#### 打包发布 Android 项目

##### 1. app 签名

###### 1.1 创建 keystore

如果您有现有 keystore，请跳至下一步。如果没有，请通过在运行以下命令来创建一个：

keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000

(我使用的语句:

keytool -genkey -v -keystore flutter\_wei.keystore -alias my-keystore-wei -keyalg RSA -keysize 2048 -validity 10000

)

其中：

1.keystore 参数后的值： my-release-key.keystore， 表示一会儿要生成的那个 签名文件 的名称；请先保存好这个名称，因为后面发布项目期间会用到它。

2. alias 参数后面的值：my-key-alias，后续发布项目期间也会用到，因此也需要保存好这个参数值；注意：这个参数的值可以根据自己的需要进行自定制。

3. 当运行这个命令的时候，需要输入一系列的参数，和相关的口令的密码，一定要保存好相关的密码，后期会用到。

4. 当生成了签名之后，这个签名，默认保存到了自己的用户目录下 C:\Users\用户名目录\my-release-key.keystore (就在自己的项目目录中!)

5. 将自己的签名证书 copy 到 android/app 目录下。

注意：保持 keystore 文件的私密性； 不要将它加入到公共源代码控制中（最好的方式，是将 keystore 文件添加到 .gitignore 忽略文件）。

注意: keytool 可能不在你的系统路径中。它是 Java JDK 的一部分，它是作为 Android Studio 的一部分安装的。有关具体路径，请百度。

###### 1.2 引用应用程序中的 keystore

创建一个名为 <app dir>/android/key.properties 的文件，其中包含对密钥库的引用：

storePassword=<password from previous step>

keyPassword=<password from previous step>

keyAlias=key

storeFile=<location of the key store file, e.g. /Users/<user name>/key.jks>

(

我的:

storePassword=123456

keyPassword=123456

keyAlias=my-keystore-wei

storeFile=C:\workSpace\_flutter\flutter\_wei.keystore

)

###### 1.3 在 gradle 中配置签名

通过编辑 <app dir>/android/app/build.gradle 文件为您的应用配置签名：

1、替换:

android {

为：

def keystorePropertiesFile = rootProject.file("key.properties")

def keystoreProperties = new Properties()

keystoreProperties.load(new FileInputStream(keystorePropertiesFile))

android {

2、替换:

buildTypes {

release {

// TODO: Add your own signing config for the release build.

// Signing with the debug keys for now, so `flutter run --release` works.

signingConfig signingConfigs.debug

}

}

为：

signingConfigs {

release {

keyAlias keystoreProperties['keyAlias']

keyPassword keystoreProperties['keyPassword']

storeFile file(keystoreProperties['storeFile'])

storePassword keystoreProperties['storePassword']

}

}

buildTypes {

release {

signingConfig signingConfigs.release

}

}

##### 2. 构建一个发布版（release）APK

本节介绍如何构建发布版（release）APK。如果您完成了前一节中的签名步骤，则会对APK进行签名。

使用命令行:

cd <app dir> (<app dir> 为您的工程目录).

运行 flutter build apk -t lib\main\_test.dart (flutter build 默认会包含 --release 选项).

打包好的发布APK位于 <app dir>/build/app/outputs/apk/app-release.apk。

##### 3. 在设备上安装发行版 APK

按照以下步骤在已连接的Android设备上安装上一步中构建的APK

使用命令行:

用USB您的Android设备连接到您的电脑

cd <app dir> .

运行 flutter install .