# How to build an Angular ionice project in a mobile devices

\*\*\*If the following issue appears,

Google Play 中已存在“com.ionicframework.starter”，请使用其他软件包名称。

================the follow the steps below:

You should make this change in the **config.xml** found at the root of your project. For example you might have something like this :

<widget id="com.ionicframework.exampleproject223738" version="0.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">

Change the id to whatever you wish your package to be and it will work for the entire project, Android and iOS.

\*\*\* if : git bash said: no cordova platform,

Then: try to run: $ionic serve

\*\*\*\*\* if: $cordova build, Git bash says: unquoted value..

Then: try put all folder under the c:Users/Mave/ directory

1.

Double click Genimotion

2.open Git bash, cd ngMileageApp

$ionic platform android

(or following $, type )$cordova platform add android

3, then $ ionic build

(or, following $ type) $ cordova build

Must have platforms and plugins forders within ngMileageApp, otherwise, you would always get a default package name and never be able to change the package name.

There are platforms and plugins forders appeared outside ngMileageApp folder. And CordovaApp\_debug.apk this file would appear in platforms/ant-built/android,

What is more (! The path of platforms folder: c:/users/mave/platforms)

3.5

So first, we need to generate a release build of our app, targeted at each platform we wish to deploy on. Before we deploy, we should take care to adjust plugins needed during development that should not be in production mode. For example, we probably don't want the debug console plugin enabled, so we should remove it before generating the release builds:

$ cordova plugin rm org.apache.cordova.console

4, $ cordova build --release android

CordovaApp-release-unsigned.apk in platforms/android/ant-built

5. To generate a release build for Android, we first need to make a small change to the AndroidManifest.xml file found in platforms/android. Edit the file and change the line:

<application android:debuggable="true" android:hardwareAccelerated="true" android:icon="@drawable/icon" android:label="@string/app\_name">

and change android:debuggable to "false":

<application android:debuggable="false" android:hardwareAccelerated="true" android:icon="@drawable/icon" android:label="@string/app\_name">

Let's generate our private key using the keytool command that comes with the JDK. If this tool isn't found, refer to the [installation guide](http://ionicframework.com/docs/guide/installation.html):

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

You'll first be prompted to create a password for the keystore. Then, answer the rest of the nice tools's questions and when it's all done, you should have a file called my-release-key.keystore created in the current directory.---c:users/mave/ngMileageApp

**Note**: Make sure to save this file somewhere safe, if you lose it you won't be able to submit updates to your app!

To sign the unsigned APK, run the jarsigner tool which is also included in the JDK: before run the jarsigner, must put: CordovaApp-release-unsigned.apk & my-release-key.keystore in the same direction

ngMileageApp folder (current folder)

$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore CordovaApp-release-unsigned.apk alias\_name

Note: after the step above, no new file appeared.

Finally, we need to run the zip align tool to optimize the APK:

$ zipalign -v 4 CordovaApp-release-unsigned.apk Mileage\_Rat20150311.apk

Now we have our final release binary called Mileage\_Rat20150310.apk and we can release this on the Google Play Store for all the world to enjoy!

========================

\*\*\*Only way to re-choose the language for your app in google play is: delete it & reload it