<http://ionicframework.com/docs/guide/publishing.html>

Chapter 6: Publishing your app

Now that we have a working app, we are ready to push it live to the world! Since [Drifty](http://drifty.com/), the creators of Ionic, already submitted the Todo app from this guide to the app store, chances are you'll want to follow this chapter with a new app that you make on your own.

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

Android Publishing

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">

Now we can tell cordova to generate our release build:

$ cordova build --release android

Then, we can find our *unsigned* APK file in platforms/android/bin. In our example, the file wasplatforms/android/bin/HelloWorld-release-unsigned.apk. Now, we need to sign the unsigned APK and run an alignment utility on it to optimize it and prepare it for the app store. If you already have a signing key, skip these steps and use that one instead.

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.

**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:

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

This signs the apk in place. Finally, we need to run the zip align tool to optimize the APK:

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

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

*(There are a few other ways to sign APKs. Refer to the official*[*Android App Signing*](http://developer.android.com/tools/publishing/app-signing.html)*documentation for more information.)*

**http://stackoverflow.com/questions/10396185/jarsigner-is-unable-to-open-jar-file-nooyawk-apk-nooyawk-is-app-name**

[**jarsigner is unable to open jar file: NooYawk.apk .(NooYawk is App name)**](http://stackoverflow.com/questions/10396185/jarsigner-is-unable-to-open-jar-file-nooyawk-apk-nooyawk-is-app-name)

**2 Answers**

[active](http://stackoverflow.com/questions/10396185/jarsigner-is-unable-to-open-jar-file-nooyawk-apk-nooyawk-is-app-name?answertab=active#tab-top)[oldest](http://stackoverflow.com/questions/10396185/jarsigner-is-unable-to-open-jar-file-nooyawk-apk-nooyawk-is-app-name?answertab=oldest#tab-top)[votes](http://stackoverflow.com/questions/10396185/jarsigner-is-unable-to-open-jar-file-nooyawk-apk-nooyawk-is-app-name?answertab=votes#tab-top)

|  |  |  |  |
| --- | --- | --- | --- |
| up vote4down vote | Unless NooYawk.apk is in your current working directory, you need to supply a full path to the APK file.   |  |  | | --- | --- | | [share](http://stackoverflow.com/a/10397125)|[improve this answer](http://stackoverflow.com/posts/10397125/edit) | answered May 1 '12 at 11:38  [[https://www.gravatar.com/avatar/5ff53c2b4f6cc66ee989bb9d9effe119?s=32&d=identicon&r=PG](http://stackoverflow.com/users/115145/commonsware)](http://stackoverflow.com/users/115145/commonsware)  [CommonsWare](http://stackoverflow.com/users/115145/commonsware) **420k**38886959 | |

**http://stackoverflow.com/questions/24442213/cannot-find-zip-align-when-publishing-app**

[**cannot find zip-align when publishing app**](http://stackoverflow.com/questions/24442213/cannot-find-zip-align-when-publishing-app)

|  |  |  |  |
| --- | --- | --- | --- |
| up vote62down vote[favorite](http://stackoverflow.com/questions/24442213/cannot-find-zip-align-when-publishing-app)  **16** | cannot.find.zip.align=The zipalign tool was not found in the SDK.  Please update to the latest SDK and re-export your application  or run zipalign manually.  Aligning applications allows Android to use application resources  more efficiently.  This is the message I receive when I try to publish my app. I'm using the latest revision of Android SDK Tools (23 which was released today) and SDK Platform-tools (20 which was also released today). I get an APK out of it, but if I try to upload it to Google Play I get an error complaining that it is not zip aligned.  Running zipalign manually fixes the apk, but does anyone know what causes this and how to fix it?  [[http://i.stack.imgur.com/tKsDb.png](http://stackoverflow.com/questions/tagged/android)android](http://stackoverflow.com/questions/tagged/android)   |  |  | | --- | --- | | cordova build --release android | cordova build --release android | |

**Can not generate HelloWorld-release-unsigned.apk.**

**Should run the following first:**

**$ ionic platform android**

**$ionic build android**

**$ionic run android**

