# Ionic Android 签名打包配置

## Android SDK Tools配置

所谓的Anrdoid打包，就是将ionic项目打包成一个可以安装在Android系统上的apk文件，打包的时候，使用的是cordova工具，不过在此之前，需要一些准备工作：配置JDK、Android SDk还有各个android版本包的下载，最重要的是需要成功的创建了一个Ionic项目，有关ionic项目创建的详细教程，请参考以下文章：  
[Ionic3 Start](https://www.jianshu.com/p/c29649b75709)

## 配置JDK

主要就是以下流程：下载 》 配置环境变量 》测试  
[JDK下载](https://link.jianshu.com?t=http:/www.oracle.com/technetwork/java/javase/downloads/index.html)  
配置环境变量：将JDK\_HOME添加到Path

命令行下测试

java

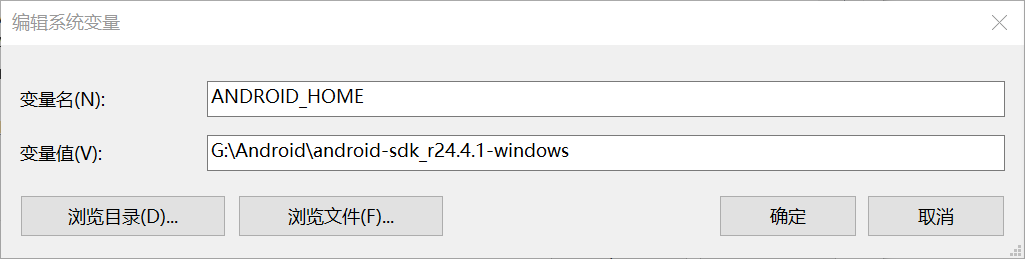
javac

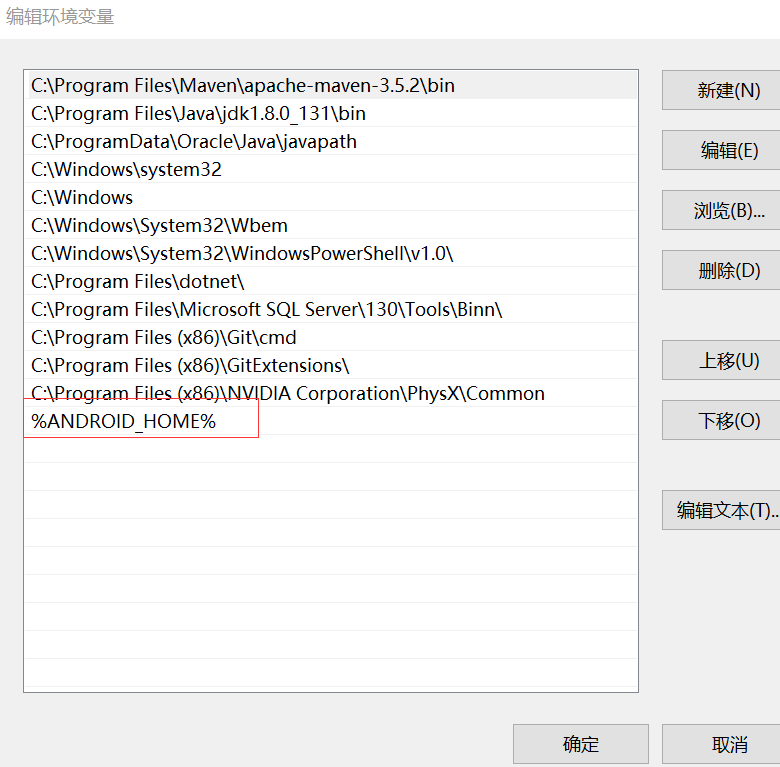
**配置Android SDK**

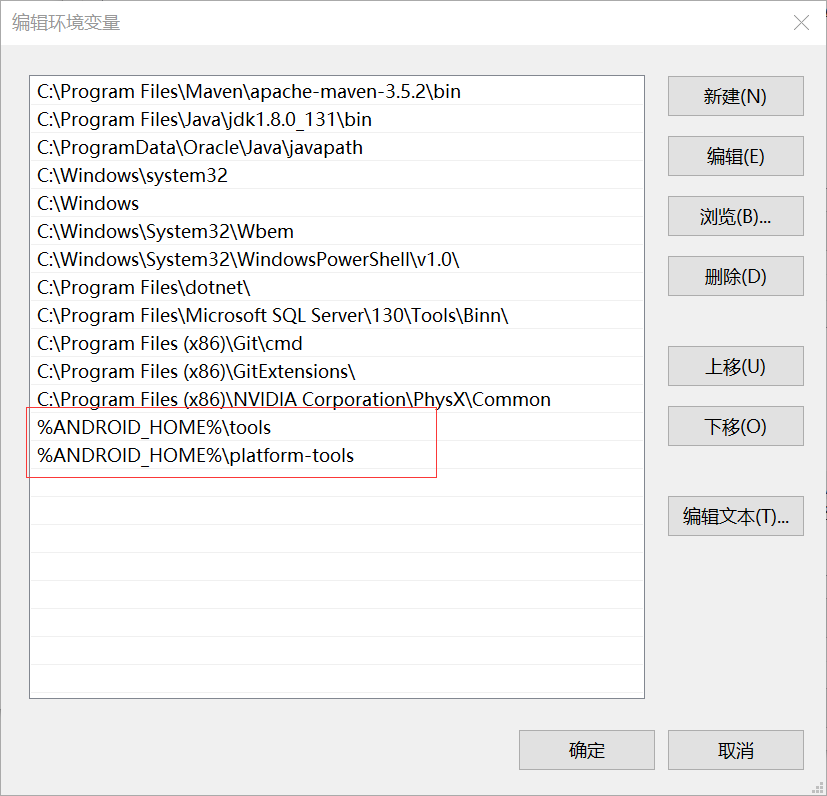


<http://www.androiddevtools.cn/>

### Android SDK 环境变量配置



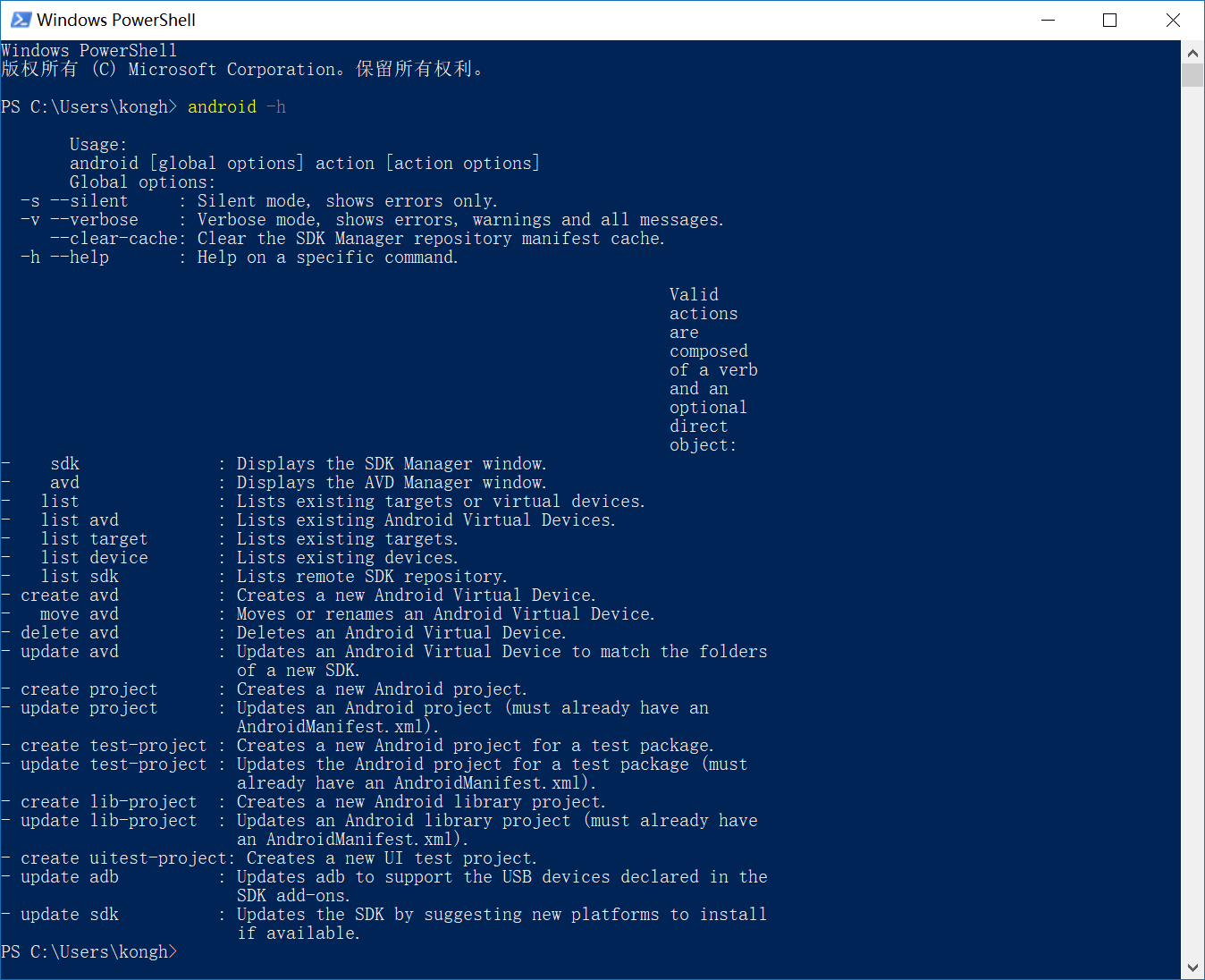




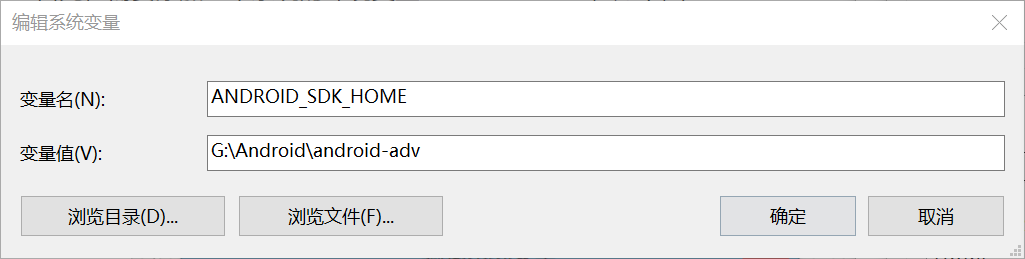
命令：

android sdk 测试环境变量是否配置成功

android -h 查看信息



### android studio AVD 默认位置环境变量



## android studio 安装

### Node安装

地址：<https://nodejs.org/zh-cn/>

代理设置：

npm set registry https://registry.npm.taobao.org # 注册模块镜像

npm set disturl https://npm.taobao.org/dist # node-gyp 编译依赖的 node 源码镜像

## 以下选择添加

npm set chromedriver\_cdnurl http://cdn.npm.taobao.org/dist/chromedriver # chromedriver 二进制包镜像

npm set operadriver\_cdnurl http://cdn.npm.taobao.org/dist/operadriver # operadriver 二进制包镜像

npm set phantomjs\_cdnurl http://cdn.npm.taobao.org/dist/phantomjs # phantomjs 二进制包镜像

npm set sass\_binary\_site http://cdn.npm.taobao.org/dist/node-sass # node-sass 二进制包镜像

npm set electron\_mirror http://cdn.npm.taobao.org/dist/electron/ # electron 二进制包镜像

npm cache clean # 清空缓存

## Ionic 安装

使用Node和NPM设置，让我们安装Ionic和Cordova CLI。

$ npm install -g ionic cordova

注意：这-g意味着这是全局安装，因此对于Window的您需要打开管理员命令提示符。对于Mac / Linux，您需要使用命令运行sudo。

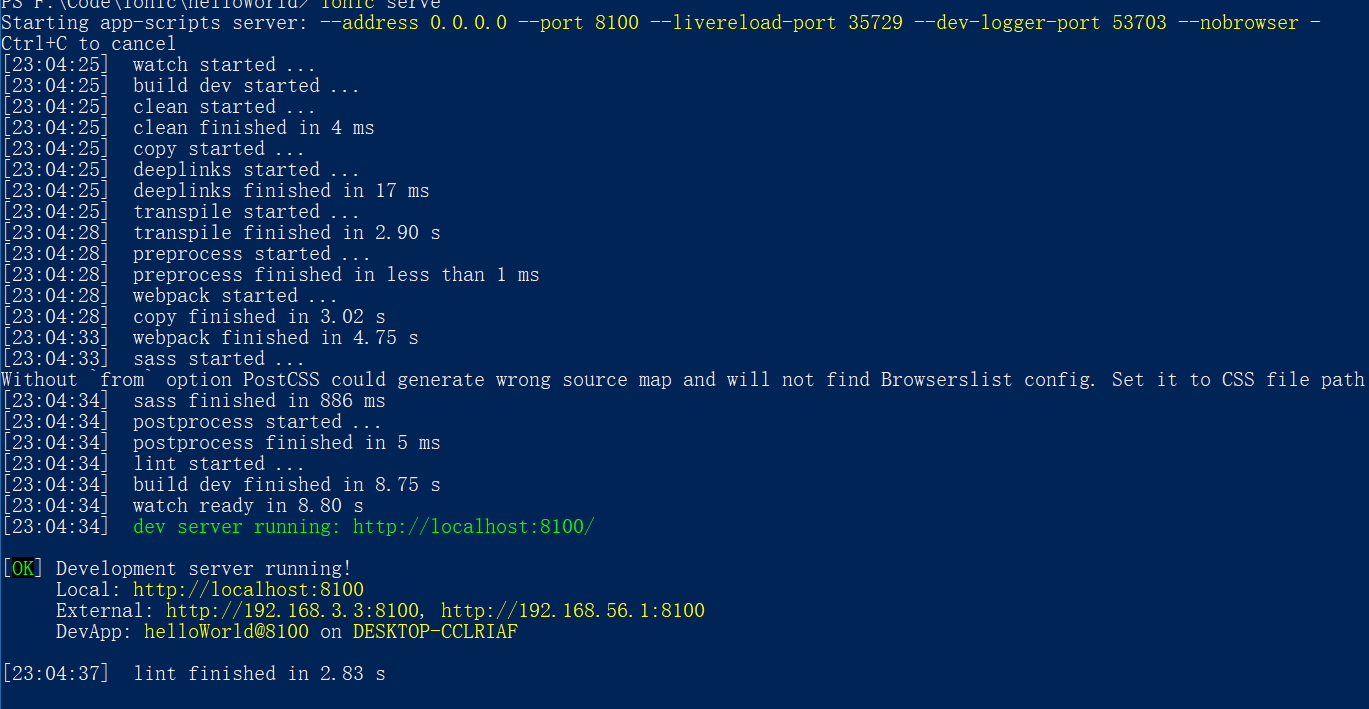
完成后，创建您的第一个Ionic应用程序：

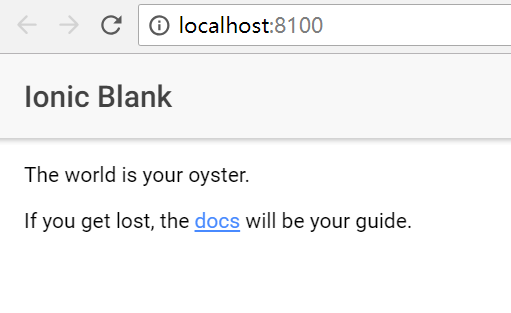
$ ionic start helloWorld blank

要运行您的应用程序，请cd进入创建的目录，然后运行 ionic serve命令以在浏览器中测试您的应用程序！

$ cd helloWorld

$ ionic serve





## Ionic 打包Android SDK

生产构建

要运行或构建您的应用以进行生产，请运行

ionic cordova run android --prod --release

# or

ionic cordova build android --prod --release

这会将您应用程序的代码缩小为Ionic的源代码，并从APK中删除任何调试功能。将应用程序部署到Google Play商店时通常会使用此功能。

签署Android APK

如果您想要在Google Play商店中发布您的应用，则必须签署APK文件。为此，您必须创建一个新的证书/密钥库。

让我们使用JDK附带的keytool命令生成您的私钥：

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

您将首先被提示为密钥库创建密码。然后，回答其他好的工具问题，当完成所有工作时，应该在当前目录中创建一个名为my-release-key.jks的文件。

注意：确保将此文件保存在安全的地方，如果您丢失了它，您将无法将更新提交到您的应用程序！

要签署未签名的APK，请运行也包含在JDK中的jarsigner工具：

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.jks android-release-unsigned.apk my-alias

这是APK的适当位置。最后，我们需要运行zip对齐工具来优化APK。zipalign工具可以在中找到/path/to/Android/sdk/build-tools/VERSION/zipalign。例如，在安装了Android Studio的OS X上，zipalign位于~/Library/Android/sdk/build-tools/VERSION/zipalign：

zipalign -v 4 android-release-unsigned.apk HelloWorld.apk

验证您的apk是否已签名运行apksigner。apksigner也可以在与zipalign工具相同的路径中找到：

apksigner verify HelloWorld.apk

现在我们有我们的最终版本二进制文件HelloWorld.apk，我们可以在Google Play商店上发布这个二进制文件，供全世界享用！