



ionic

Building Hybrid Apps with AngularJS and Ionic

Submit to Apple App Store and Google Play Store

Apple – what you need:

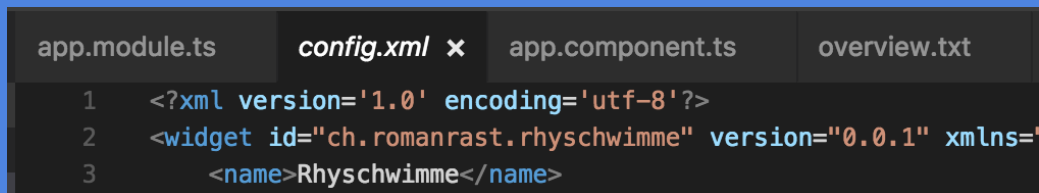
- Apple Developer Account (99\$/Year)
- Apple Developer Console
 - <https://developer.apple.com/account>
 - <https://itunesconnect.apple.com>
- Mac

Android – what you need:

- Android Dev Account (25\$ one time)
- Google Play Developer Console
 - <https://play.google.com/apps/publish>
- Windows or Mac

Submit to Apple App Store and Google Play Store

- Change id in your config.xml in your Ionic Project and version number



```
app.module.ts  config.xml x  app.component.ts  overview.txt
1  <?xml version='1.0' encoding='utf-8'?>
2  <widget id="ch.romanrast.rhyschwimme" version="0.0.1" xmlns=
3  <name>Rhyschwimme</name>
```

Android

Create .apk for Google Play Store:

- In the root of your App type following into your console:

```
cordova build --release android
```

You will find the apk in `platforms/android/build/outputs/apk/android-release-unsigned.apk`

- Create signing key – xxx.keystore

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

You will find the keystore in the root of your project

Android

Sign the apk:

- Into your console type:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore android-release-unsigned.apk my_alias
```

- Optimzie apk (less RAM usage)

```
zipalign -v 4 android-release-unsigned.apk APPNAME.apk
```

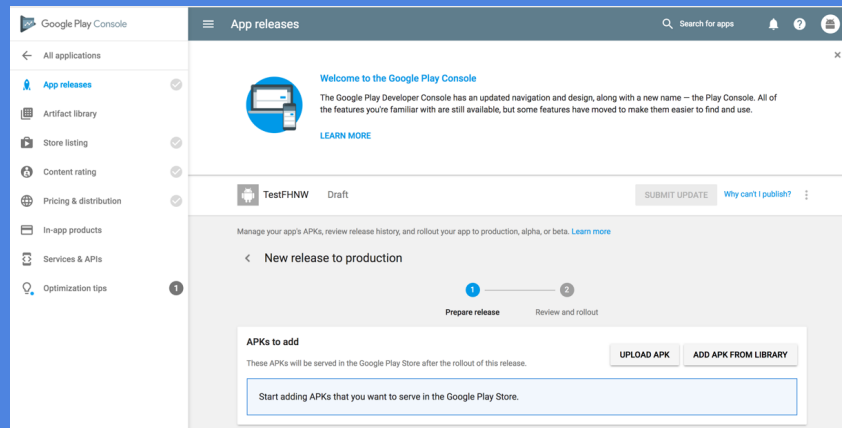
Android

You should have to following files now:

- app-release-unsigned.apk
- app-release-key.keystore
- **APPNAME.apk (FINAL apk)**

Upload your App now to
the Google Play Store:

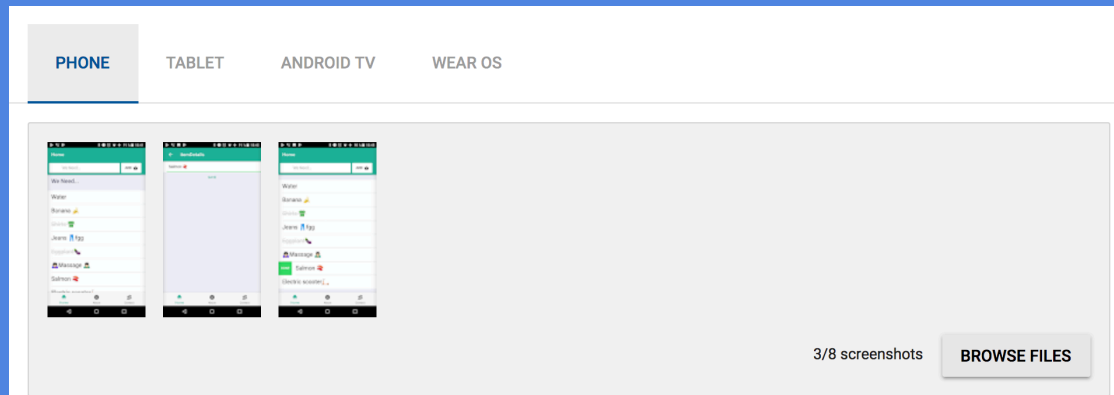
<https://play.google.com/apps/publish/>



Screenshots

Android:

- JPEG or 24-bit PNG (no alpha)
- Min length 320px, max length 3840px
- At least 2 screenshots
- If for tablet, then at least one 7-inch and one 10-inch screenshot



App Icon & Graphics

Hi-res icon *

Default – English (United States) – en-US

512 x 512

32-bit PNG (with alpha)

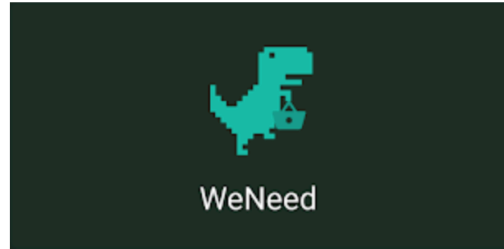


Feature Graphic *

Default – English (United States) – en-US

1024 w x 500 h

JPG or **24-bit PNG** (no alpha)



Promo Graphic

Default – English (United States) – en-US

180 w x 120 h

JPG or **24-bit PNG** (no alpha)



Testing


The screenshot shows the Google Play Console interface for the app 'WeNeed', which is in a 'Published' state. The left sidebar contains navigation links: 'All applications', 'Dashboard', 'Statistics', 'Android vitals', 'Development tools', 'Release management' (expanded), 'Release dashboard', 'App releases' (selected), 'Android Instant Apps', and 'Artifact library'. The main content area is titled 'App releases' and displays three testing tracks: Beta, Alpha, and Internal test. Each track has a 'MANAGE' link. The Beta and Alpha tracks show a message to 'Add APKs to beta/alpha to make your app available for open or closed beta/alpha testing.' The Internal test track shows a message: 'You have a release in internal test that hasn't been rolled out' with an 'EDIT RELEASE' button. The top right of the console shows the app name 'WeNeed', a status bar with notification, help, and profile icons, and a 'Published' label.

Google Play Console


App releases

WeNeed
Published

Beta [MANAGE BETA](#)

 Add APKs to beta to make your app available for open or closed beta testing.

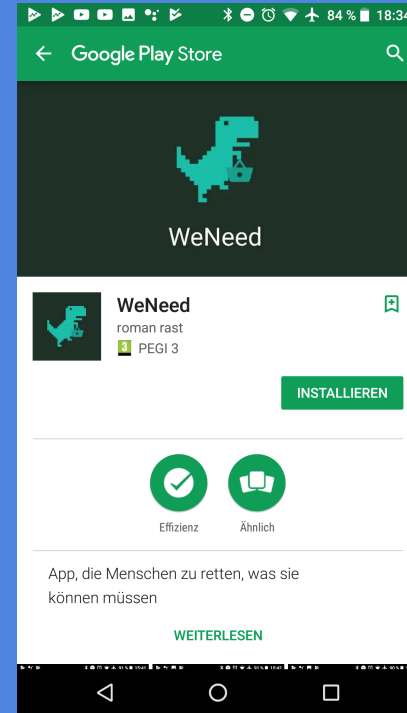
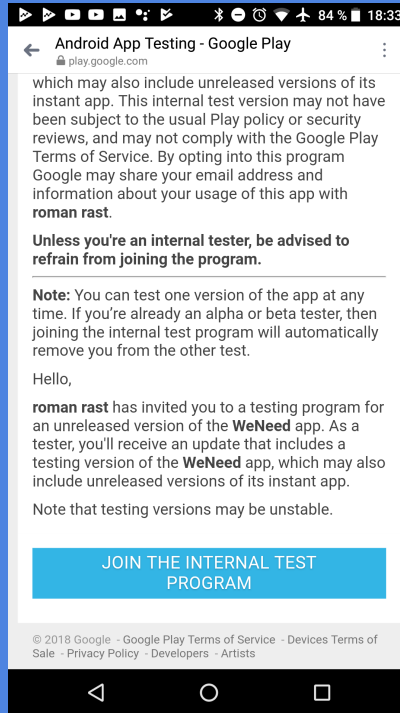
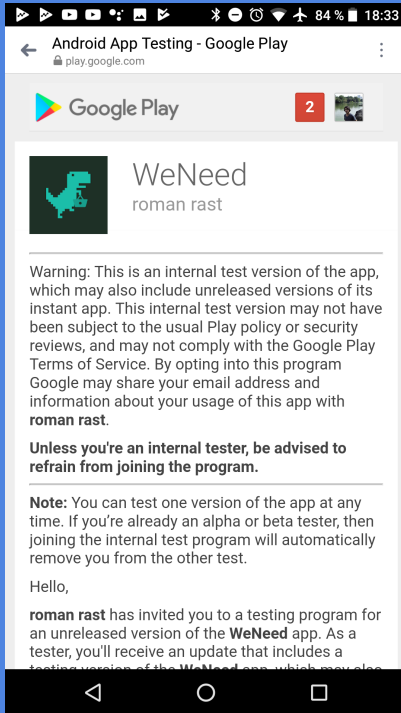
Alpha [MANAGE ALPHA](#)

 Add APKs to alpha to make your app available for open or closed alpha testing.

Internal test [MANAGE INTERNAL TEST](#)

You have a release in internal test that hasn't been rolled out [EDIT RELEASE](#)

Testing



iOS

- Create App ID on <https://developer.apple.com/account/>
- Request a Certificate from a Certificate Authority
 - xxx.certSigningRequest
- Get Certificate
 - Upload xxx. xxx.certSigningRequest on Apple Developer > Certificates, Identifiers & Profiles > Certificates/Production
 - Download xxx.cer file
- Generate .p12 file
 - Double click xxx.cer file > will show up in Keychain Access App under Login
 - Right click and choose > Export “iPhone Distribution...”
- Get provisioning profile
 - On Apple Developer > Certificates, Identifiers & Profiles > Certificates/Provisioning Profiles/Distribution select your App ID > select your certificate > enter a profile name > download

iOS

You should have 4 files (keep them save!)

- xxx.certSigningRequest
- xxx.cer
- xxx.p12
- xxx.mobileprovision

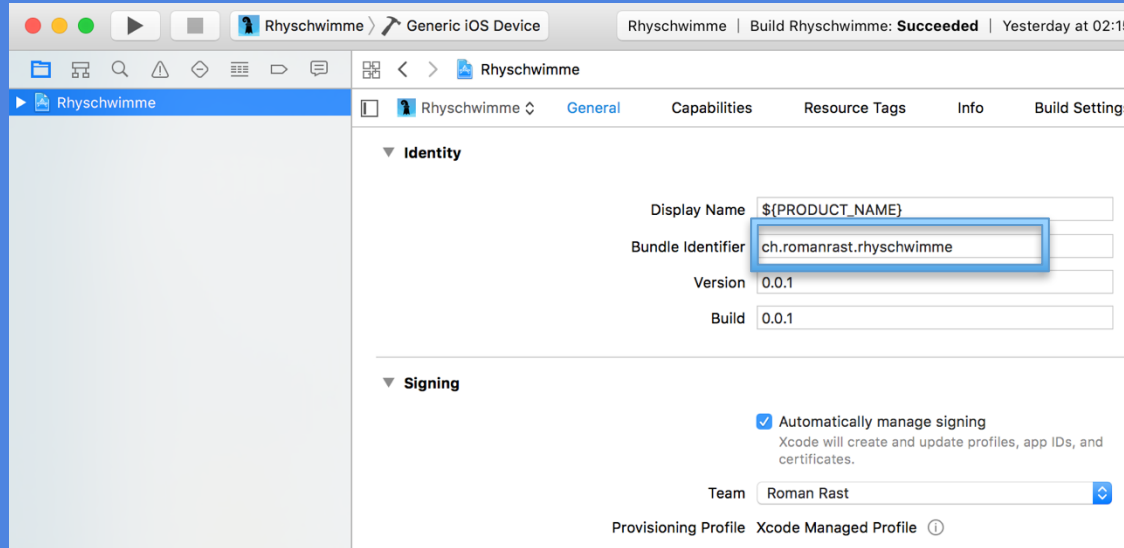
Prepare App fore release

In the root of your App type following into your console:

```
cordova build --release ios
```

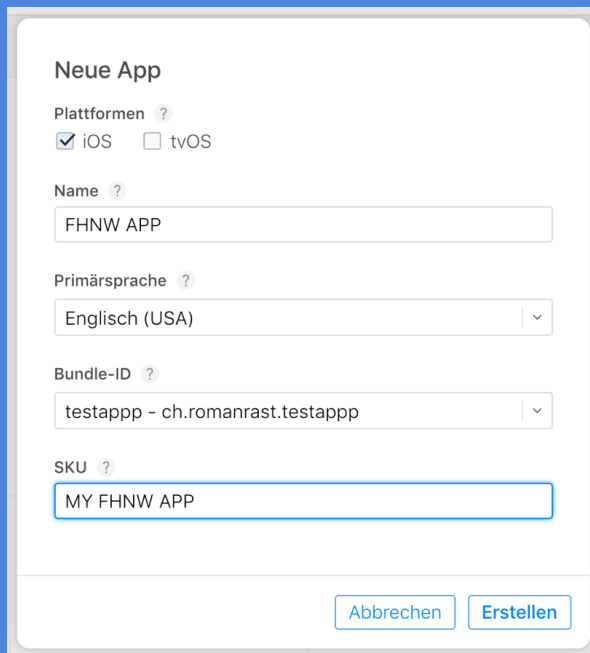
Prepare App fore release

Open xxx.xcodeproj file



Upload App to iTunes Connect

- Go to <https://itunesconnect.apple.com/login>
 - Create new App
 - Enter Bundle-ID
 - Setup Screenshots
 - Setup App infos



The screenshot shows the 'Neue App' (New App) form in iTunes Connect. The form is titled 'Neue App' and contains several fields for app information. The 'Plattformen' (Platforms) section has a checked box for 'iOS' and an unchecked box for 'tvOS'. The 'Name' field contains 'FHNW APP'. The 'Primärsprache' (Primary Language) dropdown is set to 'Englisch (USA)'. The 'Bundle-ID' dropdown is set to 'testappp - ch.romanrast.testappp'. The 'SKU' field contains 'MY FHNW APP'. At the bottom right, there are two buttons: 'Abbrechen' (Cancel) and 'Erstellen' (Create).

Neue App

Plattformen ?
☒ iOS ☐ tvOS

Name ?
FHNW APP

Primärsprache ?
Englisch (USA) ▼

Bundle-ID ?
testappp - ch.romanrast.testappp ▼

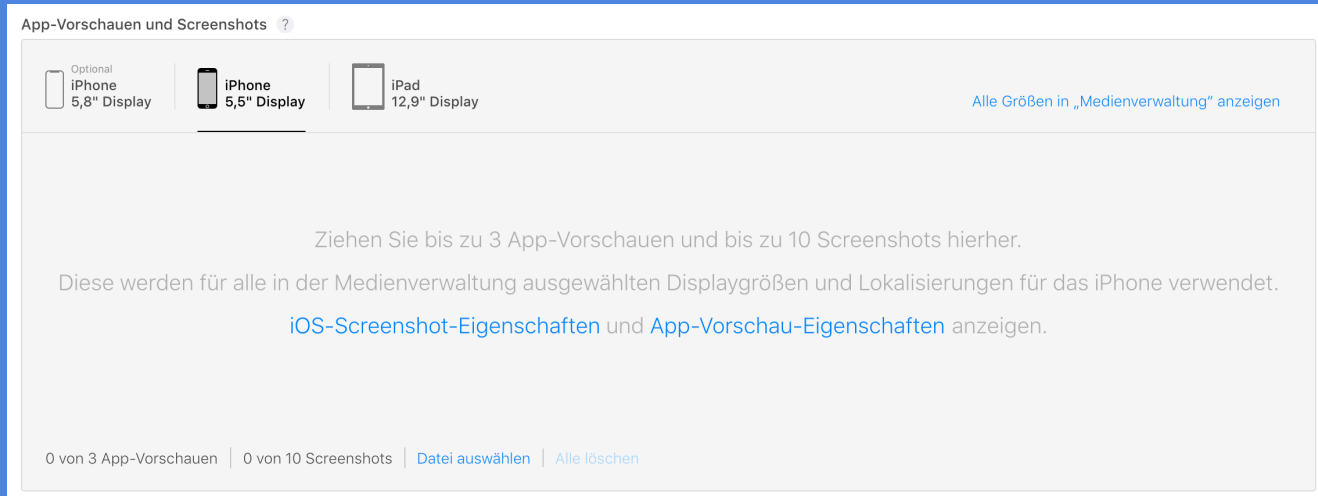
SKU ?
MY FHNW APP

Abbrechen Erstellen

Screenshots

iOS:

- 5.5-Inch Retina Display (1242 x 2208 pixels)
- 12.9-Inch Retina Display (2048 x 2732 pixels)



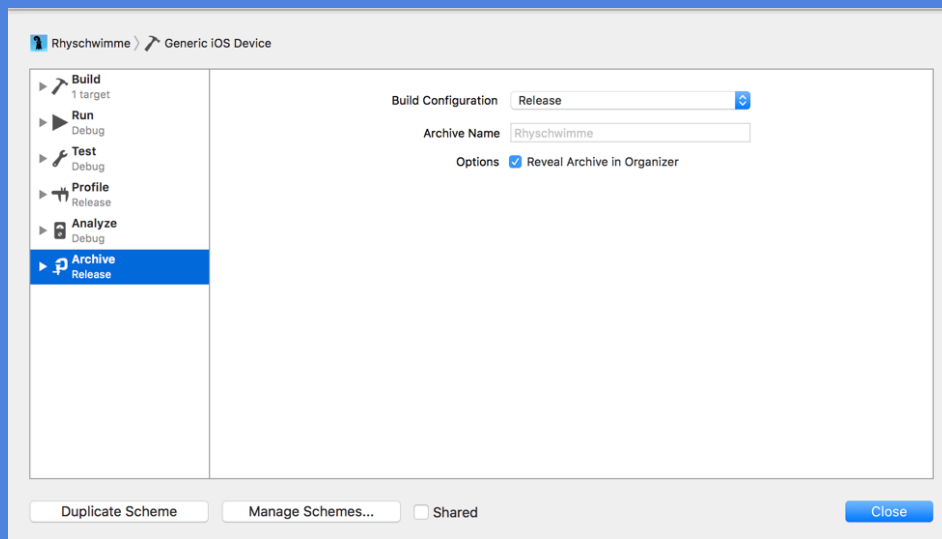
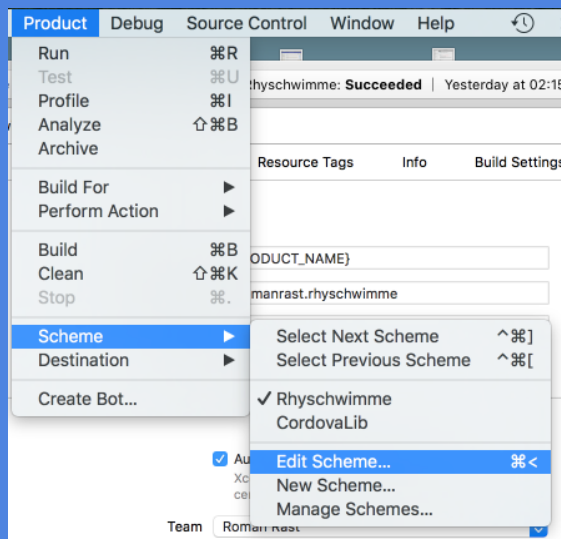
Screenshots

iOS:

- Run in simulator -> create screenshots by pressing cmd+s. This will save a screenshot to your desktop

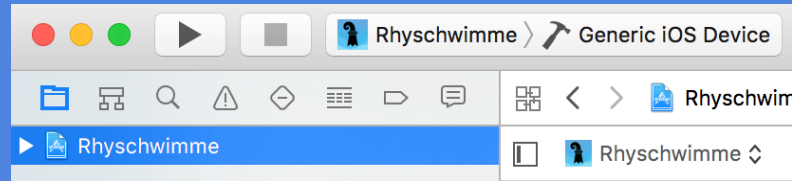
Upload App to iTunes Connect

- In xCode under the Application Menu:
Product > Scheme > Edit Scheme > Archive set the Build Configuration to Release



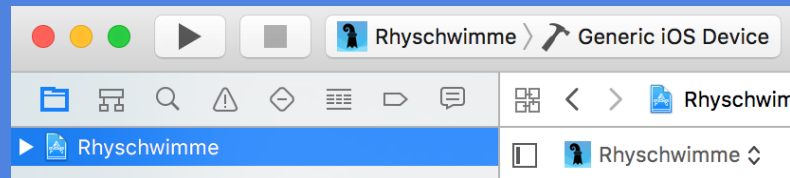
Upload App to iTunes Connect

- Set Device to Generic iOS Device

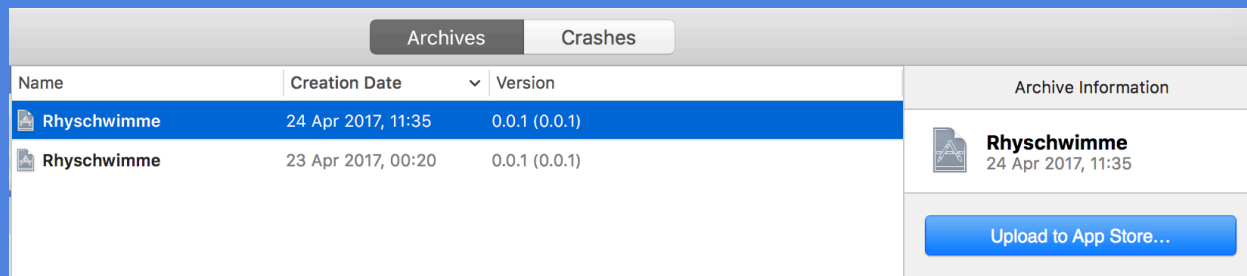
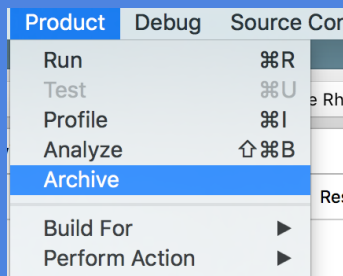


Upload App to iTunes Connect

- Set Device to Generic iOS Device



- Product > Archive > Choose your App > upload to App Store

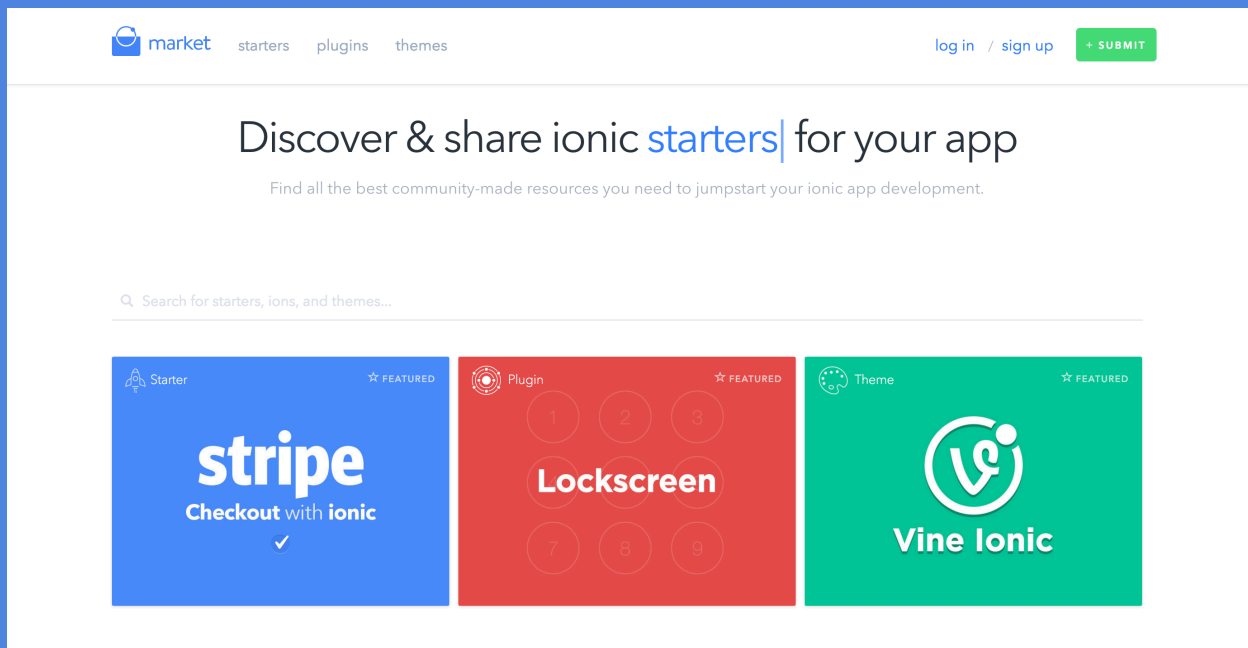


Market.ionic.io

Themes

Starters

Plugins



<http://market.ionic.io/>

More...

- Internationalization
- Push notification
- Deploy and testing
- Update your app
- Animations



ionic