签名APP

构建release包需要用到keystore，如果你还没有keystore可以通过一下方式来创建：

创建keystore

Mac：

```
keytool -genkey -v -keystore ~/key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias key
```

Windows：

```
keytool -genkey -v -keystore c:\Users\USER_NAME\key.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias key
```

- -keystore：用户指定存储路径

然后按照提示进行输入，完成所有输入之后会在上述路径中创建一个key.jks文件。然后将该文件复制到Flutter项目下的android目录下。

配置keystore

在你的flutter应用/android目录下创建key.properties文件，然后添加：

```
storePassword=123456

keyPassword=123456

keyAlias=key
storeFile=../key.jks
```

注意：这里的密码需要和keystore配置的密码一致

配置打包签名

用AS打开你的flutter应用/android/app/build.gradle文件然后在android代码块上面添加：

```
def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}

android {

      ...
}
```

然后在android代码块中添加signingConfigs：

```
 signingConfigs {
     release {
         keyAlias keystoreProperties['keyAlias']
         keyPassword keystoreProperties['keyPassword']
         storeFile file(keystoreProperties['storeFile'])
         storePassword keystoreProperties['storePassword']
     }
 }
```

最后在buildTypes代码块中添加release配置：

```
 buildTypes {
     release {
         signingConfig signingConfigs.release
     }
 }
```

构建release包

构建全部架构的安装包

```
./gradlew assembleRelease
```

构建出来的Release包是包含所有ABI架构的。

打出的包将被放在trip_flutter/build/app/outputs/apk/release/app-release.apk路径下。

构建单一架构的安装包

```
cd <flutter应用的android目录>
flutter build apk --split-per-abi
```

- flutter build：命令默认会构建出release包
- --split-per-abi：表示构建单一架构

上传应用

安装包构建好之后发布到Android各大应用市场，无论是长传到那个应用市场首先需要注册该平台的开发者通过开发者认证后便可进行应用上传了。

开发者注册和应用上传各大平台都有详细的说明教程和问题，下面分享国内比较大的应用市场：

- 华为：https://consumer.huawei.com/cn/
- 小米：https://dev.mi.com/console/
- OPPO：https://open.oppomobile.com/
- vivo：https://dev.vivo.com.cn/home
- 百度：https://app.baidu.com/apps/

有需要的小伙伴可以按照官方的说明进行注册和上传应用。

FAQ

FAQ在不断更新，遇到打包问题不要怕，可以在课程电子书中查看这一节的FAQ。

启用代码压缩后无法打包

很多同学会发现当启用代码压缩(minifyEnabled true)后，在打release包时报错，无法打出release包，那么只需将targetSdkVersion 和 compileSdkVersion 升级到 28。关联问题 @https://github.com/flutter/flutter/issues/26860#issuecomment-469751224

安装release包运行crash报so包找不到

flutter build apk会构建出包含x86_64、arm64-v8a、armeabi-v7a架构的安装包，但如何项目中所依赖的某个库和Flutter所支持的架构不一致就会出现so找不到的crash，比如：XX三方库仅有armeabi-v7a的so，当APP被安装到支持arm64-v8a的手机上时，手机发现APP中包含arm64-v8a的目录，于是就向这个目录中查找XX三方库的so发现找不到就报错了。

解决方案是只会打armeabi-v7a架构的flutter so，所以在release包前需要添加如下配置：

```
ndk {
//            abiFilters "armeabi-v7a","arm64-v8a","x86_64","x86" //只打包flutter所支持的架构，flutter没有armeabi架构的so，加x86的原因是为了能够兼容模拟器
    abiFilters "armeabi-v7a" //release 时只打"armeabi-v7包
}
```

更多资料参考

- https://docs.flutter.dev/deployment/android#build-an-apk