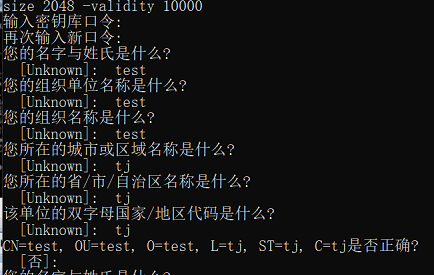
# 第一步：首先用keytool生成一个签名密钥

1. **在Windows上keytool命令放在JDK的bin目录中（比如默认安装路径C:\Java\jdk1.8.0\_91\bin里面，或者搜索keytool.exe所在文件夹）在命令行中输入以下命令:**

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

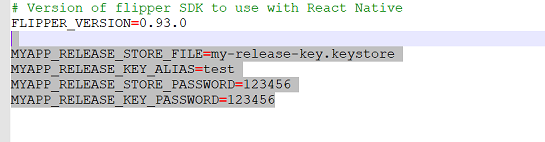
**(这条命令会要求你输入密钥库（keystore）和对应密钥的密码，然后设置一些发行相关的信息。全部填写后，会问你是否正确，按下y和回车即可。该密钥有效期为10000天。--alias参数后面的别名是你将来为应用签名时所需要用到的，所以记得记录这个别名。)**



# **第二步：**添加签名到项目的gradle配置文件，设置gradle变量

1. **最后它会在同级目录里生成一个my-release-key.keystore的密钥库文件,把刚才生成的my-release-key.keystore文件复制到你项目中的android/app文件夹里面**

1. **编辑~/.gradle/gradle.properties（全局配置，对所有项目有效）或是项目目录/android/gradle.properties（项目配置，只对所在项目有效）。如果没有gradle.properties文件你就自己创建一个，添加如下的代码（注意把其中的\*\*\*\*替换为相应密码）**



**打开gradle.properties文件 ，加入代码：**

**MYAPP\_RELEASE\_STORE\_FILE = my-release-key.keystore**

**MYAPP\_RELEASE\_KEY\_ALIAS = test**

**MYAPP\_RELEASE\_STORE\_PASSWORD = 123456**

**MYAPP\_RELEASE\_KEY\_PASSWORD = 123456**

1. **编辑你项目目录下的android/app/build.gradle，添加如下的签名配置：**

**signingConfigs {**

**release {**

**if (project.hasProperty('MYAPP\_RELEASE\_STORE\_FILE')) {**

**storeFile file(MYAPP\_RELEASE\_STORE\_FILE)**

**storePassword MYAPP\_RELEASE\_STORE\_PASSWORD**

**keyAlias MYAPP\_RELEASE\_KEY\_ALIAS**

**keyPassword MYAPP\_RELEASE\_KEY\_PASSWORD**

**}**

**}**

**}**

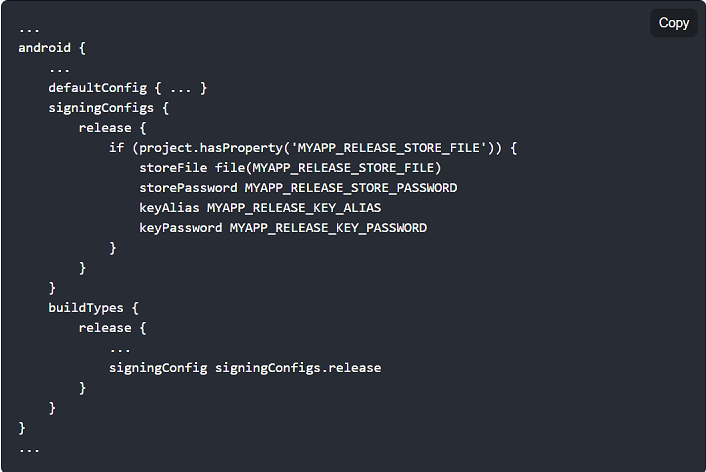
**buildTypes {**

**release {**

**signingConfig signingConfigs.release**

**}**

**}**



# **第三步：**生成发行APK包

1. **将命令行里的目录切换到你项目下的android目录里，然后执行命令：**

**gradlew assembleRelease开始打包编译**

**(若在windows的PowerShell环境中，请使用gradlew ./assembleRelease命令)**

1. **apk文件在哪里？在android\app\build\outputs\apk目录里有打包好的apk **任务完成~****

**参考：https://reactnative.cn/docs/signed-apk-android**