app打包流程

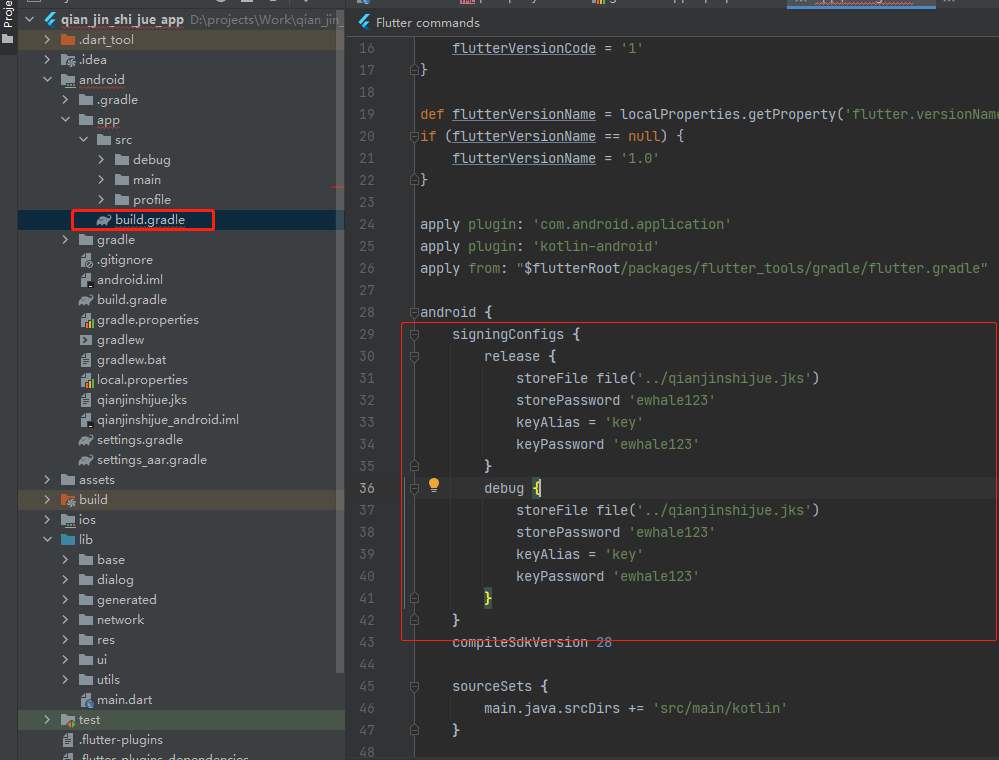
**（一）Android端**：

①方法一：使用命令行

flutter build apk；

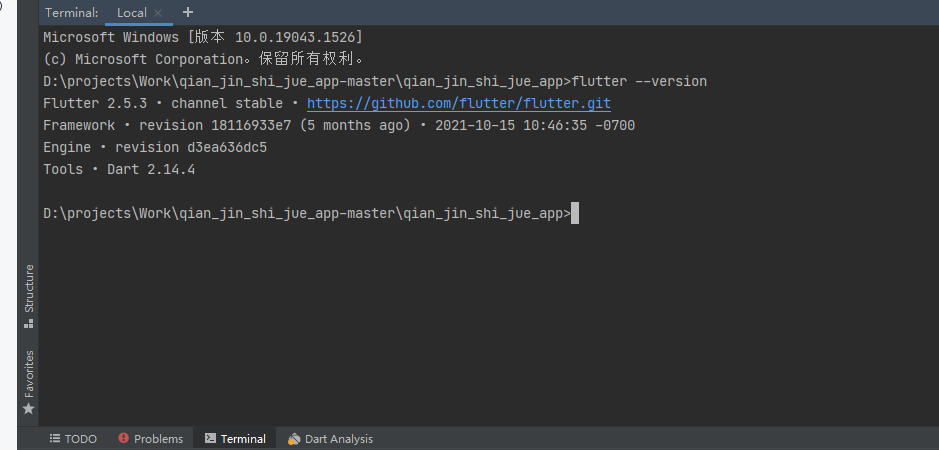
（如果没有对64位和32位具体的要求，默认打的包都是可支持64位和32位）

使用命令行前提条件：

在android -> app -> build.gradle 文件中有配置好相关的jks签名：如jks文件路径，别名（alias）,密码。项目中已经配置好了，可以直接使用命令行打包。  


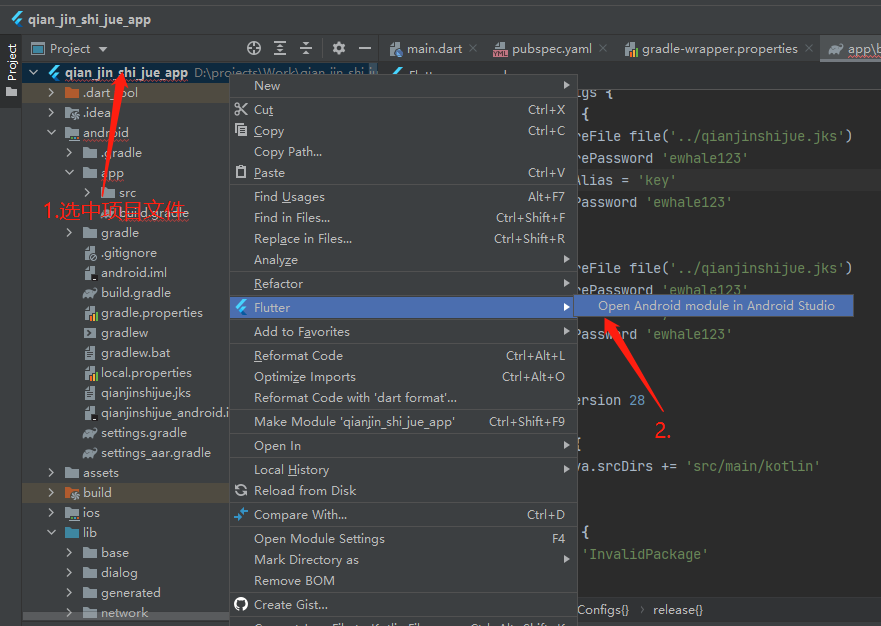
注意：使用命令行打包的时候，需要留意环境变量中的flutter版本和项目的flutter版本要一致。可在命令行输入flutter --version查看环境变量中的flutte版本

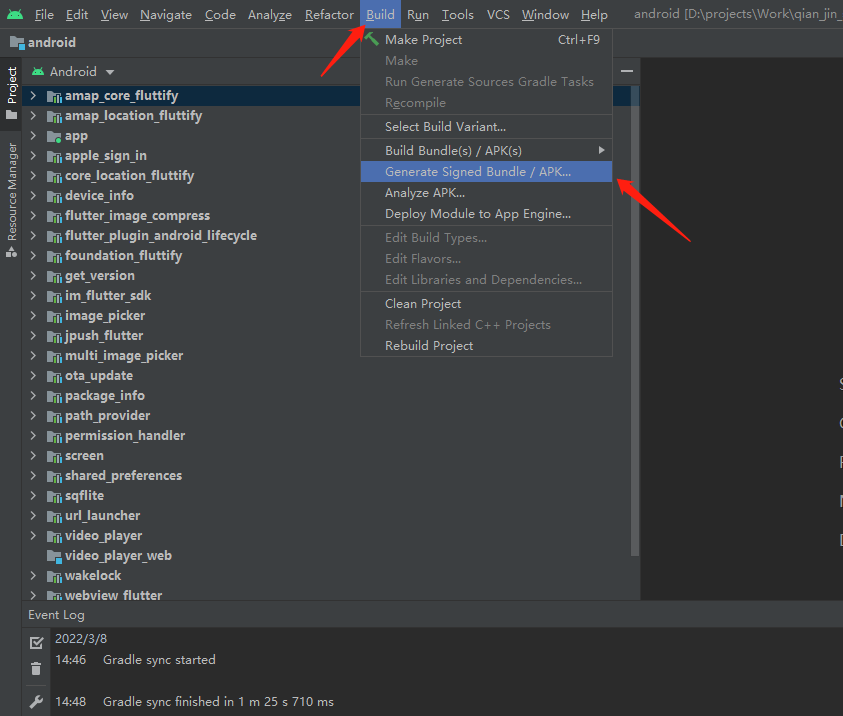
版本对的话，在控制台输入flutter build apk即可

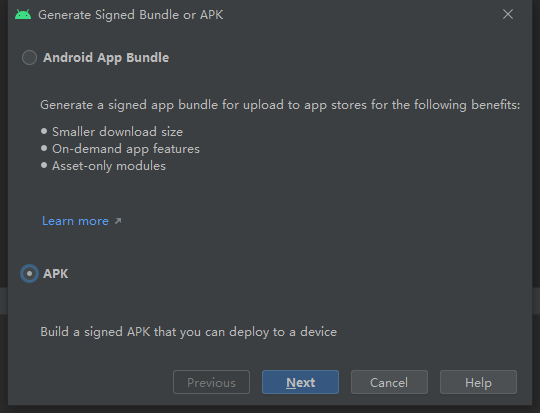


②方法二：跳转到android module

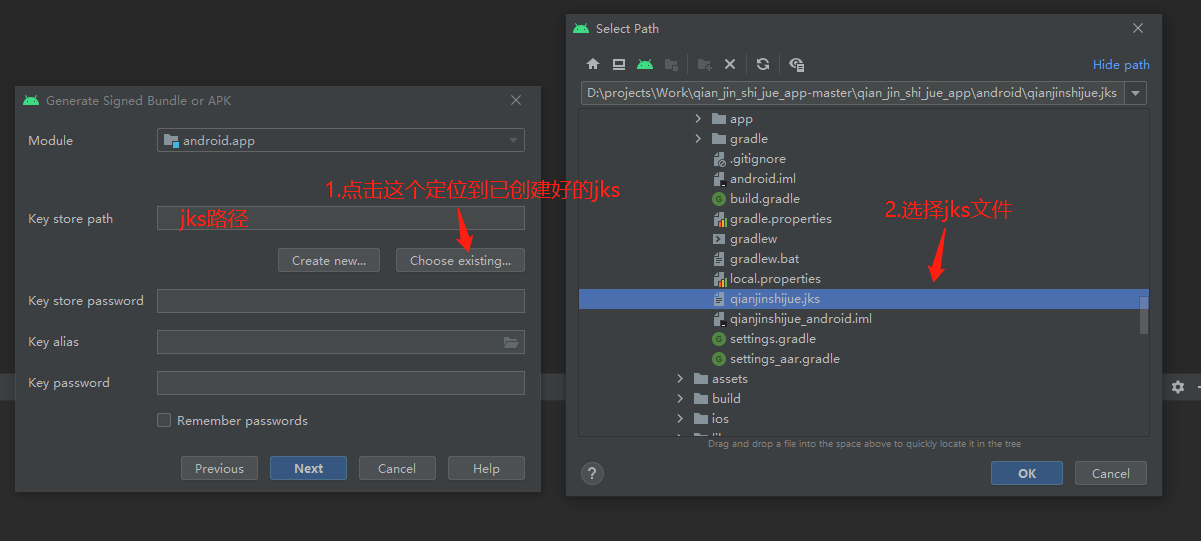
第一步：将项目通过android面板方式打开

等待项目加载，加载好后，点击tab栏中的build -》Generate Signed Bundle/apk，选择APK模式。（如果是第一次的话，加载的时间会有点久，等待一下）

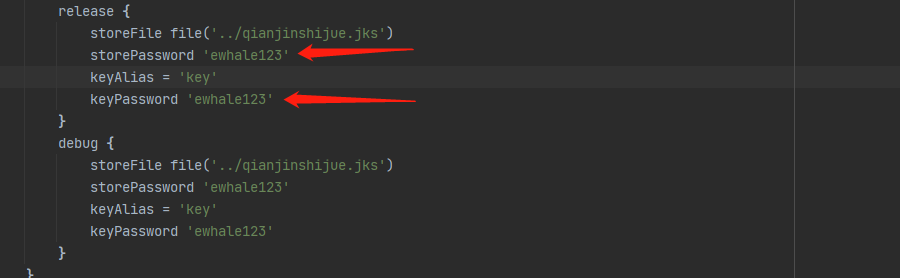




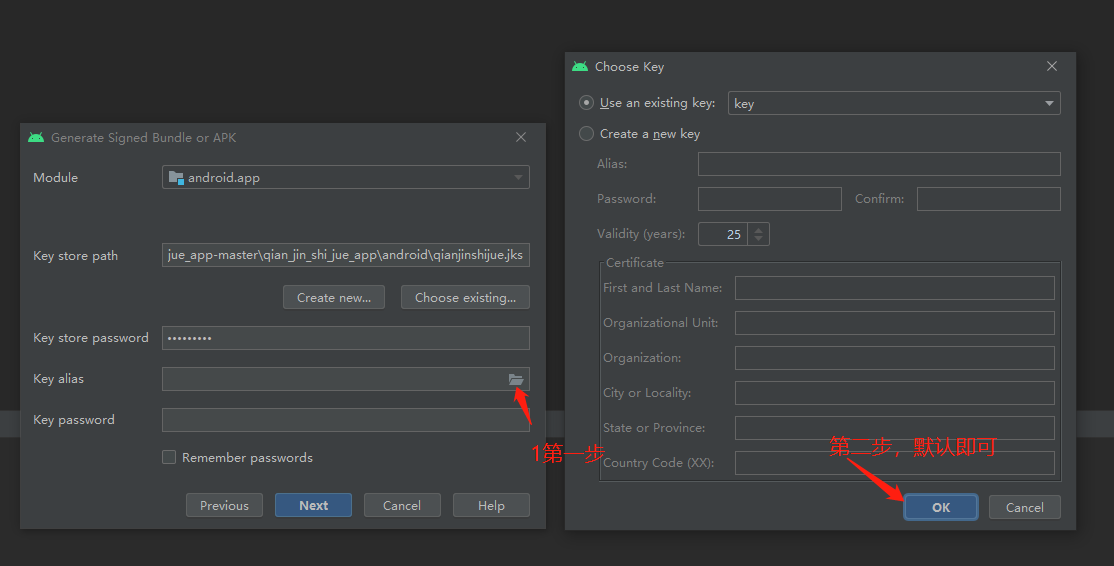
点击下一步，选择jks文件，这个项目已经有jks，点击Choose existing，选择已有的jks，并选中jks。



下面的两个密码就是在build.grald中写好的，默认是相同的：

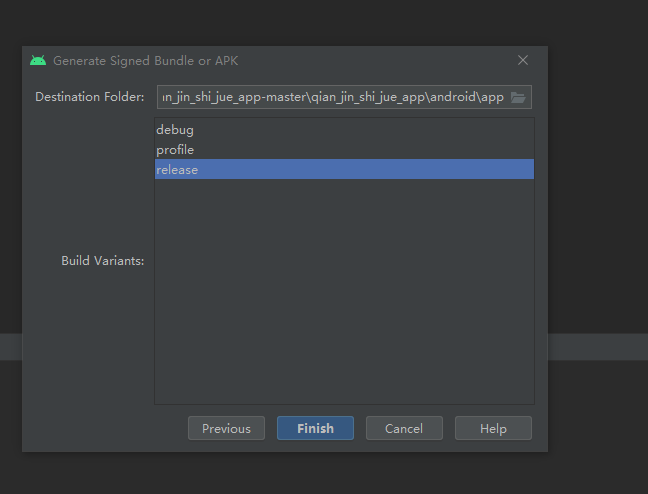


Key alias：点击右边的文件，默认即可。



选择release文件夹，点击finish即可。

注意：新版的android studio发布release的时候，如下图，不会有两个选项，如果没看到两个选项就不用管；旧版的会有两个选项，v1,v2签名方式，都选上。



1. **ios端**：

ios 端打包相较繁琐，网上可以搜索相关内容，xcode flutter 打包流程，这里只展现主要的操作，打包过程中可能会与xcode版本不同而出现其他情况。

前提：Mac OS上已配置好flutter，以及xcode

**第一步：**

双击p12文件，会解析出来证书文件，密码的话先试一下123456。然后去苹果开发者平台那边下载对应的profile文件，应该会有hoc（发布的），dev（调试用的）。

**第二步：**

同样是在android studio中打开项目(Mac OS 需要有Xcode)

以下是在命令行中操作：

刚开始拿到这个项目，先在命令行中检查下flutter版本（flutter --version），版本需要一致。

版本没问题再切换到ios 文件：cd ios

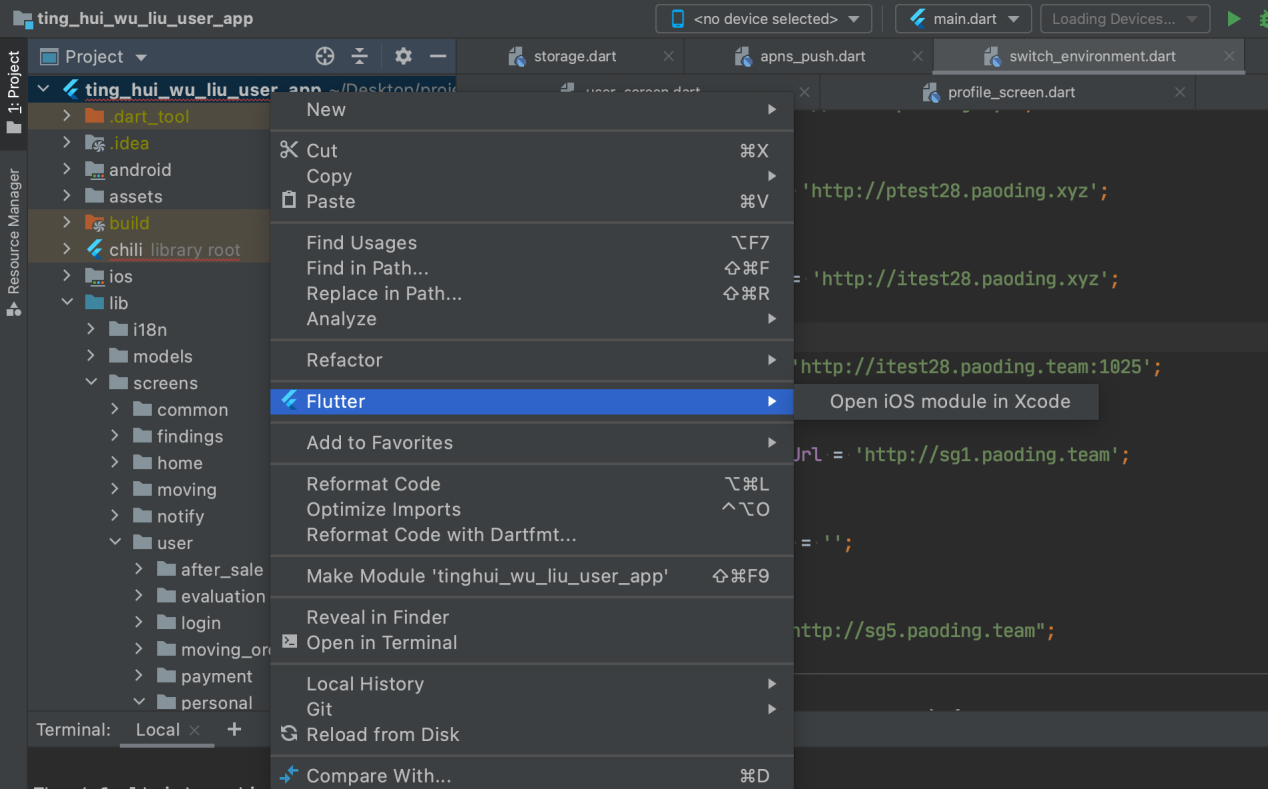
拉取插件，执行pod install

拉取完后，再切换到项目根目录： cd ../

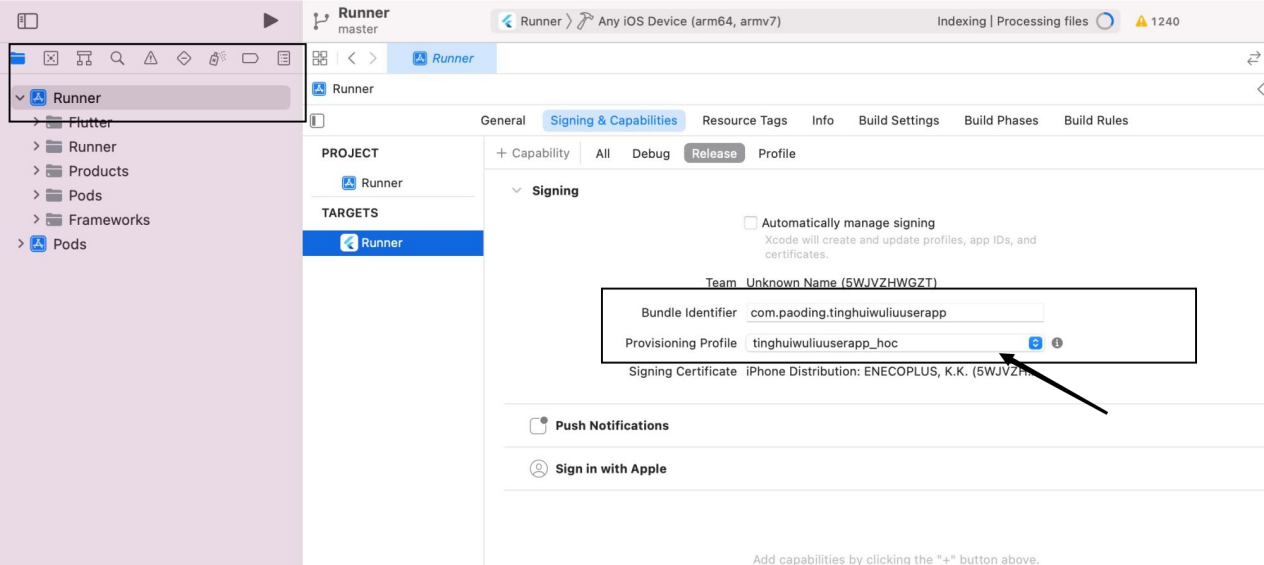
然后执行 flutter build ios

以下非命令行操作：

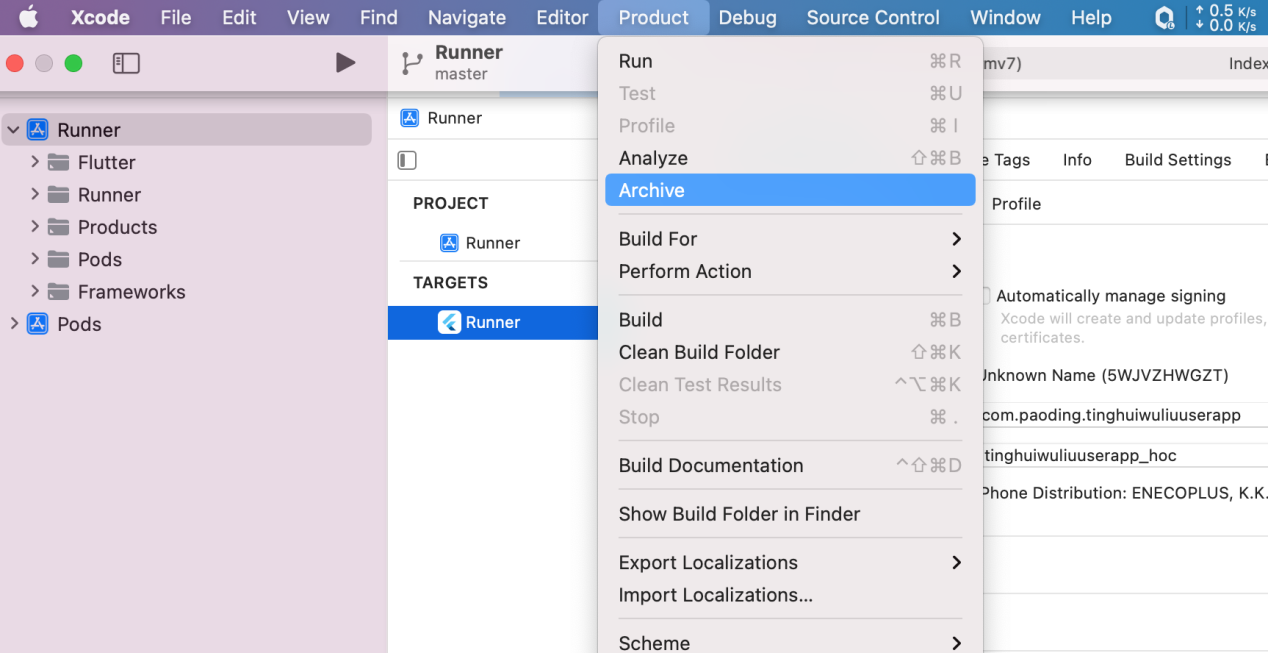
右击项目文件，跳转到Xcode



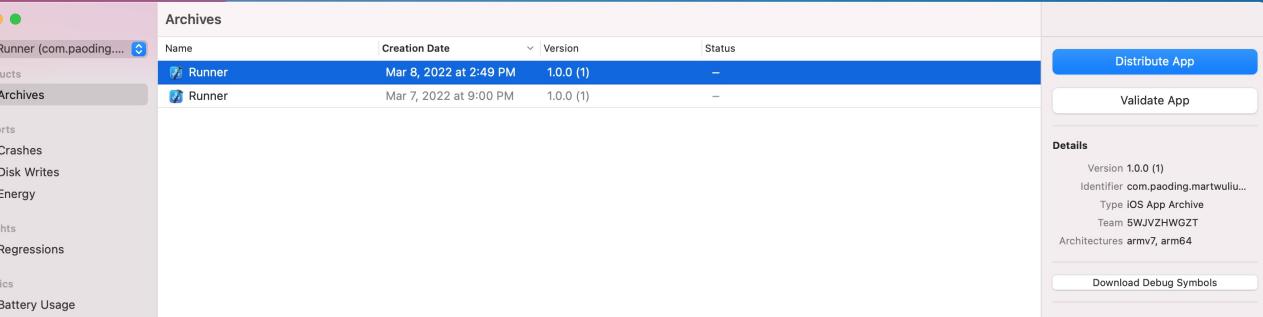
选择从开发者平台下载好的profile文件，如果是打包的话，就下载hoc的类型profile下载。



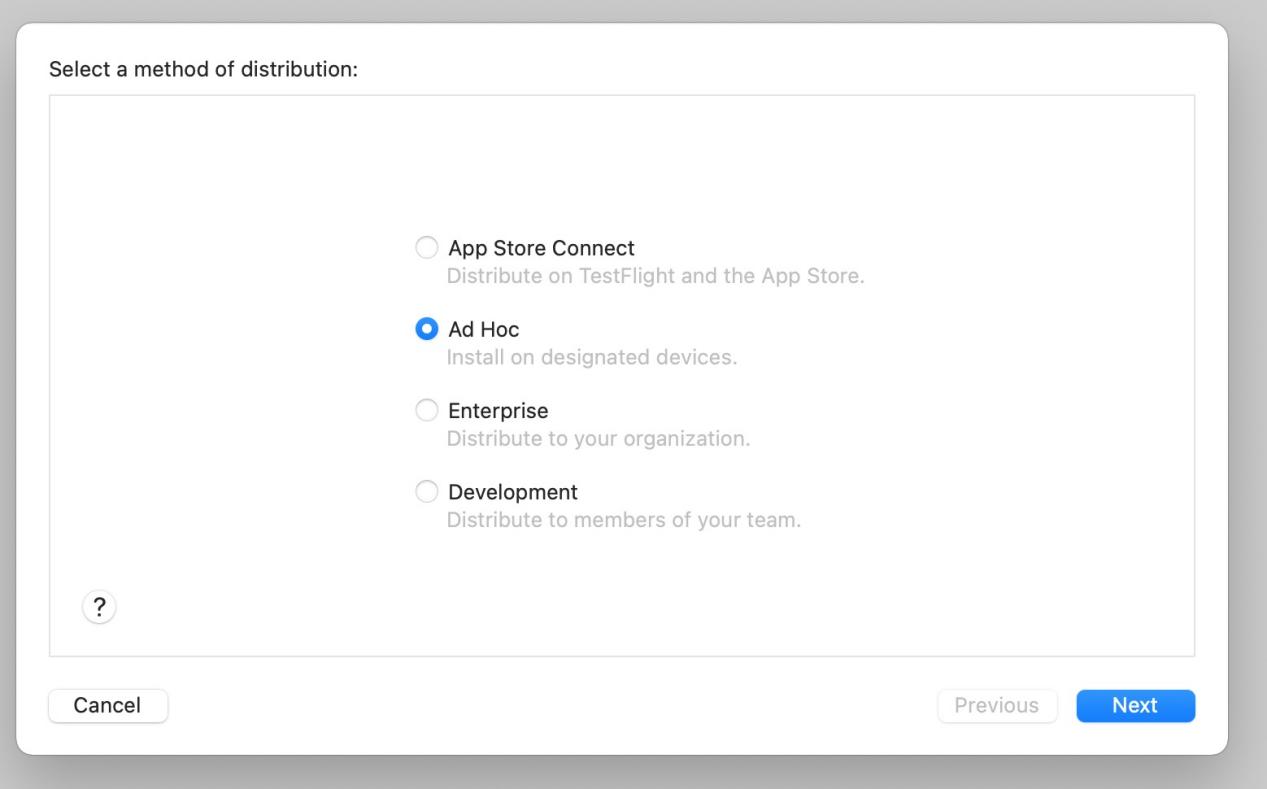
选择product -> archive,打包项目



打包好后，选择distribute app



如果只是普通的打内侧包就选择 hoc ,点击下一步后，下一步就选择默认的，然后再点击下一步。



进入到下图这个步骤，也是选择对应的证书，然后点击下一步，稍等一会儿，就可以生成对应的ipa文件。

