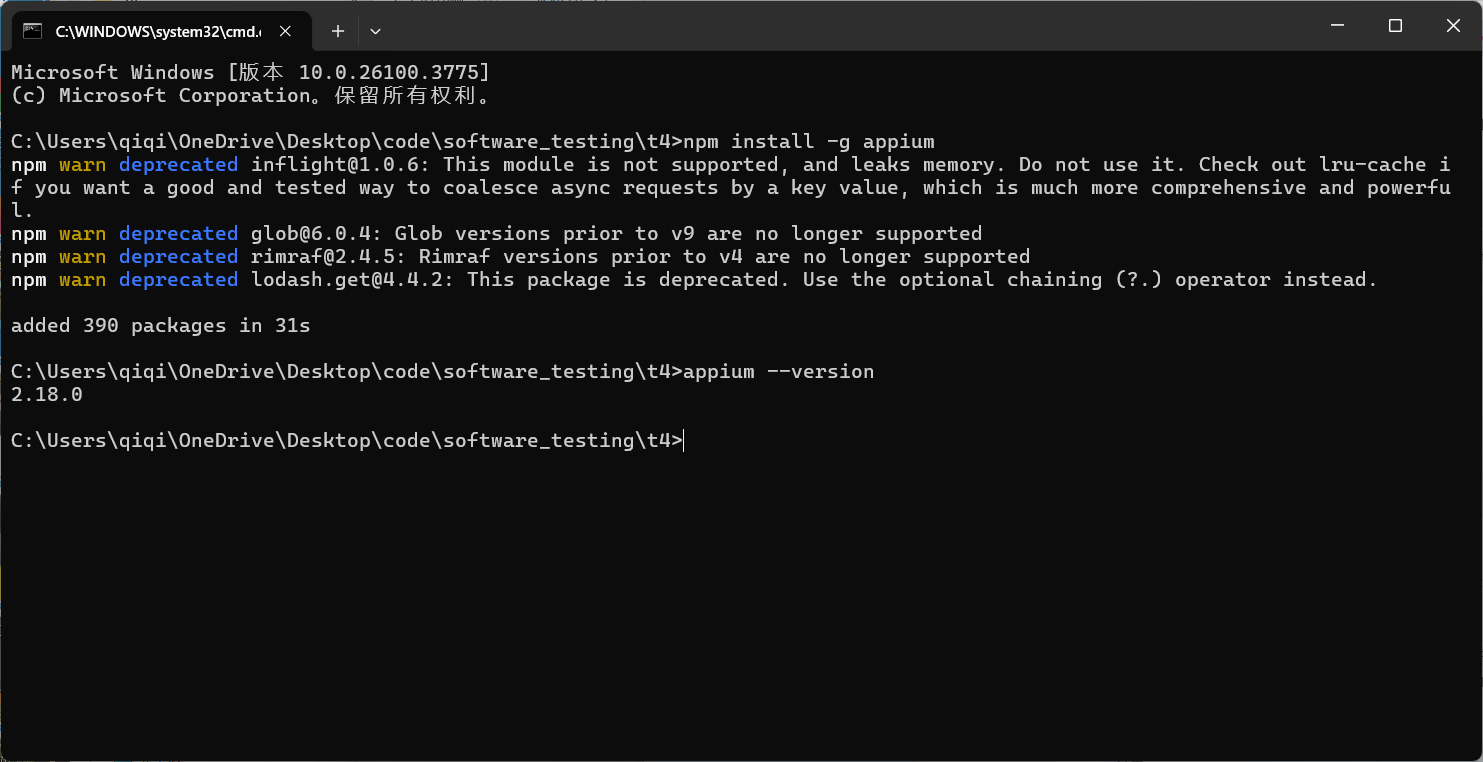


这就是成功连接了

**安装 Appium 2.0 版本：**

* 打开命令行窗口，输入以下命令进行全局安装：



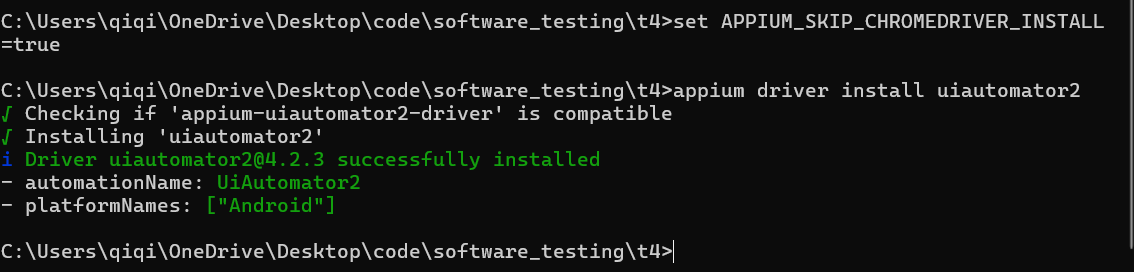


安装 UIAutomator2 驱动程序：

重要： 在安装驱动之前，先设置环境变量 APPIUM\_SKIP\_CHROMEDRIVER\_INSTALL 为 true

# Windows Command Prompt

set APPIUM\_SKIP\_CHROMEDRIVER\_INSTALL=true



安装成功

安装 Appium Python 脚本客户端：

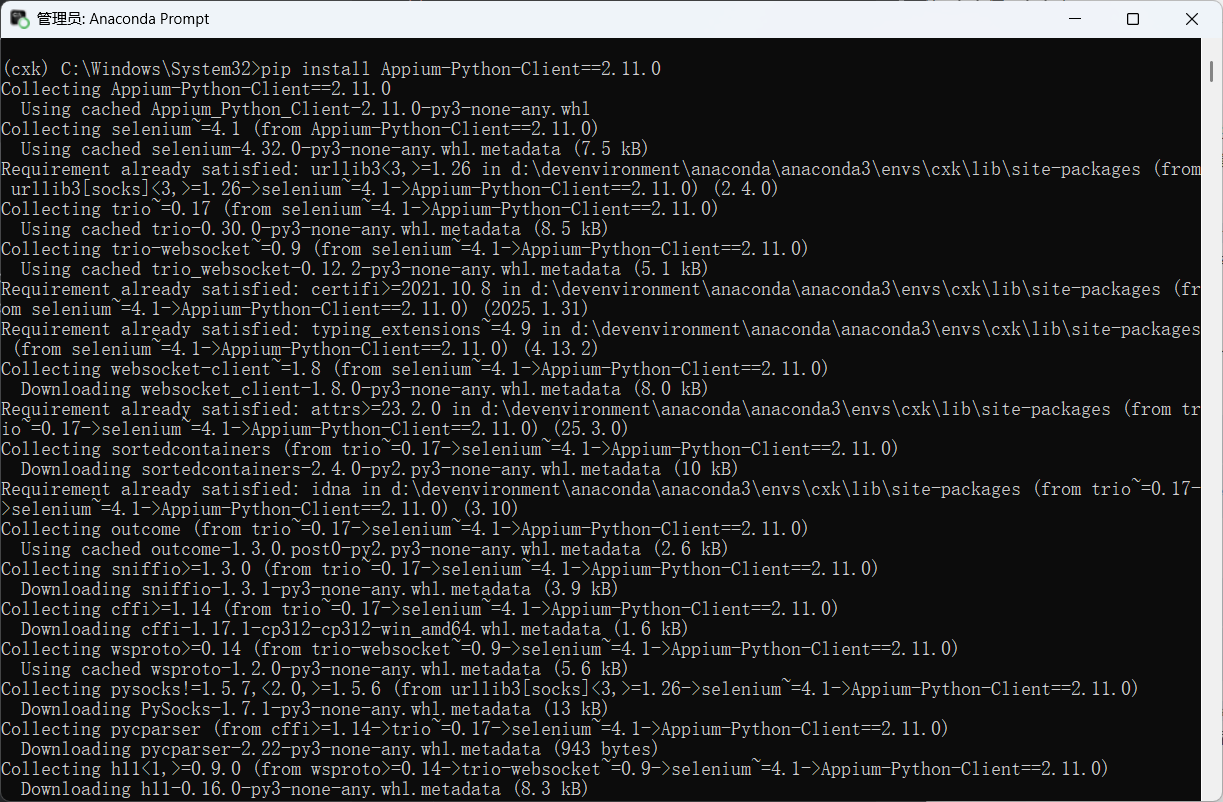
确保您已经安装了 Python。建议使用 Python 3.8 版本（根据实验指导建议）。

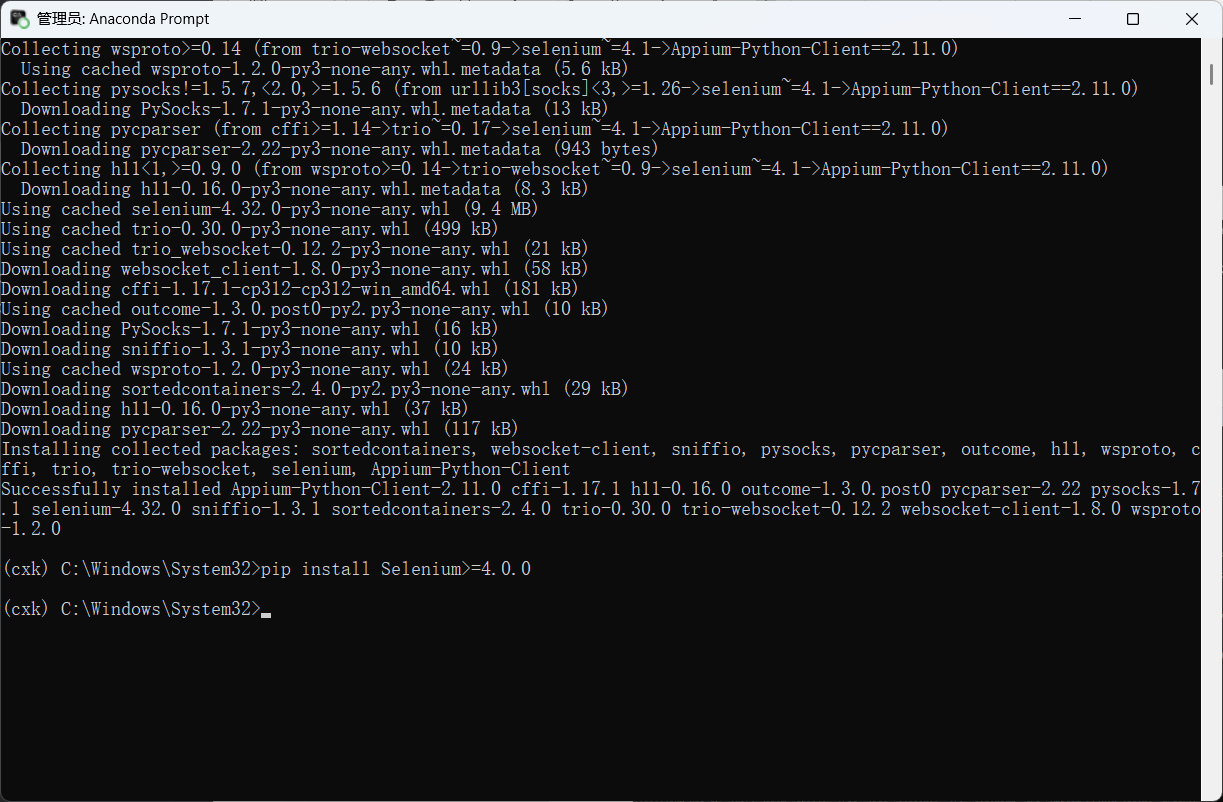
打开命令行窗口，输入以下命令使用 pip 安装：

Bash

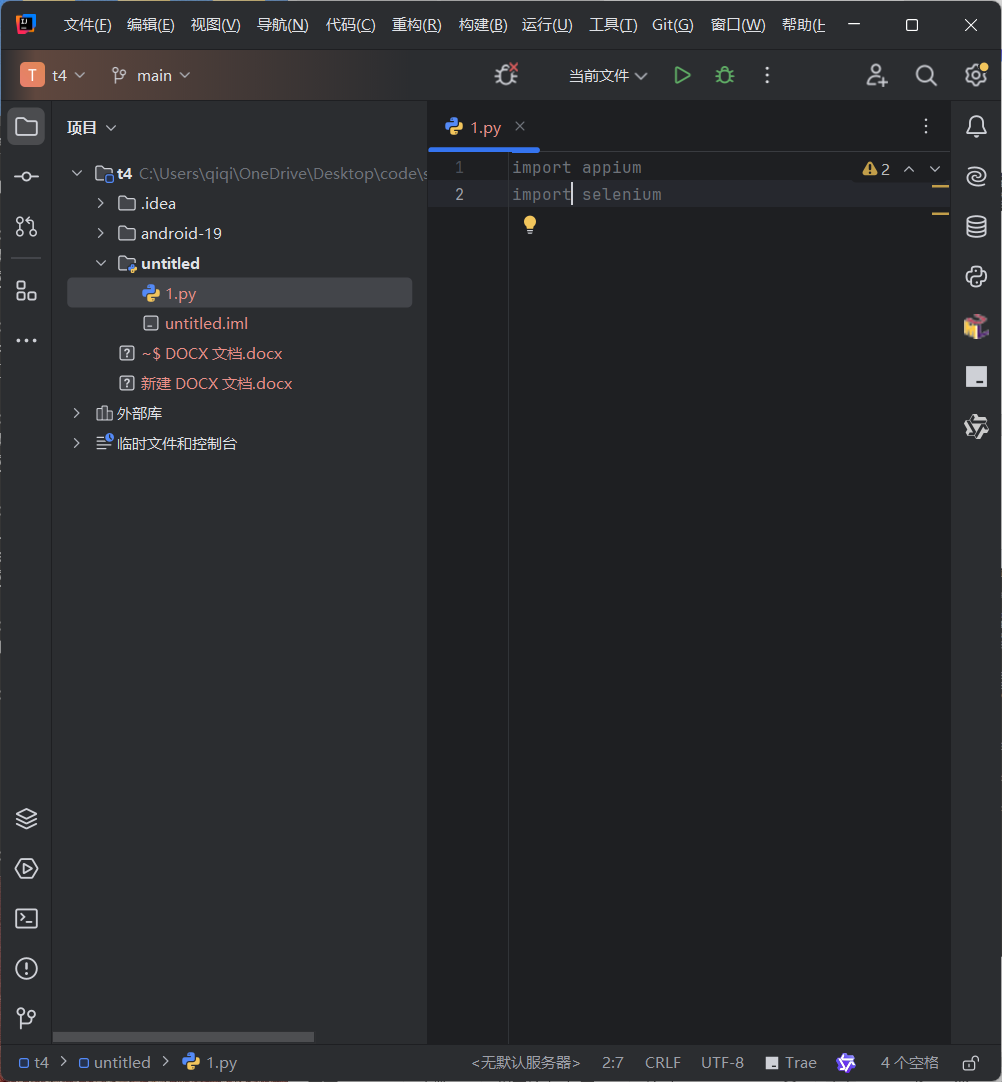
pip install Appium-Python-Client==2.11.0 # 根据实验指导建议的版本

pip install Selenium>=4.0.0 # 根据实验指导建议的版本



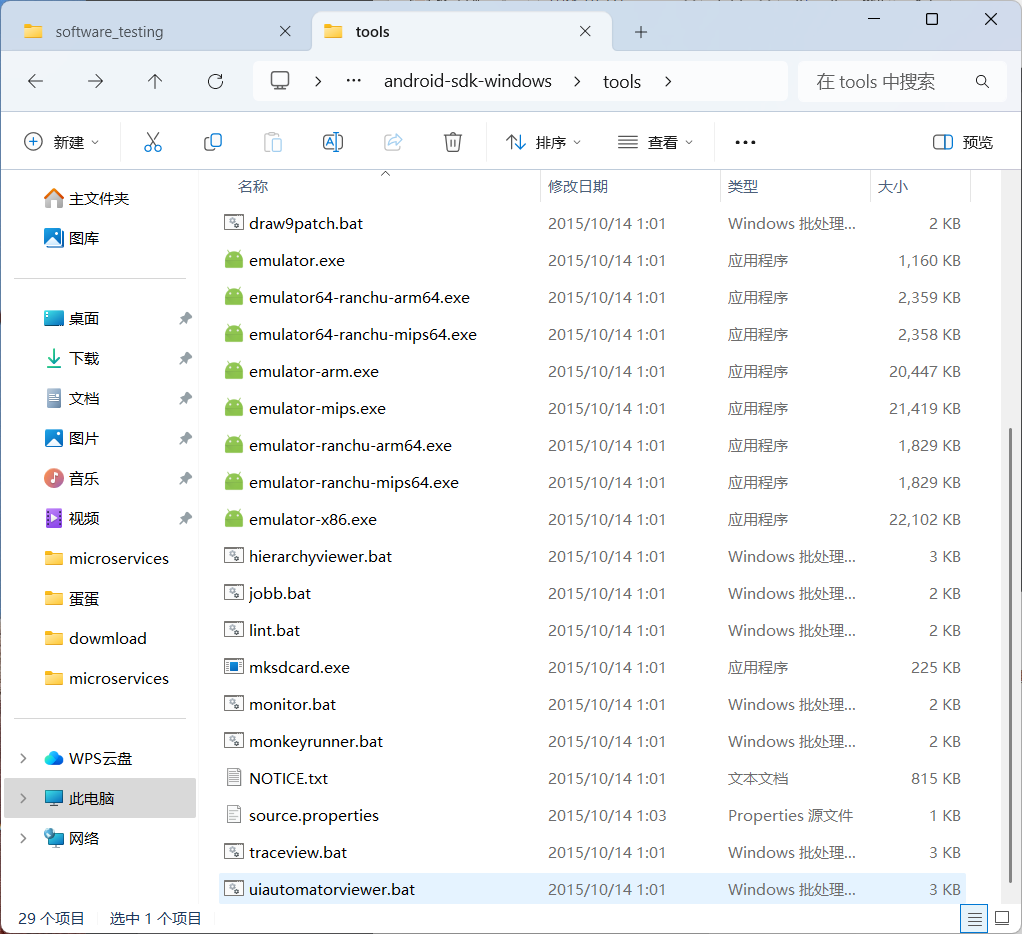


安装成功



没有报错说明安装python环境成功

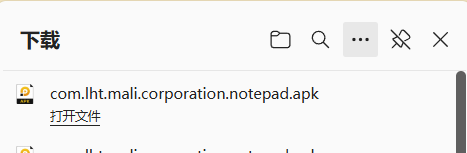
准备uiautomatorviewer工具



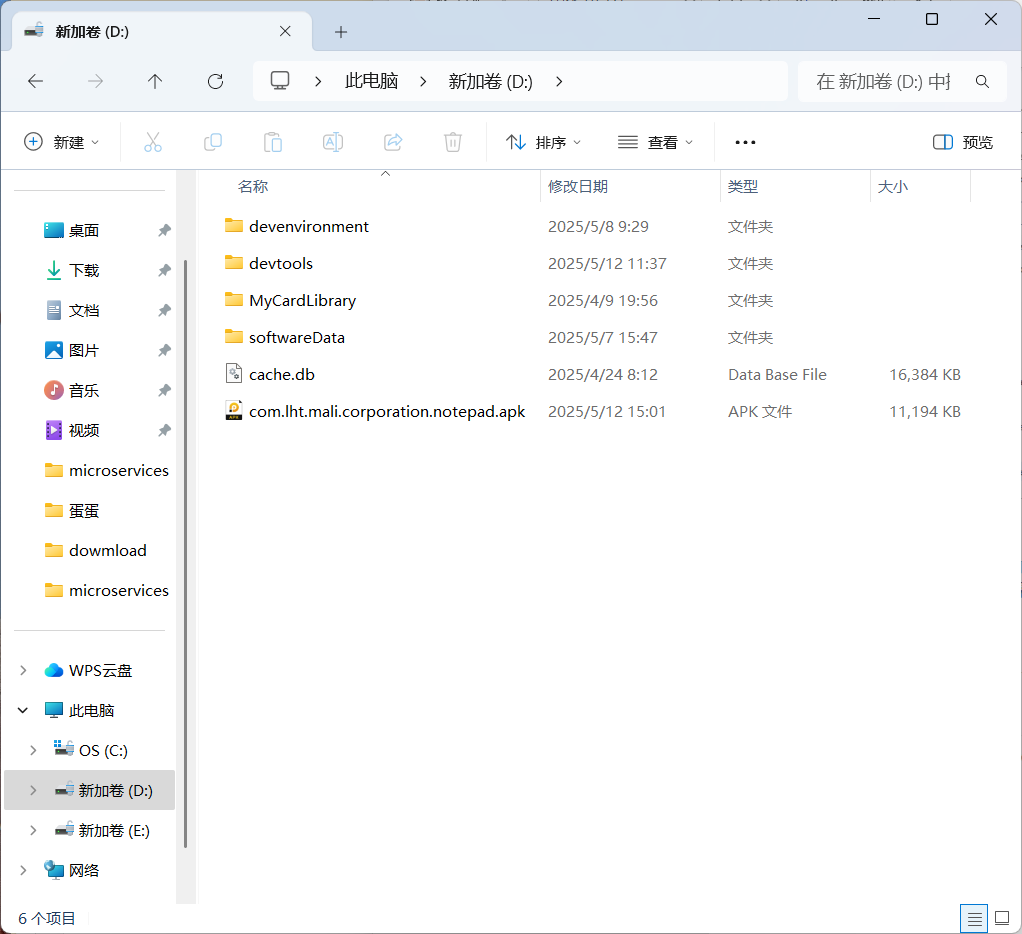
这个我用不了，jdk17，所以我用更先进的（后文）**Appium Inspector**

除非用jdk8，不过我不想，太老了。

**至此，环境搭建阶段完成。**



移动到"D:\com.lht.mali.corporation.notepad.apk"



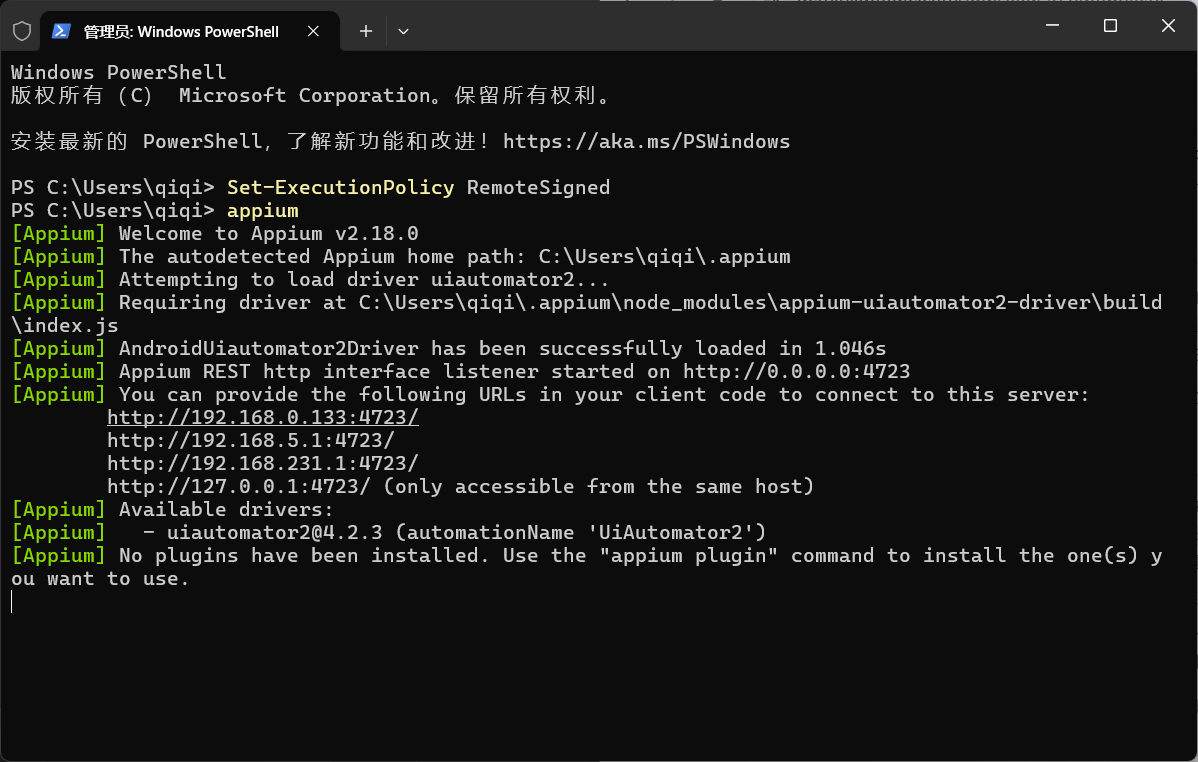
打开一个新的命令行窗口，

· **PowerShell（以管理员身份运行）**：这是一个打开修改的步骤，因为执行策略通常需要管理员权限。右键单击 PowerShell 图标，选择“以管理员身份运行”。

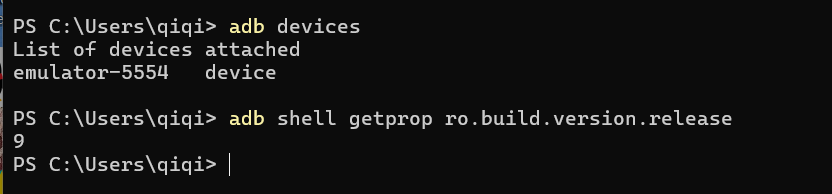
· **输入以下命令并按回车**：

· Set-ExecutionPolicy RemoteSigned

直接输入appium并回车。应该看到Appium服务启动，监听在默认地址http://localhost:4723。保持这个窗口开启，不要关闭。

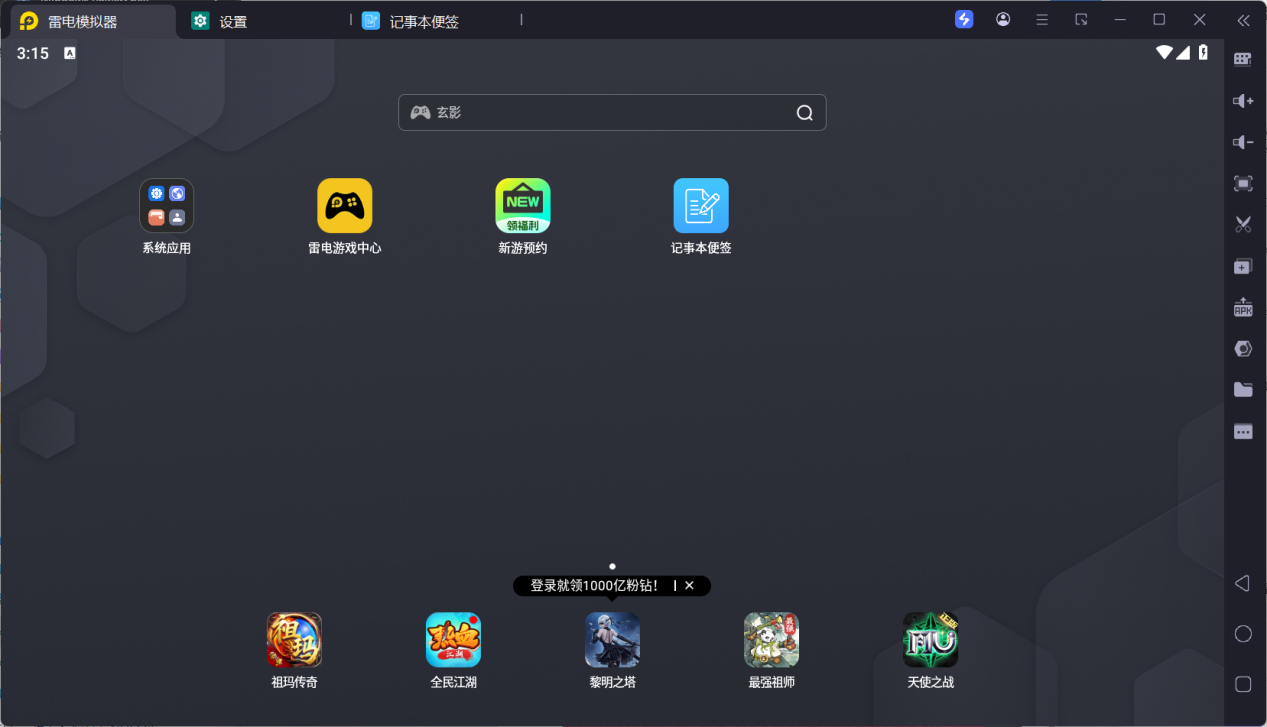


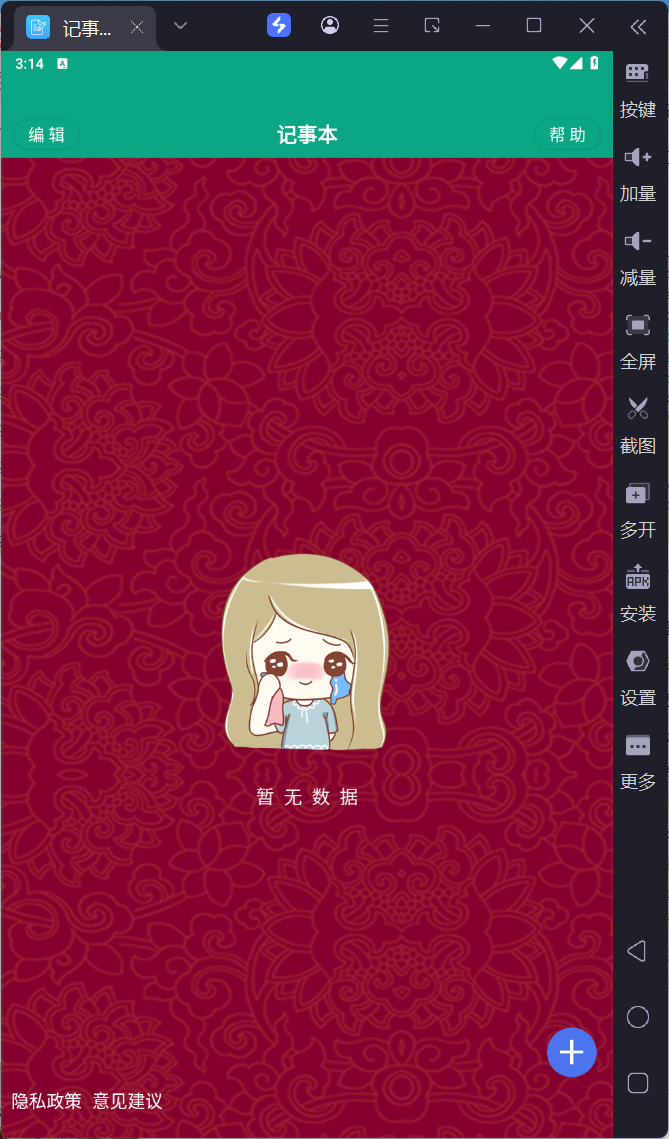
**查找**platformVersion**和**deviceName**：** adb shell getprop ro.build.version.release获取Android版本；adb devices获取设备ID。模拟器名称通常在模拟器窗口标题栏或设置里看到。



**查找**appPackage**和**appActivity**：**

* 确保设备已连接，Appium服务已启动。
* 在模拟器/真机上手动打开“记事本便签app”。

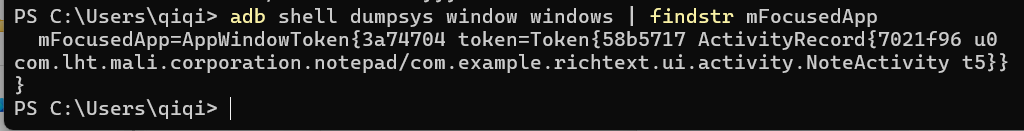




· 在模拟器或真机上，**手动打开并确保“记事本便签app”显示在屏幕上，并且是当前最难的应用程序。**

· **不要切换到其他应用程序，立即**返回您的命令行窗口。

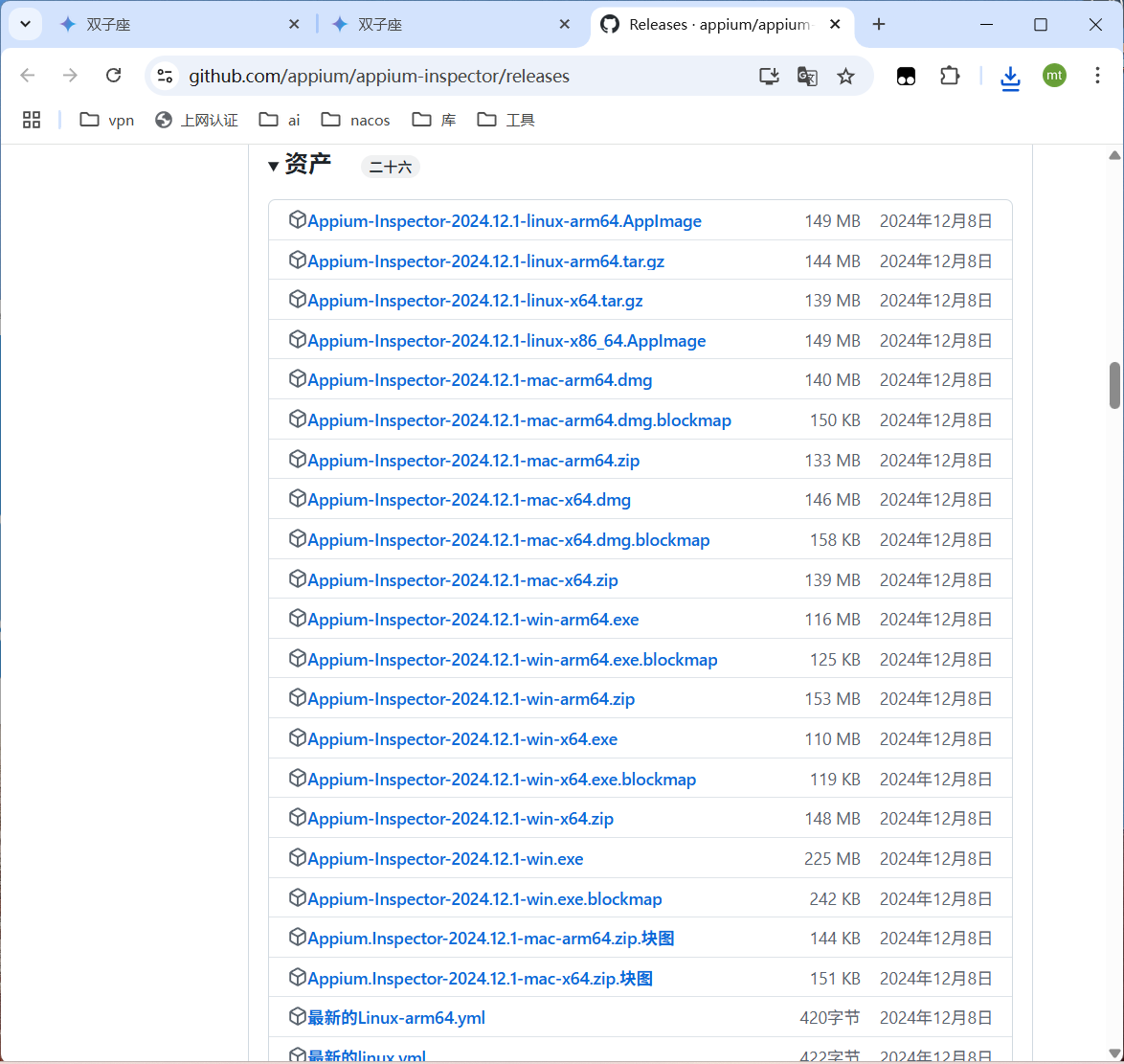
打开命令行窗口，输入adb shell dumpsys window windows | findstr mFocusedApp。



所以得到  


使用**Appium Inspector**来代替uiautomatorviewer。Appium Inspector 是 Appium 官方提供的一个功能更强大、更稳定且不易出现这种兼容性问题的 UI 元素检查工具。

1. **下载 Appium Inspector：**可以从 Appium 的 GitHub Release 页面下载最新版本的 Appium Inspector 桌面应用程序。通常可以在这里找到：[https://github.com/appium/appium-inspector/releases](https://github.com/appium/appium-inspector/releases" \t "_blank)下载对应您网络的版本（如.exeWindows 的文件）。
2. **启动Appium Server：**确保Appium Server已经在运行中（就是您之前打开的那个监听在http://localhost:4723的命令行窗口）。
3. **使用Appium Inspector连接：**打开下载好的Appium Inspector应用程序，配置连接信息（通常默认就是http://localhost:4723），然后点击“Start Session”连接到您的设备和应用就可以查看并定位元素了。



安装启动后填入json运行

{

"platformName": "Android",

"platformVersion": "9",

"deviceName": "emulator-5554",

"appPackage": "com.lht.mali.corporation.notepad",

"appActivity": "com.example.richtext.ui.activity.NoteActivity",

"noReset": true,

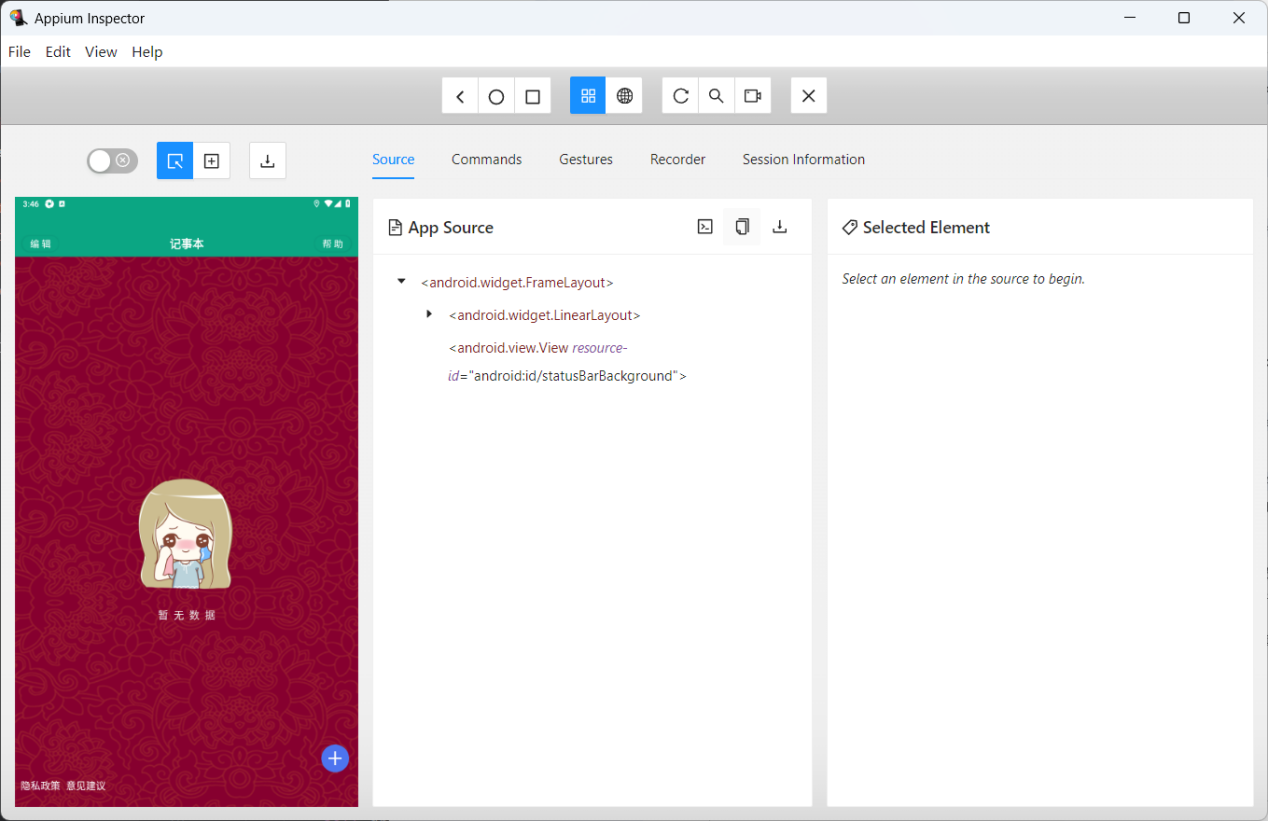
"unicodeKeyboard": true,

"resetKeyboard": true,

"automationName": "Uiautomator2"

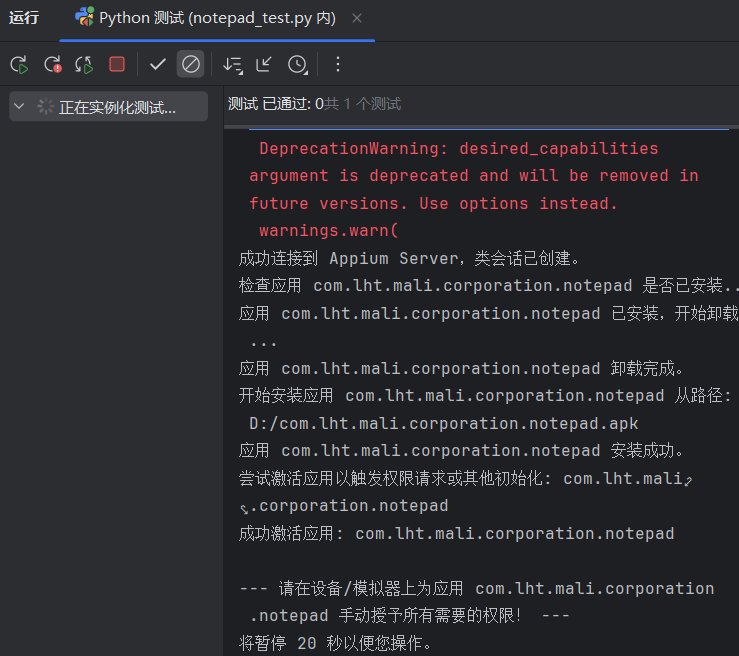
}

随后得到对应信息，对应修改脚本

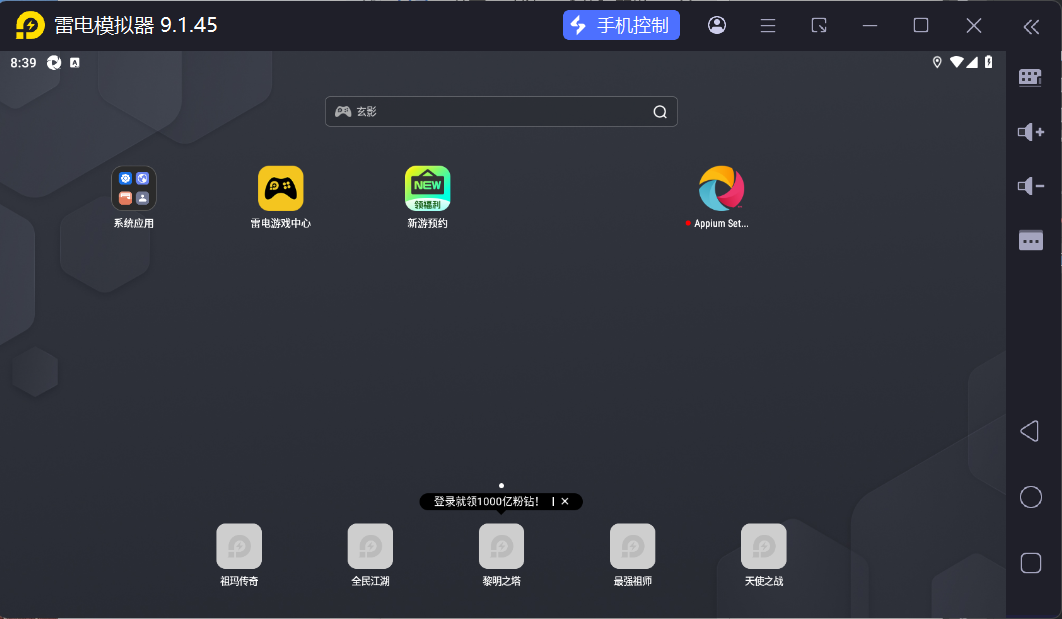


通过这个软件来获取页面信息，方便脚本编写

**开始测试**



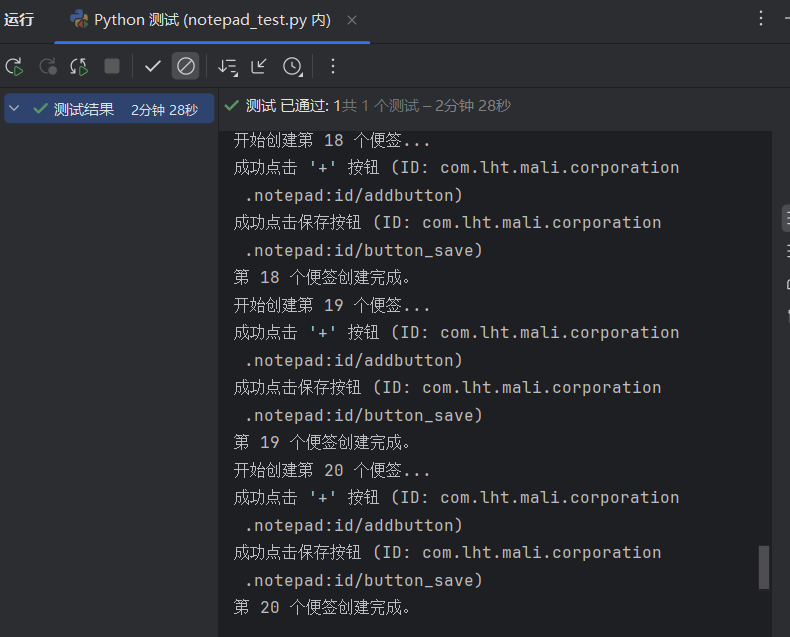
重装成功



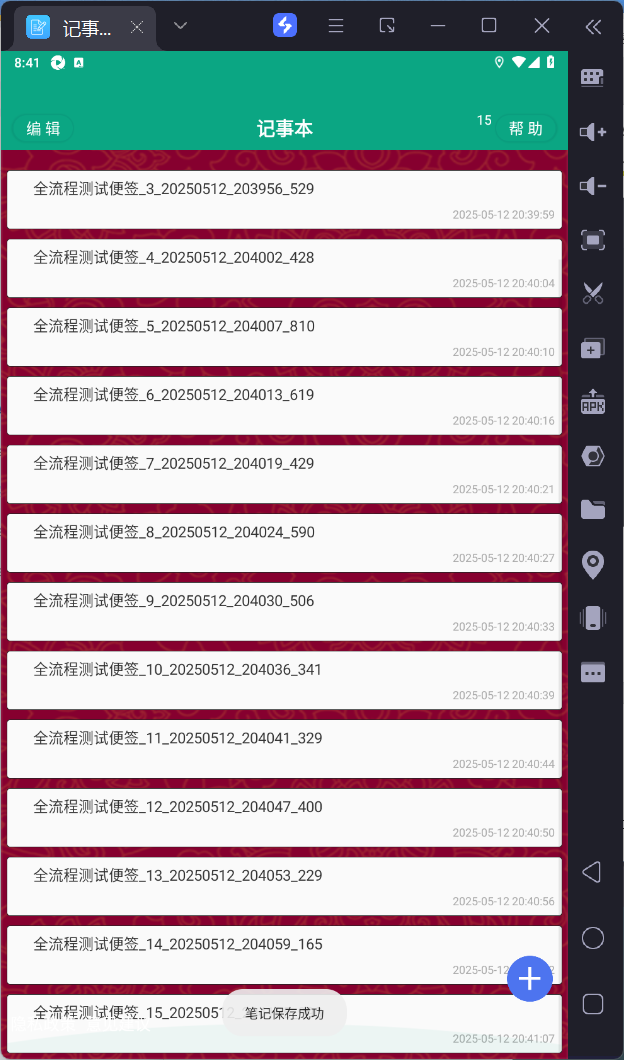
进入重新授权，这里我是手动等待授权，方便一些



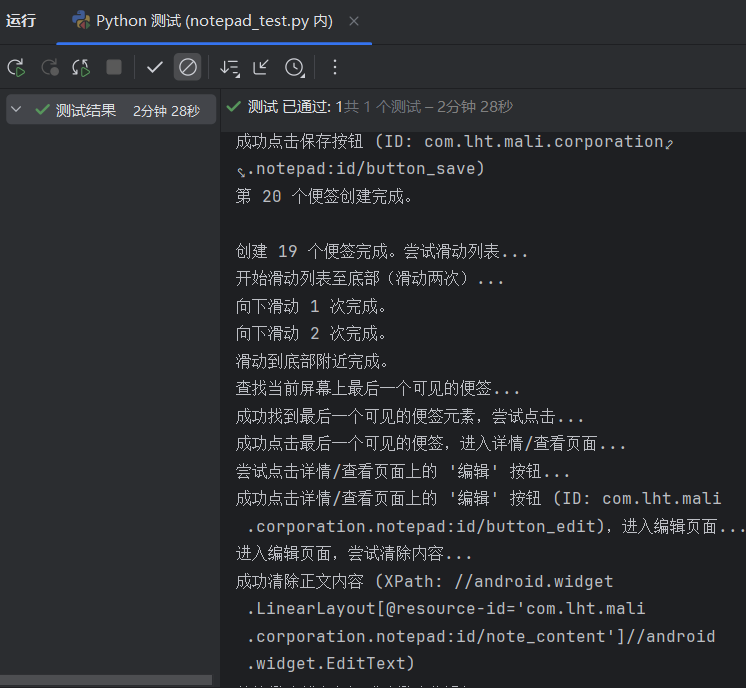
创建标签



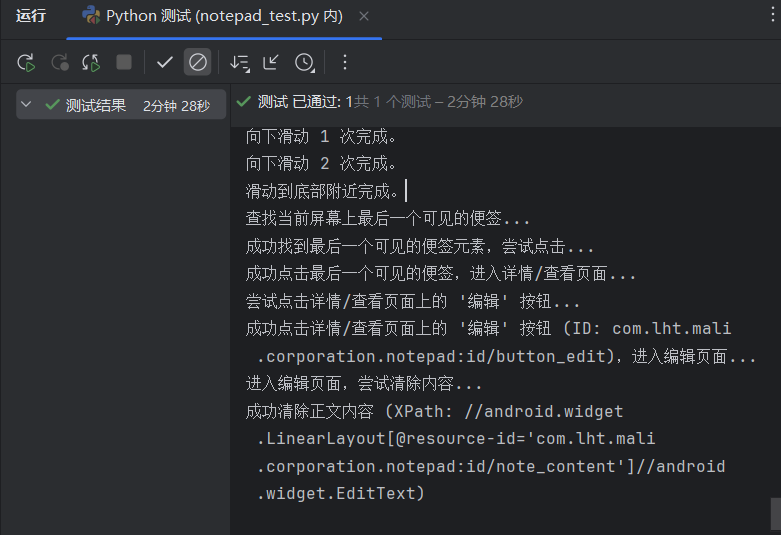
自动创建便签20条



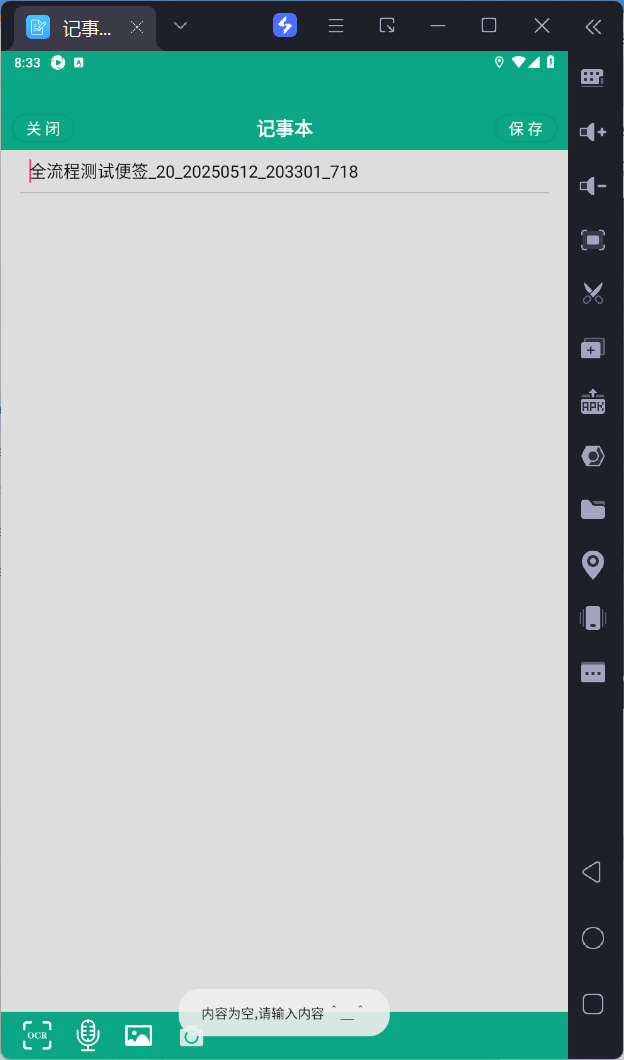
寻找最后的便签



进入编辑

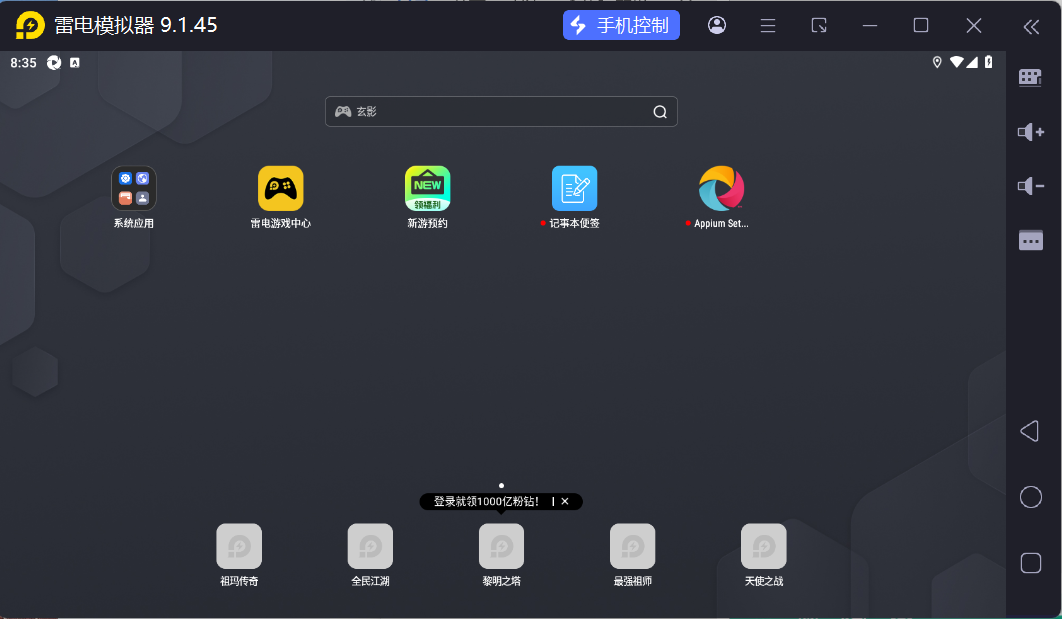


清空

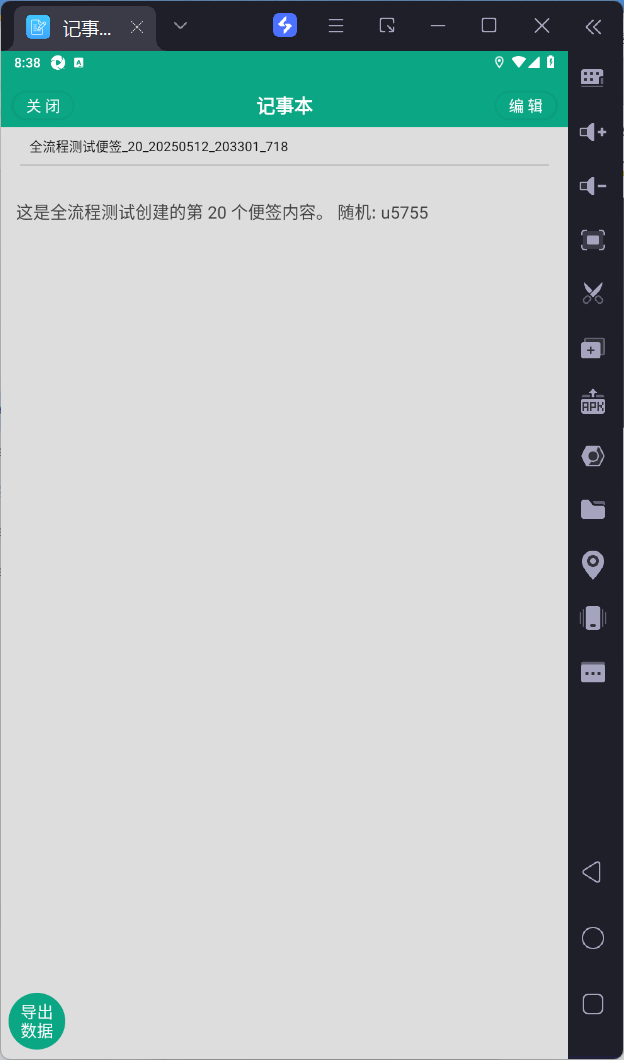


保存不了，显然不允许存空内容

自动退出app



我们重新进app发现



内容并没有变化，所以猜测正确，不能保存空内容

至此实验成功，测试通过



代码附上