Setting up ionic project in linux

1. Firstly need to install Node.js

commands:

sudo apt-get update

sudo apt-get install nodejs

2. Install cordova:

$ sudo npm install -g cordova

3. Install ionic framework

sudo npm install -g ionic

4. Check installed ionic version

ionic -v

5. Create Your Project

ionic start HelloWorld blank

6. Add respective platforms android / ios

ionic platform add ios

ionic platform add android

7. Build the android app

ionic build android

Note: Go to platforms -> android -> build -> outputs ->apk ->android-debug.apk

android-debug.apk is the apk build

Publishing Ionic app(android)

1. So first, we need to generate a release build of our app, 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 cordova-plugin-console

2. To generate a release build for Android, we can use the following cordova cli command:

cordova build --release android

This will generate a release build based on the settings in your config.xml. Your Ionic app will have preset default values in this file, but if you need to customize how your app is built

3. Next, we can find our unsigned APK file in platforms/android/build/outputs/apk. In our example, the file wasplatforms/android/build/outputs/apk/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

4. Let’s generate our private key using the keytool command that comes with the JDK

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

5. 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

6. This signs the apk in place. Finally, we need to run the zip align tool to optimize the APK. The zipalign tool can be found in/path/to/Android/sdk/build-tools/VERSION/zipalign. For example, on OS X with Android Studio installed, zipalign is in~/Library/Android/sdk/build-tools/VERSION/zipalign:

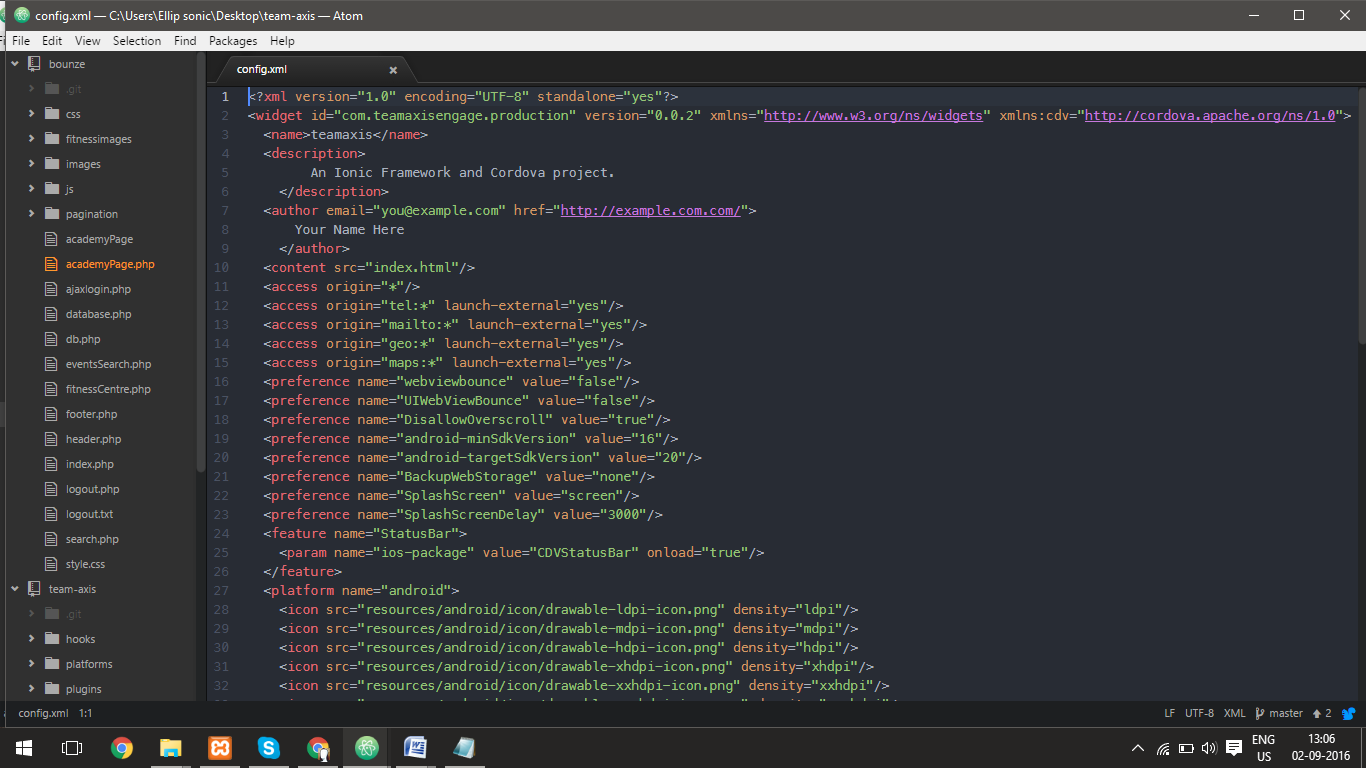
zipalign -v 4 HelloWorld-release-unsigned.apk HelloWorld.apk

we have our final release binary called HelloWorld.apk and we can release this on the Google Play Store

7. Finally upload the release apk to your created playstore account

8.updating app:

In order for the Google Play Store to accept updated APKs, you’ll need to edit the config.xml file to increment the version value, then rebuild the app for release.



**References:** [**http://ionicframework.com/docs/guide/publishing.html**](http://ionicframework.com/docs/guide/publishing.html)

[**https://youtu.be/57NteGoqFVQ**](https://youtu.be/57NteGoqFVQ)