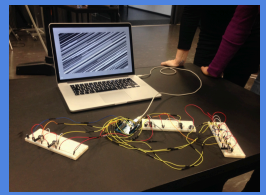
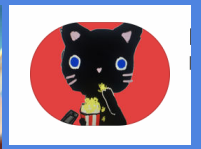
ionic

Building Hybrid Apps with AngularJS and Ionic

Roman Rast

- **What I do**
  - Mobile App Dev
  - UX / Interaction Design
  - Frontend
  - VR
  - Zühlke
- **Slides and stuff**
  - https://github.com/Zuehlke/fhnw-mobile-workshop.git
- **Flipboard** my Ionic magazin +360 followers
  - http://flip.it/q-lfz
- **Twitter**
  - @romanrast
- **Contact**
  - roman.rast@zuehlke.com



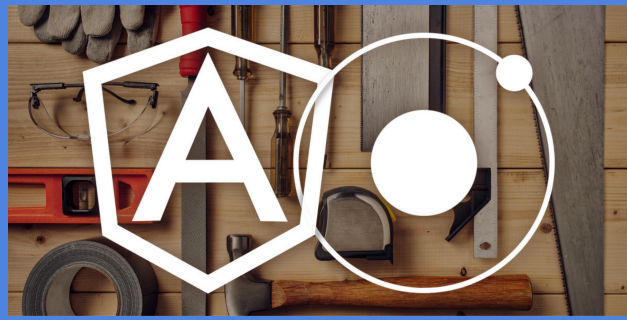


Hello Cat Stickers 2

Use Web Technologies

Ionic

- HTML5 Framework for hybrid mobile apps
- Build on AngularJS
- Uses Cordova
- First Ionic1 alpha in 2013, 1.0 Beta 2014, 1.0 2015
- Ionic 2.0 in January 2017
- High performance
- iOS > 7, Adnroid > 4.1
- Open source
- Many Frameworks and libraries built-in
  - AngualrJS, Cordova, Gulp, Sass, NodeJS, Bower, hammer.js
- Custom directives/components -> <ion-list>
- Drifty Co

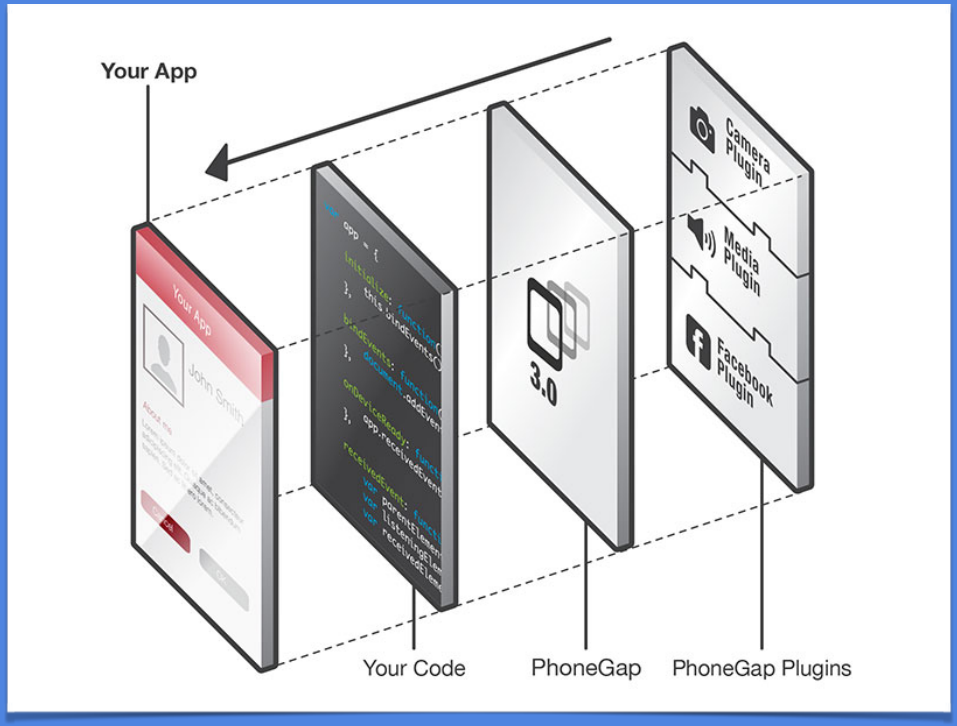


Architecture

- Your App
- Ionic
- AngularJS
- WebView (Cordova)
- Native SDK

APACHE CORDOVA™

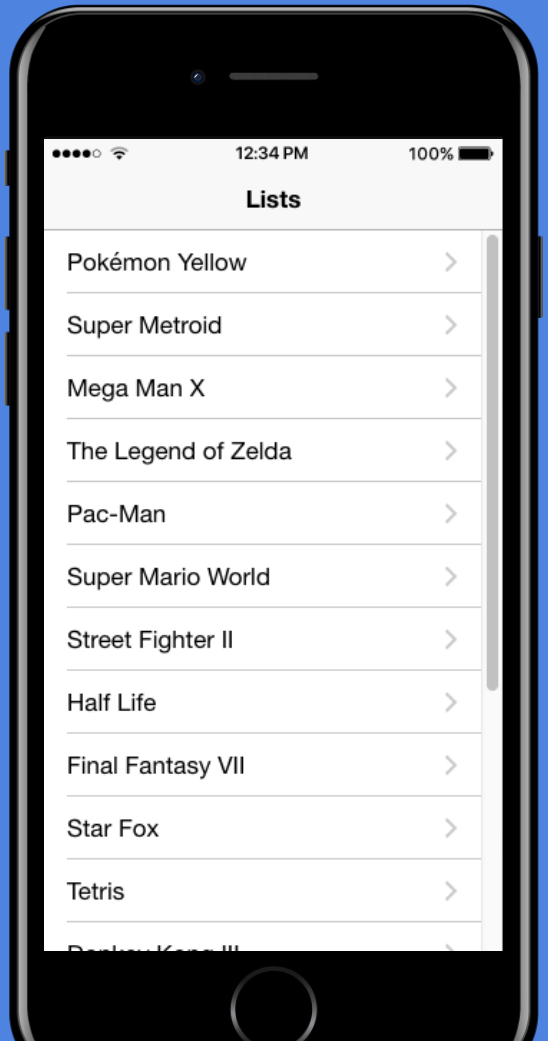
- Wraps app into a native app
- App uses WebView (browser of the phone without url bar and other stuff)
- Offers APIs to access camera, contacts, accelerometer finger print scanner etc.
- Develop a single code base for iOS and Android...
- PhoneGap/Cordova - Adobe


Your App
Your Code
PhoneGap
PhoneGap Plugins
Camera Plugin
Media Plugin
Facebook Plugin
3.0

Pro | Cons

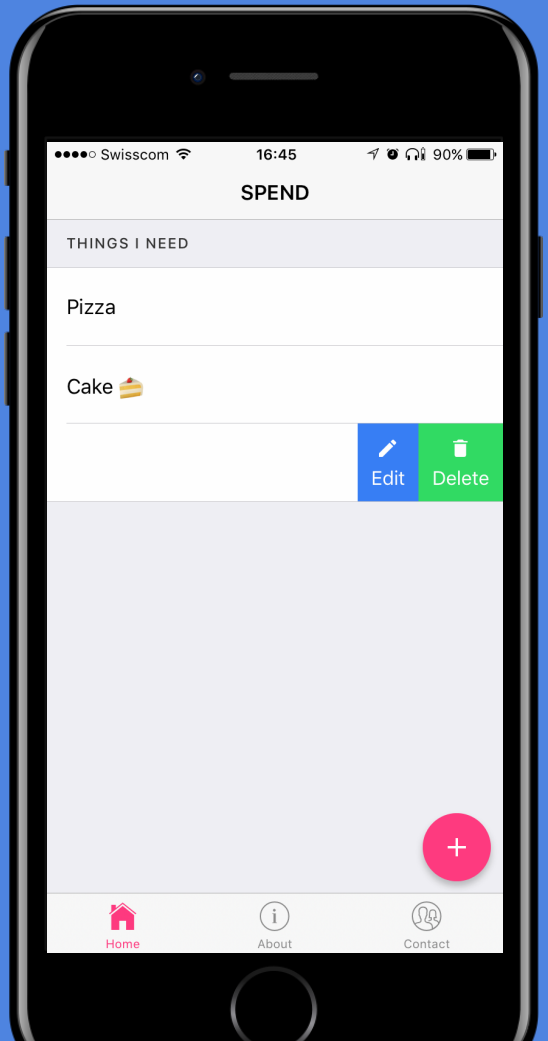
Pro	Cons
Open-source and free	Rendering complex UI can be slow
AngularJS, HTML, CSS	Not the best choice for graphic intense UI
SASS support	Android can be a hard to show same perfomance on mid-range phones
Offers large number of 3rd party resources, plugins, thems, starter apps (Ionic Native Plugins, ionic market, ionicons, ionic.io)	
Great forum	
One code base for iOS and Android (almost)	

Lists



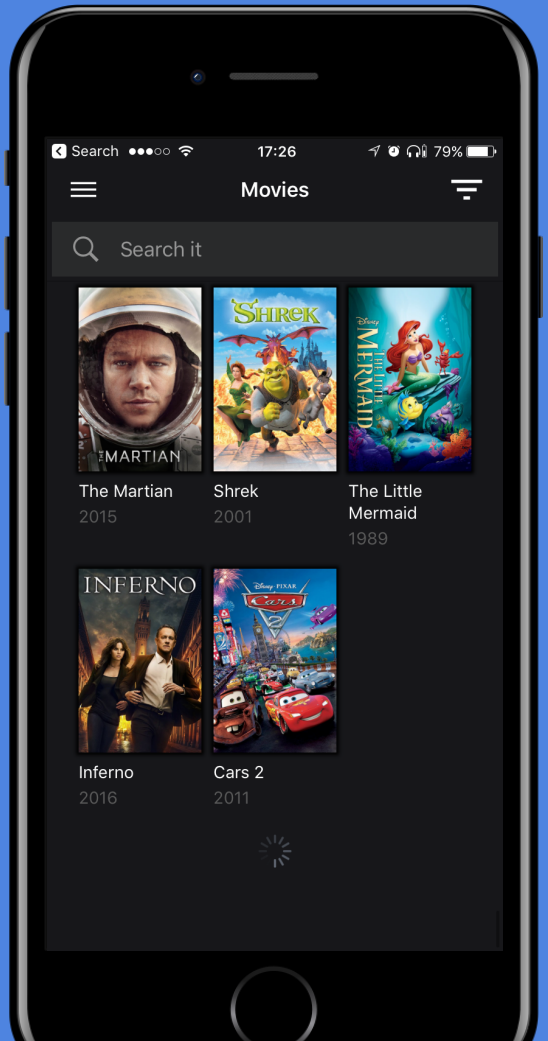

```
<ion-list>
  <button ion-item *ngFor="let item of items" (click)="itemSelected(item)">
    {{ item }}
  </button>
</ion-list>
```

Sliding List



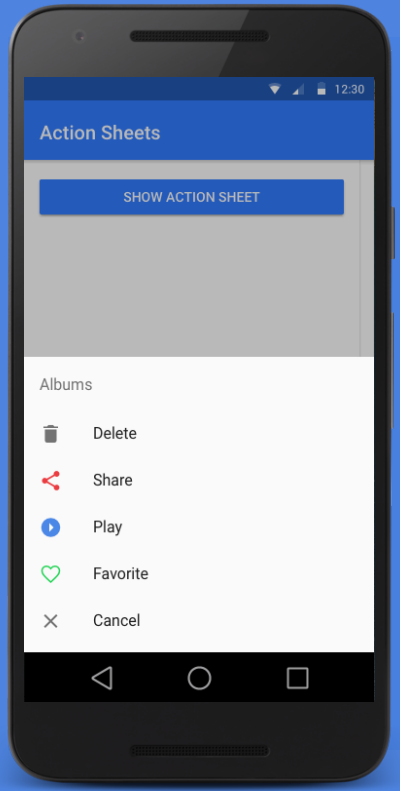

```
<ion-list>
   <ion-list-header>
       Things I need
   </ion-list-header>
   <ion-item-sliding *ngFor="let item of items | async">
       <ion-item (tap)="viewItem(item.$key, item.name)">
       {{item.name}}
       </ion-item>
       <ion-item-options side="right">
           <button ion-button color="primary" (click)="updateItem(item.$key, item.name)">
               <ion-icon name="md-create"></ion-icon>
               Edit
           </button>
           <button ion-button color="secondary" (click)="removeItem(item.$key)">
               <ion-icon name="md-trash"></ion-icon>
               Delete
           </button>
       </ion-item-options>
   </ion-item-sliding>
</ion-list>
```

Infinite Scroll




```
<ion-infinite-scroll (ionInfinite)="doInfinite($event)">
    <ion-infinite-scroll-content></ion-infinite-scroll-content>
</ion-infinite-scroll>
```

Action Sheet

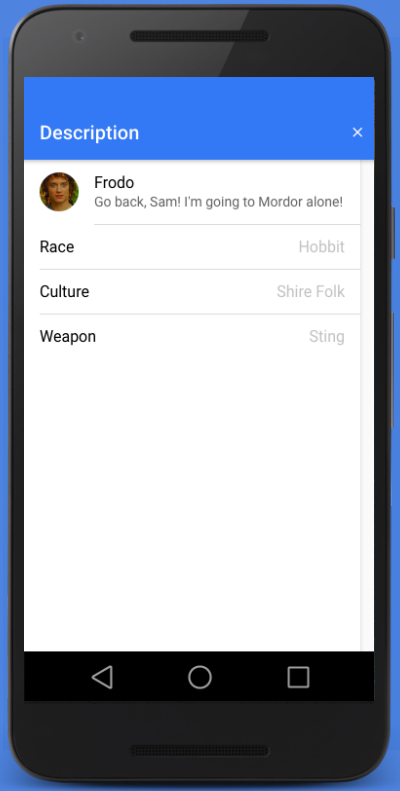


Action Sheet

- AngularJS Service
- Inject into controllers

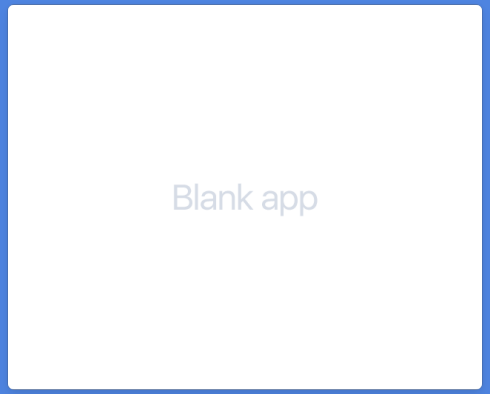
```
$ionicActionSheet.show({
  titleText: 'Action Sheet Example',
  buttons: [
    { text: 'Share' },
    { text: 'Move' },
  ],
  destructiveText: 'Delete',
  cancelText: 'Cancel',
  buttonClicked: function(index) {
    console.log('BUTTON CLICKED', index);
    return true;
  }
});
```

Modal

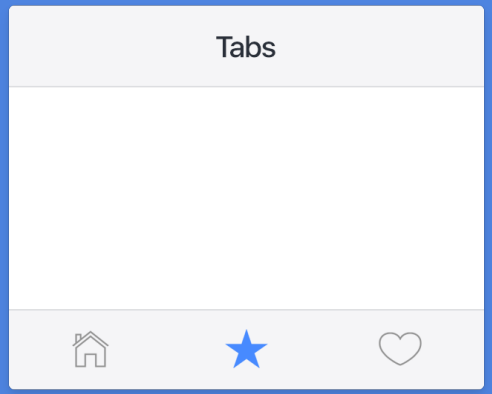



```
import { ModalController } from 'ionic-angular';
import { ModalPage } from './modal-page';

export class MyPage {
    constructor(public modalCtrl: ModalController) {
    }

    presentModal() {
        let modal = this.modalCtrl.create(ModalPage);
        modal.present();
    }
}
```

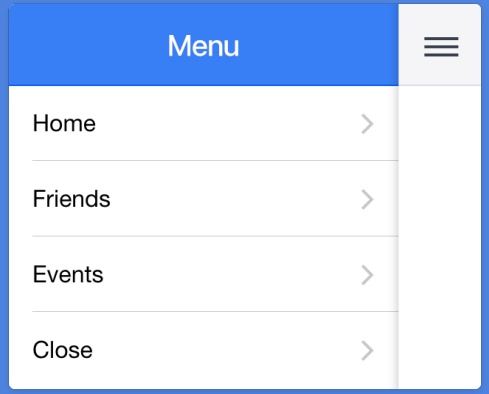
Templates



`ionic start myApp blank --v2`



`ionic start myApp tabs --v2`

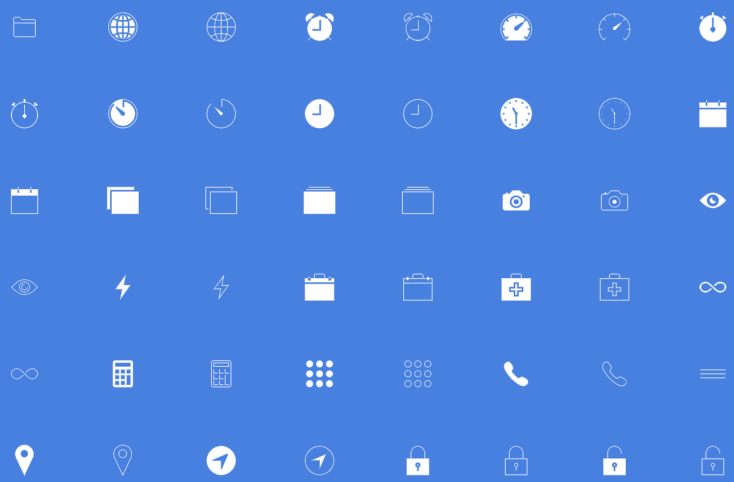


`ionic start myApp sidemenu --v2`

- Nested views
- Each tab has its own nav history

Ionicons

<ion-icon name="md-trash"></ion-icon>



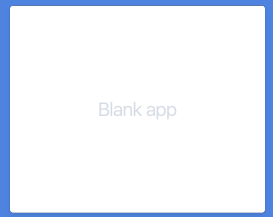
Over 700 MIT licensed font-icons included

ionicons.com

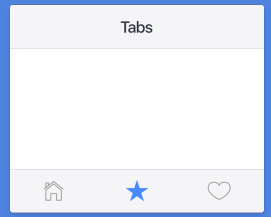
Installation

- Install Node.js (https://nodejs.org/)
- $ npm install -g cordova ionic
- $ sudo ionic start --v2 myApp tabs
- $ cd myApp
- (change access rights if necessary on mac)
- $ sudo ionic platform add ios | android
- $ sudo ionic build ios | android
- $ sudo ionic serve
- $ sudo ionic run ios | android (when deive is pluged in)
- $ sudo ionic build ios | android

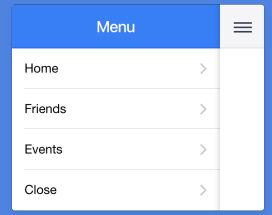
Keep xcode and iOS versions always up to date!



`ionic start myApp blank --v2`



`ionic start myApp tabs --v2`

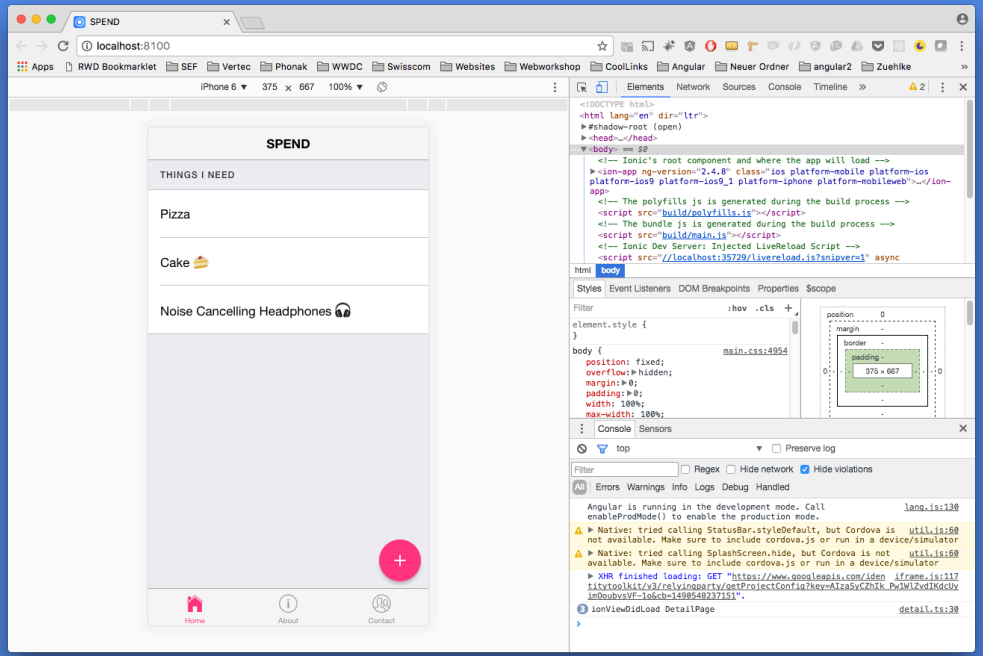


`ionic start myApp sidemenu --v2`

Debugging & Testing

- Use Google CHROME!
- Developer Tools

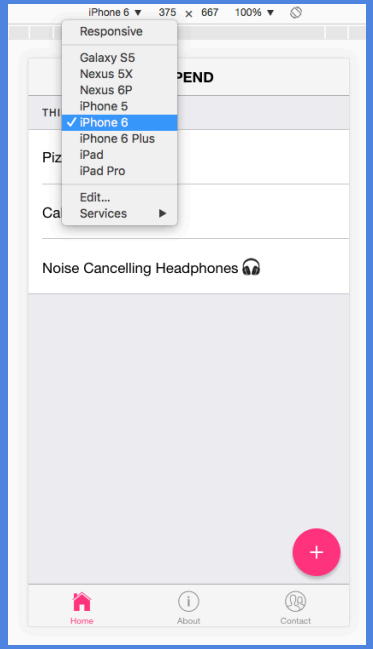
Chrome Developer Tools



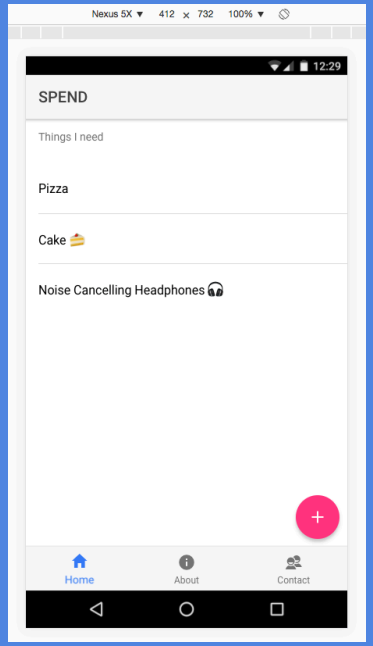
View > Developer > Developer Tools

Shortcut Mac alt+cmd+i

Chrome Developer Tools



iOS Design



Android Design

Plugins & Ionic Native

- Out of the box
  - cordova-plugin-splashscreen
  - cordova-plugin-statusbar
  - ionic-plugin-keyboard

- More on Ionic Native
    - http://ionicframework.com/docs/v2/native/
    - Camera, Barcode Scanner, Geolocation, Geofence, Gyroscope,...

CLI

$ cd myApp

Add iOS and Android platform for cordova

$ ionic platform add ios | $ ionic platform add android

Build Debug version

$ ionic build ios | $ionic build android

Build Prod version

$ ionic build ios --release

Run in Browser

$ ionic serve

$ ionic serve --lab

Run in emulator

$ ionic emulate ios

Run on real device if connected

$ ionic run ios

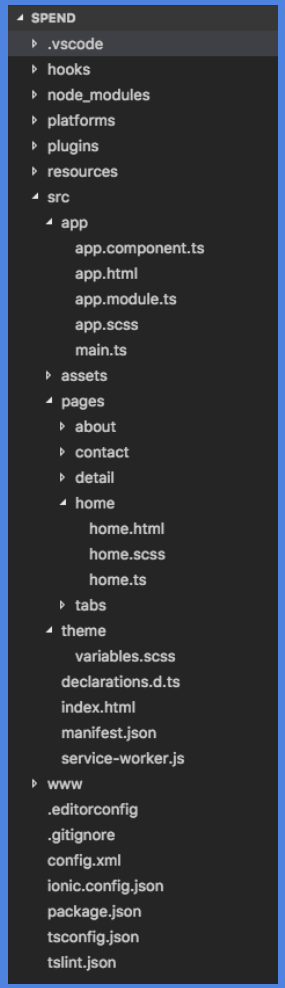
$ ionic run android

$ ionic package debug android (debug or release | android or ios)

App Structure

- Work in the **src** folder on the **root** of your app
- The code will be compiled and copied to www/ios and android
- For iOS go to platforms/ios/xxx.xcodeproj to run app on device or Simulator
- For Android plug in your Android Phone and type:
  - ionic run android

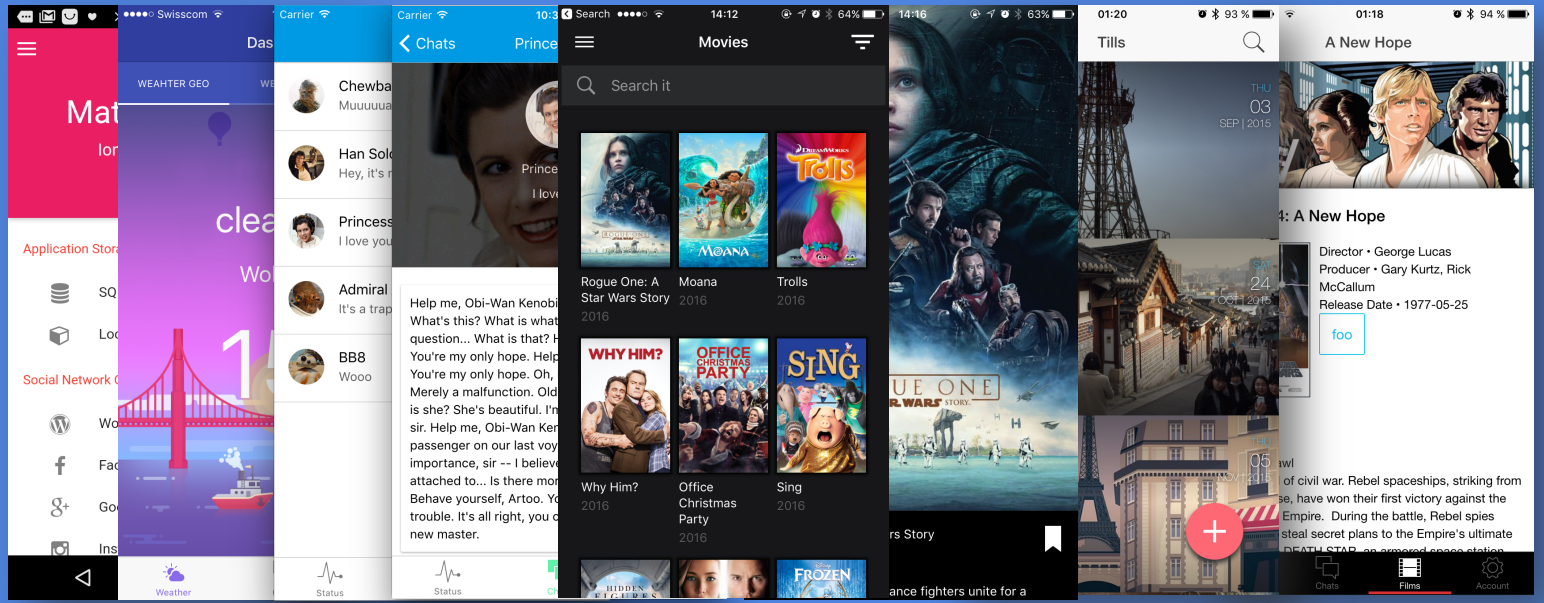
App Structure



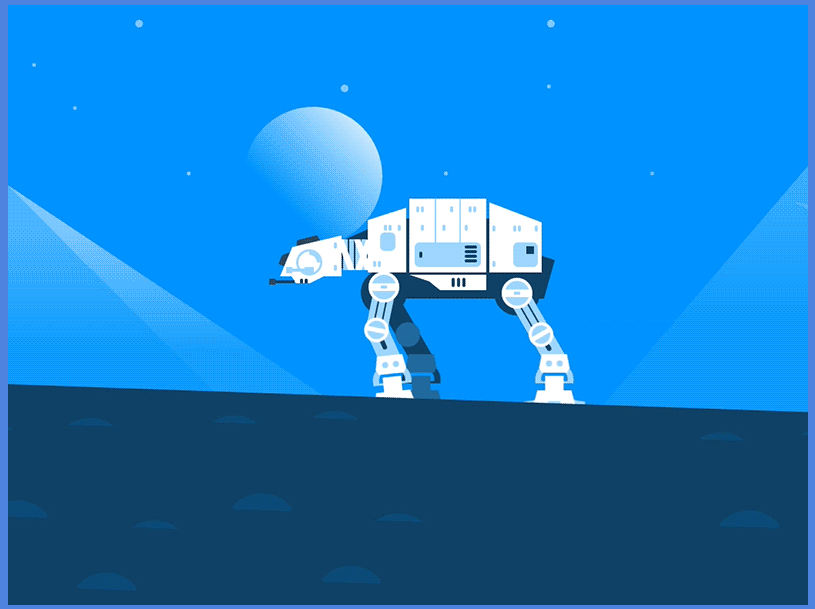
Prepare Your Device

- Android - activate developer mode
  - Settings/about phone tip on Build number until message appears
  - Go back to Settings, then Developer options and check USB debugging
- iPhone
  - Just plug in an press play in xcode – yay!

Demo Time

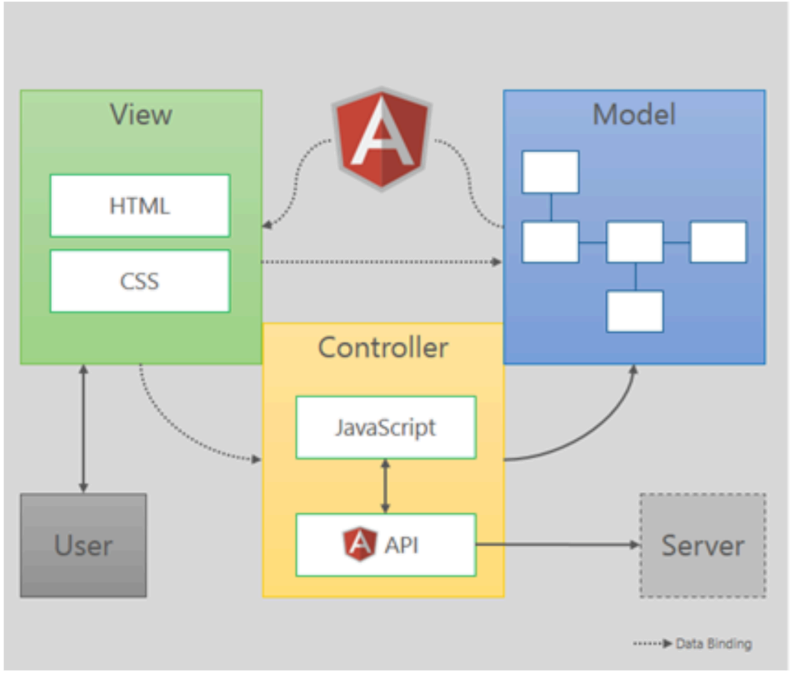

Swisscom
WEAHTER GEO
Weather
Carrier
Chewba
Han Solo
Princess
Admiral
BB8
Wooo
Status
Chats
Status
Search
14:12
64%
Movies
Search it
Rogue One: A Star Wars Story
2016
Moana
2016
Trolls
2016
Why Him?
2016
Office Christmas Party
2016
Sing
2016
FROZEN
14:16
63%
01:20
93 %
Tills
THU
03
SEP | 2015
01:18
94 %
A New Hope
Director • George Lucas
Producer • Gary Kurtz, Rick McCallum
Release Date • 1977-05-25
foo
Chats
Films
Account

Coding Time



Copy the code on Google docs https://goo.gl/E4UFs3

MVC



MVC

Architectural Pattern

Layers

View: Visual appearance (declarative languages)

Model: Data model of the app (JavaScript objects)

Controller: Adds behavior (imperative languages)

Workflow

User interacts with the view

Changes the model, calls controller (data binding)

Controller manipulates model, interacts with server

AngularJS detects model changes and updates the view (two-way data binding)

MVC

```
<ion-grid>
 <ion-row wrap>
   <ion-col width-33 *ngFor="let movie of movies">
   <div tappable (click)="viewItem(this.movie)">
   <img width="100%" class="boarder-shadow layzimg" onload="this.style.opacity='1'"  height="100%" [src]="movie.images.poster">
   </div>
   <div class="description">
     <p style="color: white">{{movie.title}}</p>
     <p style="color: #5b5b5b">{{movie.year}}</p>
   </div>
   </ion-col>
 </ion-row>
</ion-grid>
```

View

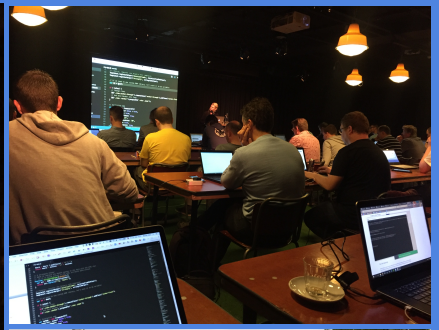
```
import { Component } from '@angular/core';

import { NavController } from 'ionic-angular';
import { ModalController } from 'ionic-angular';
import { MovieData } from '../../providers/movie-data';
import { ItemDetailPage } from '../item-detail/item-detail';
import { ModalSearchPage } from '../modal-search/modal-search';


@Component({
  selector: 'page-page1',
  templateUrl: 'page1.html',
  providers: [MovieData]
})
export class Page1 {
  movies: any;
  pageCounter: any = 1;
  search: any = "";
  navOptions: any;

  constructor(public navCtrl: NavController, public movieService: MovieData, public modalCtrl: ModalController) {
    this.getMovies();
    this.navOptions = {
      animation: 'md-transition'
    };
  }

  getMovies(){
    this.movieService.getMovies()
    .then(data => {
      this.movies = data;
      console.log("fooo " + this.movies);
    });
  }

  viewItem(item){
    this.navCtrl.push(ItemDetailPage, {
      item: item
    });
  }
```

Controller

```
import { Injectable } from '@angular/core';
import { Http } from '@angular/http';
import 'rxjs/add/operator/map';

/*
  Generated class for the MovieData provider.

  See https://angular.io/docs/ts/latest/guide/dependency-injection.html
  for more info on providers and Angular 2 DI.
*/
@Injectable()
export class MovieData {
  data: any;

  constructor(public http: Http) {
    console.log('Hello MovieData Provider');
  }

  getMovies(){
    return new Promise(resolve => {
      this.http.get('https://tv-v2.api-fetch.website/movies/1?sort=trending&order=-1')
      .map(res => res.json())
      .subscribe(data => {
        console.log(data);
        //return data;
        this.data = data;
        resolve (this.data);
      });
    });

  }
```

Model

PhoneGap Day



Template expressions {{...}}

{{title}}

Will be evaluated first and then converted to a string

```
<!-- "The sum of 1 + 1 is 2" -->
<p>The sum of 1 + 1 is {{1 + 1}}</p>

<!-- "The sum of 1 + 1 is not 4" -->
<p>The sum of 1 + 1 is not {{1 + 1 + getVal()}}</p>
```

home.html

```
<h1>{{title}}</h1> <!– will be evaluated as my app -->
```

home.ts

```
title: string; //member variable can be use with this.
this.title = “my app”;
```

Two-way data binding

To use two-way data binding us “the banana in a box” syntax

home.html

```
<ion-label>{{ name }}</ion-label>
<ion-input [(ngModel)]="name"></ion-input>
```

home.ts

```
name: string;
console.log(this.name);
```

Two-way data binding -> when data changes in the view, binded data will be changed in the controller/model and vise versa

TypeScript Basic Types

Boolean
let isDone: boolean = false;

Number
All numbers in TypeScript are floating point values
let decimal: number = 6;

String
let color: string = "blue";
color = 'red';

Array
let list: number[] = [1, 2, 3];
let list: Array<number> = [1, 2, 3];

Any
let notSure: any = 4;

(click) (tap) tappable 300ms

You can call a function by using

- (click) event will call the function when tapping on an element and even when pressing and holding it then releasing it.

  <button tappable (click)="back()" ion-button>

- (tap) event will call the function when tapping on an element. Pressing and resting your finger on that element and releasing your finger won't call the function.

tappable

Elements clicked which are not a button or <a> will have a 300ms delay. To remove the 300ms use the attribute *tappable* to remove it like so:

<div tappable (tap)="viewItem(this.movie)">

Generate a new page

- Type the following line into your console to create a new page

  ionic g page myPage

- This will generate the html, ts and scss files for you
  - app/pages/my-page/my-page.html
  - app/pages/my-page/my-page.ts
  - app/pages/my-page/my-page.scss

Import a new page

- Then import and add the new page to your module list in src/app/app.module.ts

```
import { MyPage } from '../pages/my-page/my-page';
@NgModule({
 declarations: [
MyApp,
MyPage
],
imports: [
IonicModule.forRoot(MyApp)
 ],
bootstrap: [IonicApp],
entryComponents: [
MyApp,
MyPage
  ],
....
```

Generate a new service/provider

- Type the following line into your console to create a new provider

  ionic g provider MyData

- This will generate the ts file for you
  - app/providers/my-data/my-data.ts

Import new service/provider

- Then import and add the new provider to your provider list in src/app/app.module.ts

```
import { MyData } from '../providers/my-data';
…
providers: [
StatusBar,
SplashScreen,
{provide: ErrorHandler, useClass: IonicErrorHandler},
MyData ]
…

//Ionic 2 uses kebab-casing for file names (my-about-page.html) and css classes (.my-about-page), and uses PascalCasing for JavaScript classes in ES6/TypeScript (MyAboutPage).
```

SASS

Define in src/theme/variables.scss

```
$colors: (
primary:    #387ef5,
secondary:  #32db64,
danger:     #f53d3d,
light:      #f4f4f4,
dark:       #222
);
```

Use:

```
<ion-navbar color="secondary">
```

Defining variables in your components .scss file like so to make us of variables:

```
$my-variable: red;
.my-class{
        background-color:  $my-variable;
}
```

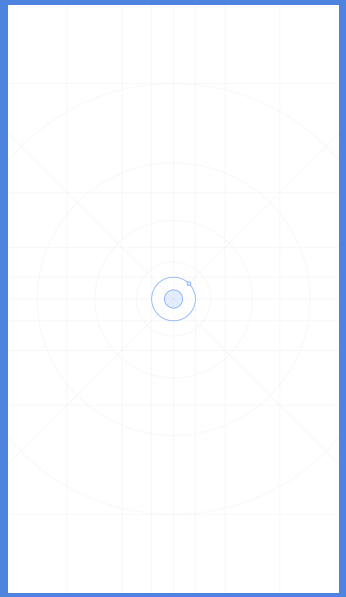
To change Ionics preset plattform variables, override them in your variables.sccs file.
You can find all variables on:
http://ionicframework.com/docs/theming/overriding-ionic-variables/

App icon and splash screen

- Place your icon and splash (png, psd or ai) in the resrouces directory of the root of your app -> myapp/resources/icon.png

- Icon min 192x192px and **NO** rounded corners
  Tip: use 1024x1024px

- For different icons/splash per plattfom place icon into resources/android/icon.png and resources/ios/icon.png

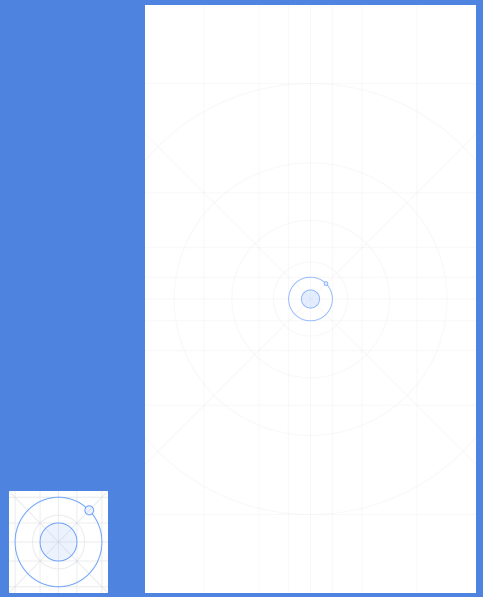


App icon and splash screen

- Generate app icons
  $ ionic resources –icon

- Generate splash screen
  $ ionic resources –splash

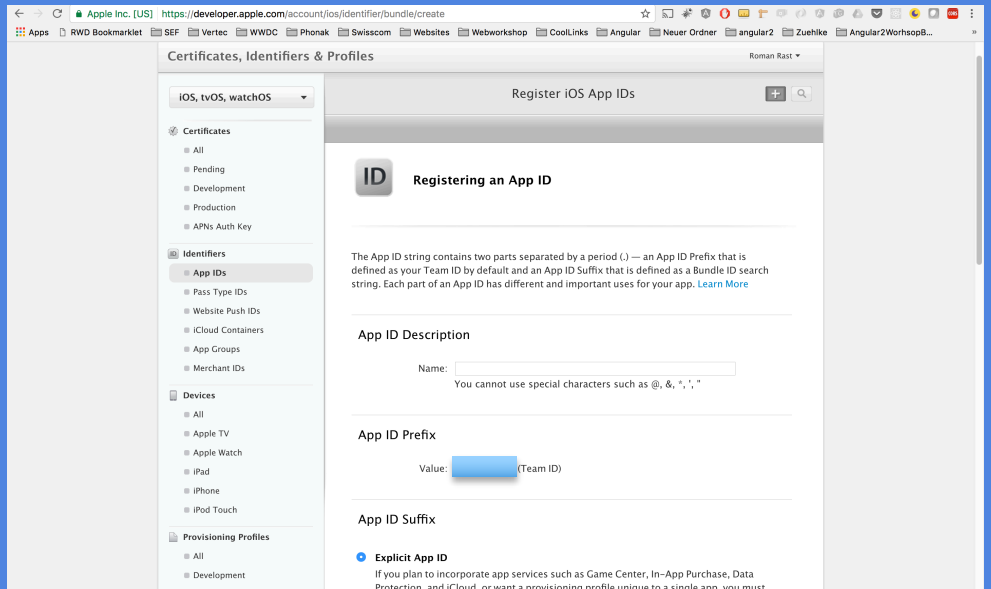
- Generate ionc & splash
  $ Ionic resources



Ready for Store & Push Notification

iOS: App ID

- Create your App ID in the Apple Developer Center
  https://developer.apple.com/account
- Certificates, Identifiers & Profiles > Identifiers > App IDs



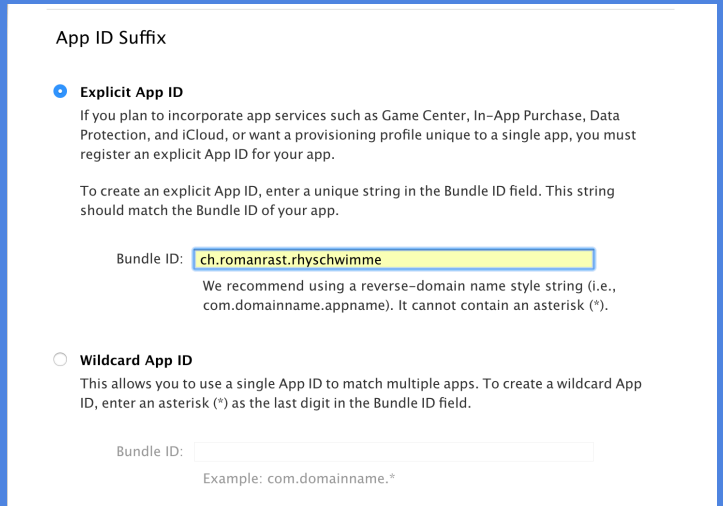
Ready for Store & Push Notification

- Select Explicit App ID and type in your Bundle ID as Reverse-Domain-Name for example ch.romanrast.rhyschwimme

- Make sure it‘s the same ID as in your Ionic Projects config.xml
- Select Push Notifications if your app should provide Push Notifications

app.module.ts | config.xml × | app.component.ts | overview.txt

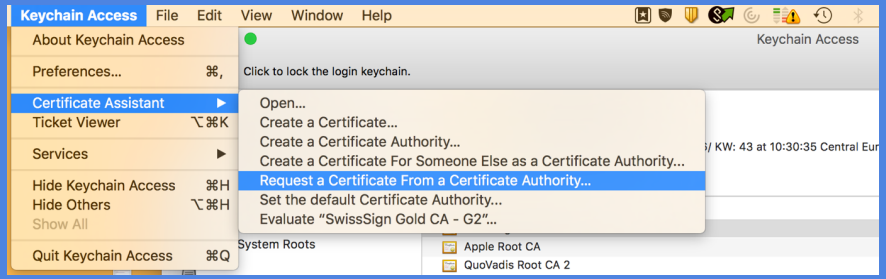
```
1  <?xml version='1.0' encoding='utf-8'?>
2  <widget id="ch.romanrast.rhyschwimme" version="0.0.1" xmlns='
3      <name>Rhyschwimme</name>
```

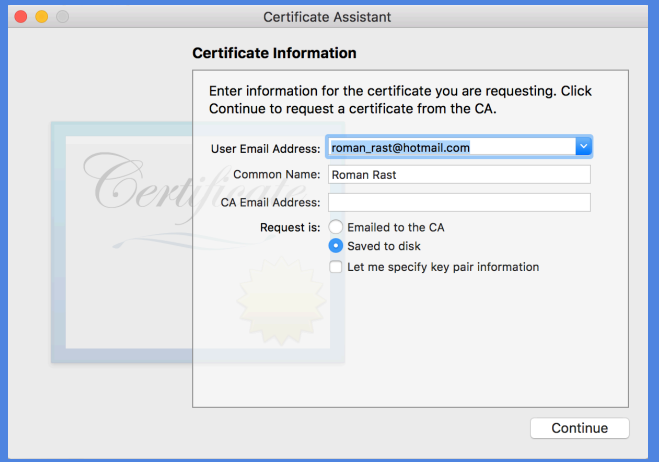


Ready for Store & Push Notification

iOS: Certificates

- First request a certificate signing request file
  - Keychain Access > Certificate Assistant > Request a Certificate From a Certificate Authority



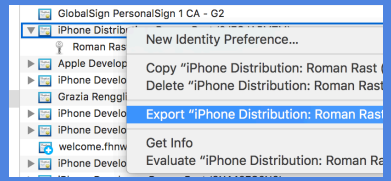


Ready for Store & Push Notification

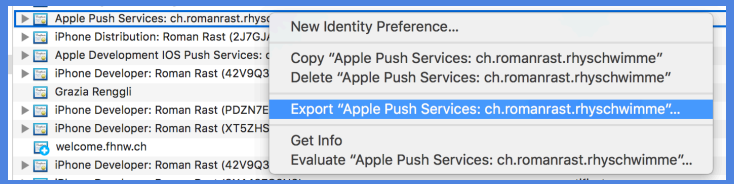
iOS: Certificates

- Next go to Certificates > Identifiers & Profiles > Certificates > Development
- Upload CSR File (xxx.certSigningRequest)
- Download the .cer file
- Double click .cer file and open it in Keychain Access > Login
- Right click > Export and save

Export for iPhone Distribution



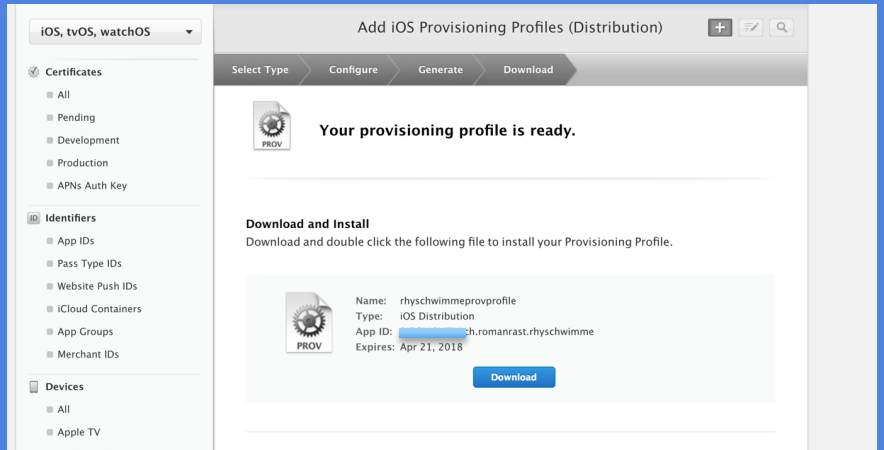
Export for Push Service



Ready for Store & Push Notification

iOS: Profisioning Profiles

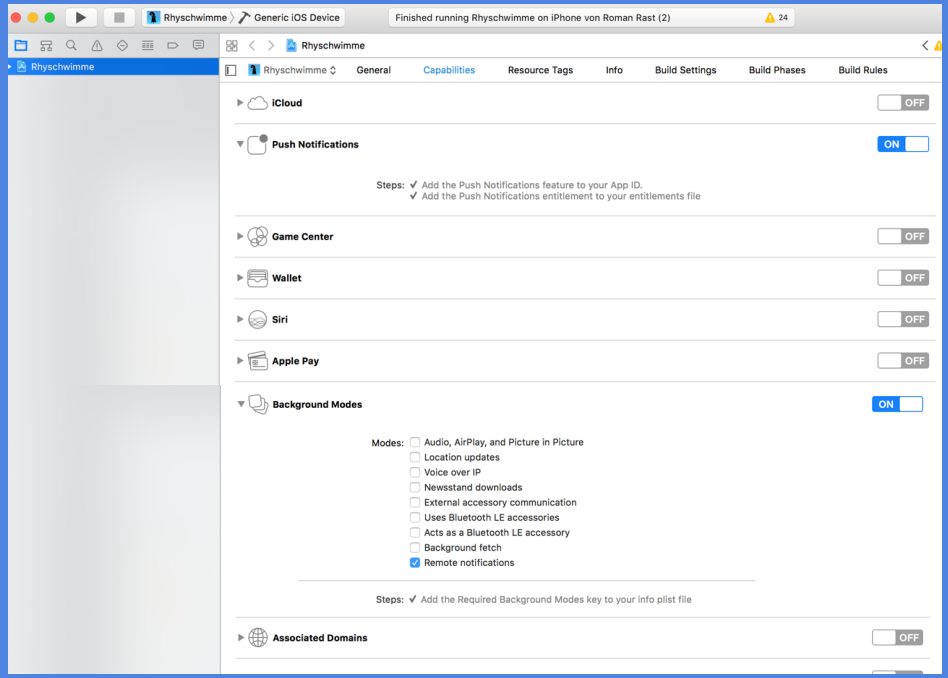
- Next go to Certificates > Identifiers & Profiles > Provisioning Profiles> Development
- Select App Store (Distribution)
- Download the .mobileprovision file
- Double click .cer file and open it in Keychain Access > Login
- Right click > Export and save



Ready for Store & Push Notification

iOS: Capabilities in Xcode

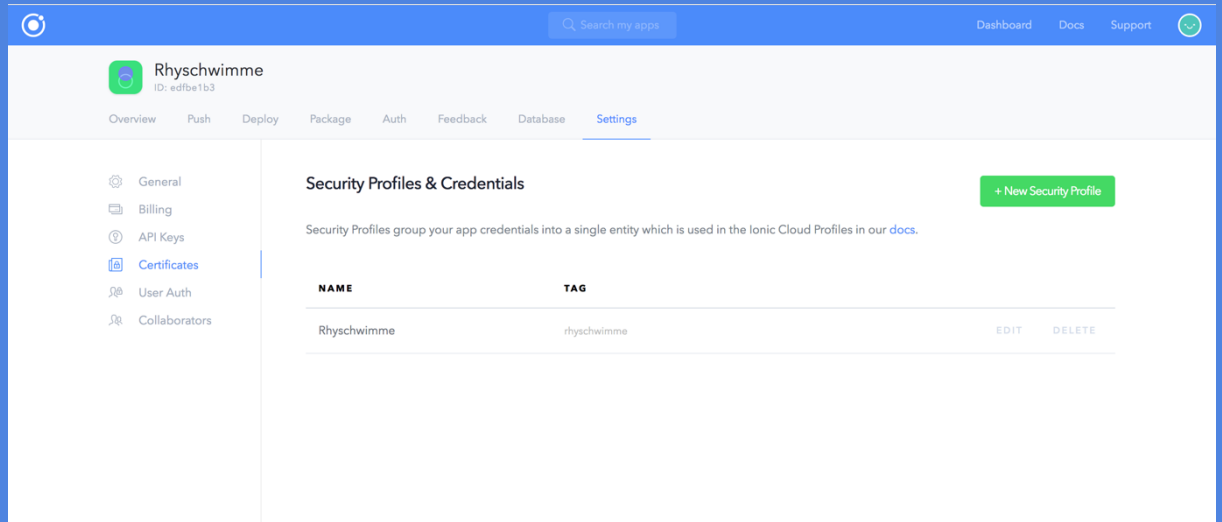
- Set Push Notification and Background Modes to ON and check Remote notification



Ready for Store & Push Notification

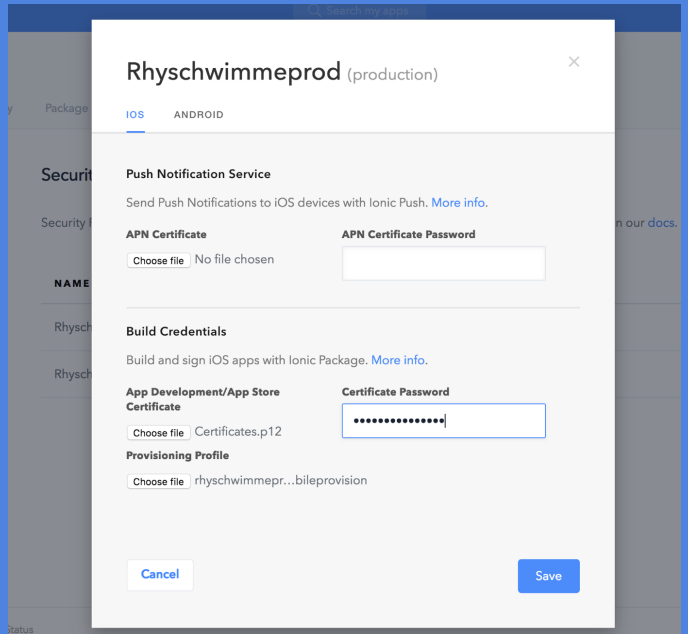
Upload your App to ionic.io

- In your terminal type ionic upload
- On https://apps.ionic.io go to your new app > Settings > Certificates > +New Security Profile



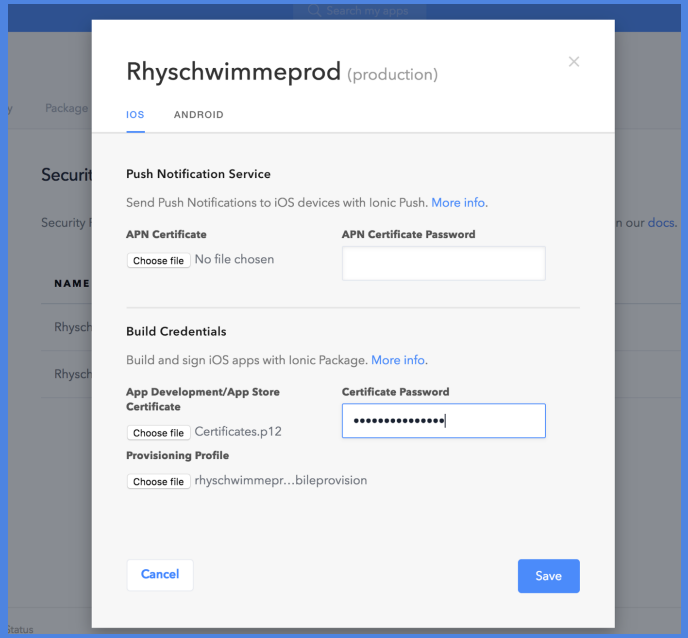
Ready for Store & Push Notification

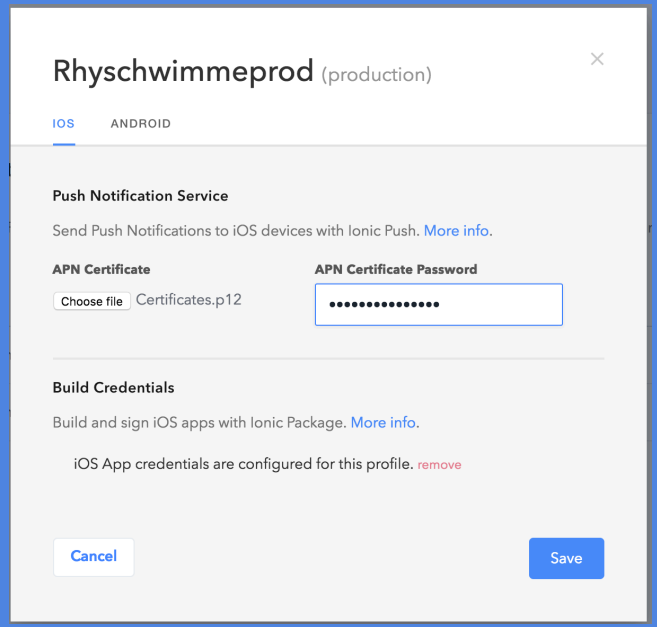
iOS: Upload your Certificates



Ready for Store & Push Notification

iOS: Upload your Certificates for Build Credentials and Push Notfication Service





Ready for Store & Push Notification

Android: Keystore

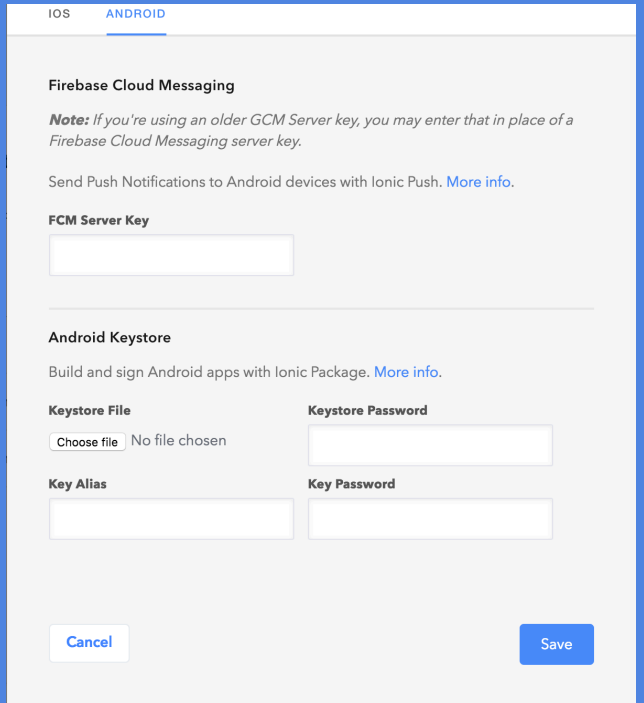
- To sign your app
- Open your terminal on the root and type

  keytool -genkey -v -keystore MY-RELEASE-KEY.keystore -alias MY_ALIAS_NAME -keyalg RSA -keysize 2048 -validity 10000
- REMEBER YOUR PASSWORD!
- Generates a .keysotre file in your root

Ready for Store & Push Notification

Android: Keystore

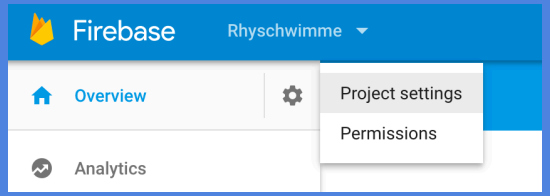
- Upload Keystore to ionic.io

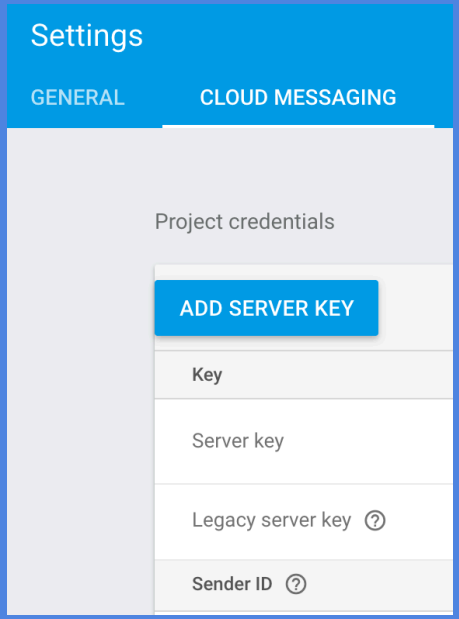


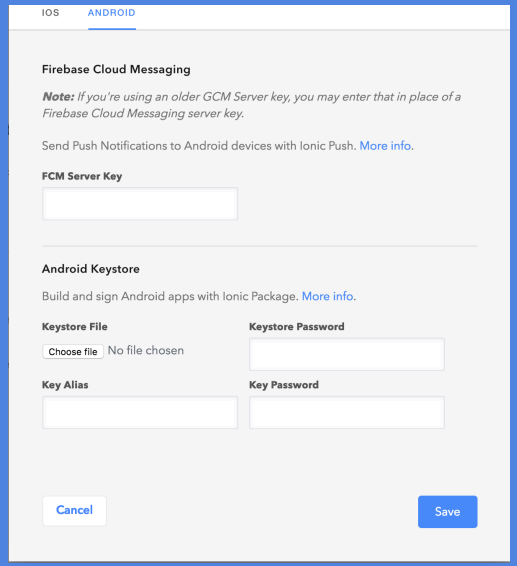
Ready for Store & Push Notification

Android: get your FMC (Firebase Cloud Messaging) Server Key

- https://console.firebase.google.com
- Create new a project and go to Project settings and find your (FMC) Server Key and Sender ID







Ready for Store & Push Notification

Install Push Plugin

- npm install @ionic/cloud-angular --save
- Cordova plugin add phonegap-plugin-push –variable SENDER_ID=123456 –save
- Use Sender ID from your Firebase project

Ready for Store & Push Notification

Import Push

- src/app/app.module.ts

```
import { CloudSettings, CloudModule } from '@ionic/cloud-angular';

const cloudSettings: CloudSettings = {
  'core': {
    'app_id': 'abcde123'
  },
  'push': {
    'sender_id': '1234567890',
    'pluginConfig': {
      'ios': {
        'badge': true,
        'sound': true
      },
      'android': {
        'iconColor': '#ffffff'
      }
    }
  }
};
```

```
imports: [
  IonicModule.forRoot(MyApp),
  CloudModule.forRoot(cloudSettings),
  BrowserModule,
  HttpModule
],
```

app_id from Ionic.io
sender_id from Firebase

Ready for Store & Push Notification

Import Push

- src/app/app.components.ts

```
import { Push, PushToken } from '@ionic/cloud-angular';
```

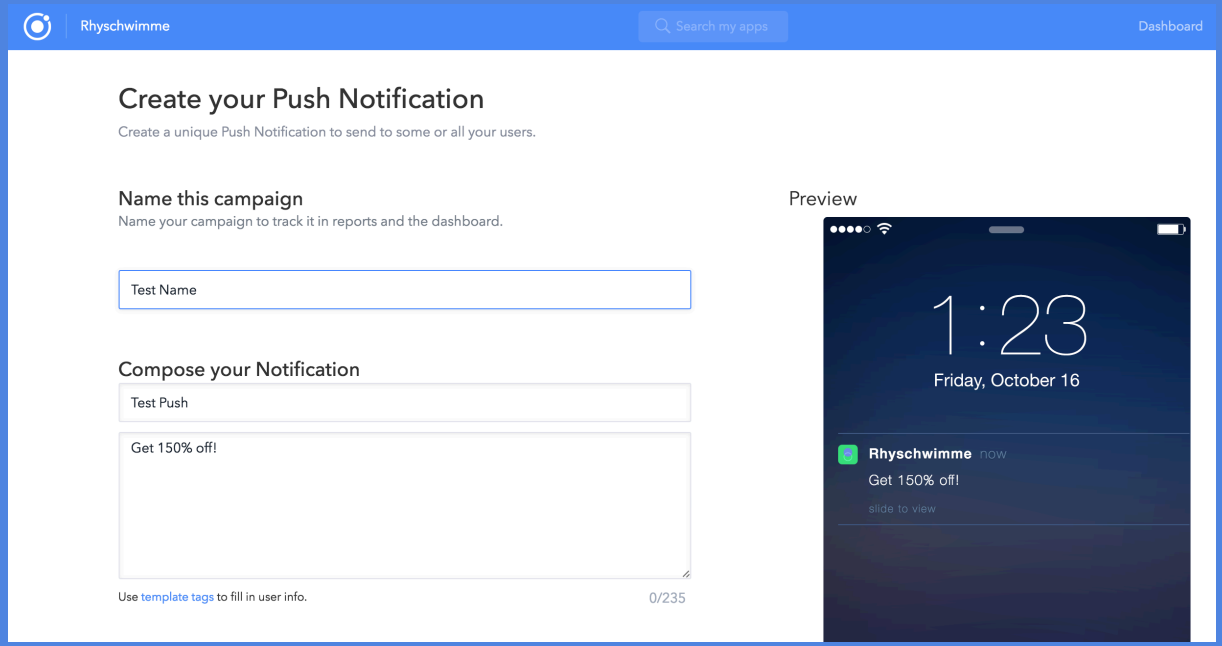
```
constructor(public platform: Platform, public statusBar: StatusBar, public splashSc
reen: SplashScreen, public push: Push) {…}
```

```
  initializeApp() {
    this.platform.ready().then(() => {
      ...
     //pushnotification
      this.push.register().then((t: PushToken) => {
        return this.push.saveToken(t);
      }).then((t: PushToken) => {
        console.log('Token saved:', t.token);
      });

      //pushnotification
      this.push.rx.notification()
      .subscribe((msg) => {
        console.log('I received awesome push: ' + msg);
      });


    });
  }
```

Ready for Store & Push Notification

Send Puhs Notification

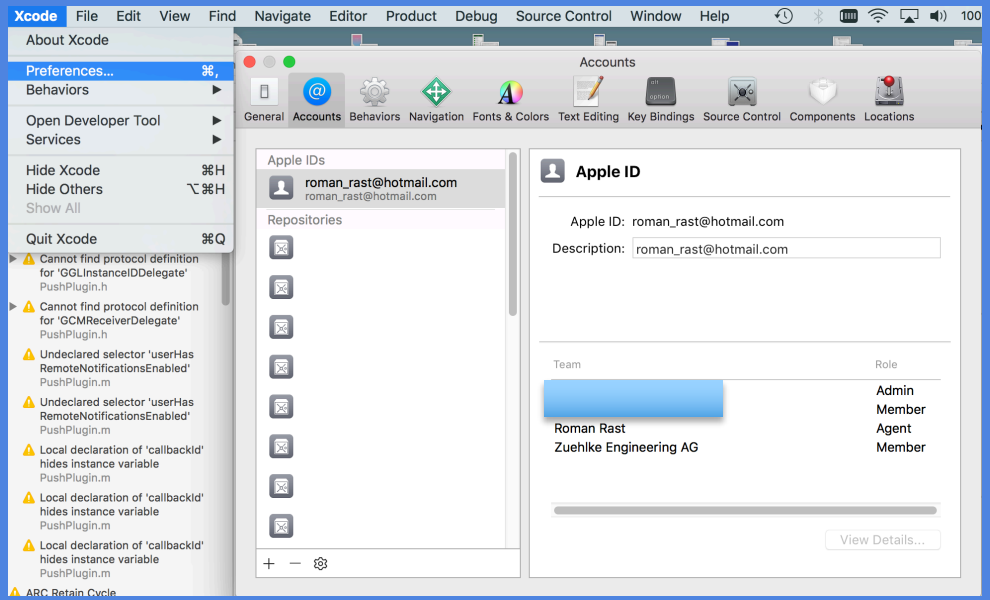
- https://apps.ionic.io > Push > Create New Push



Ready for Store & Push Notification

Release for App Store

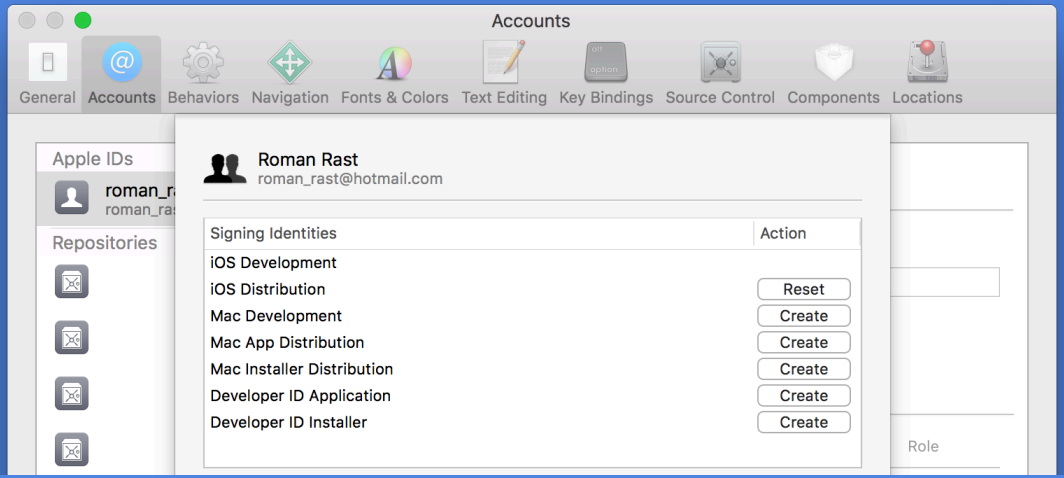
- Cordova plugin rm cordova-plugin-console
- Get signing identity > Xcode > Prefernces > Your Apple ID > Your Team > View Details



Ready for Store & Push Notification

Release for App Store

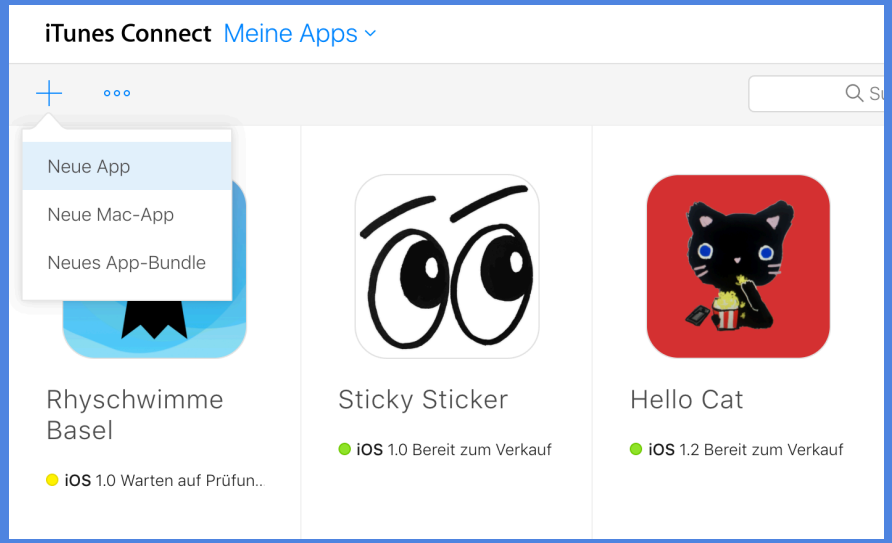
- Press Create next to iOS Distribution (once pressed it will turn into Reset)

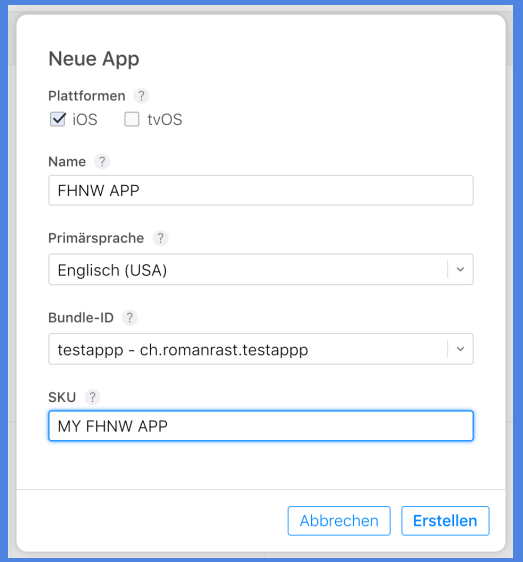


Ready for Store & Push Notification

Release for App Store

- Create App Store Listing on https://itunesconnect.apple.com/

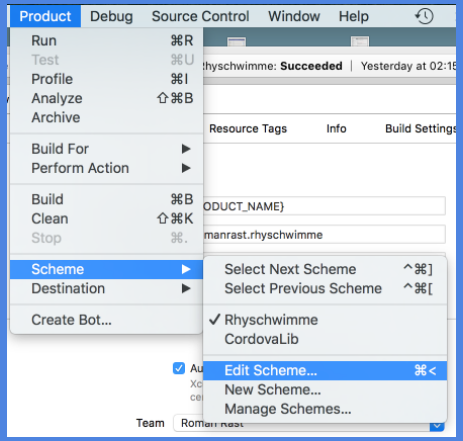


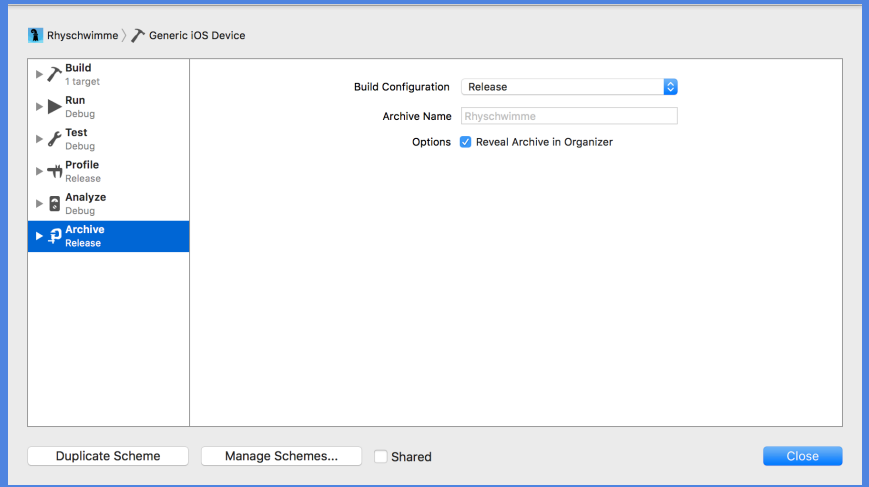


Ready for Store & Push Notification

Release for App Store

- To create a release build, type in your terminal:
  - Ionic build ios --release
- Open your Xcode Project and go to Product > Edit Scheme > Archive and set Build Configuration to Release

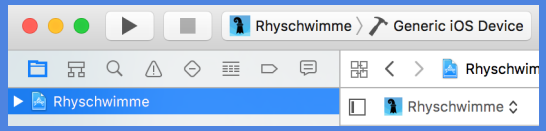


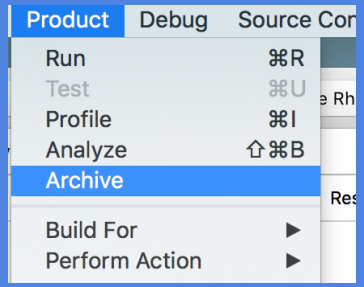


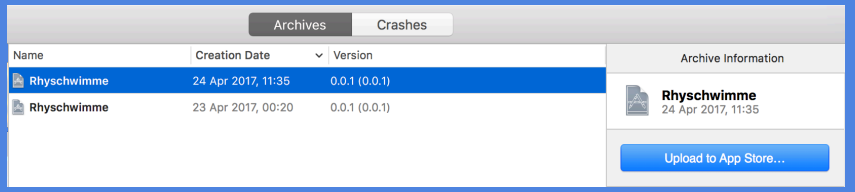
Ready for Store & Push Notification

Release for App Store

- Create Archive
  - Set Device to Generic iOS Device
  - Product > Archive > pick your app > Upload to App Store







Ready for Store & Push Notification

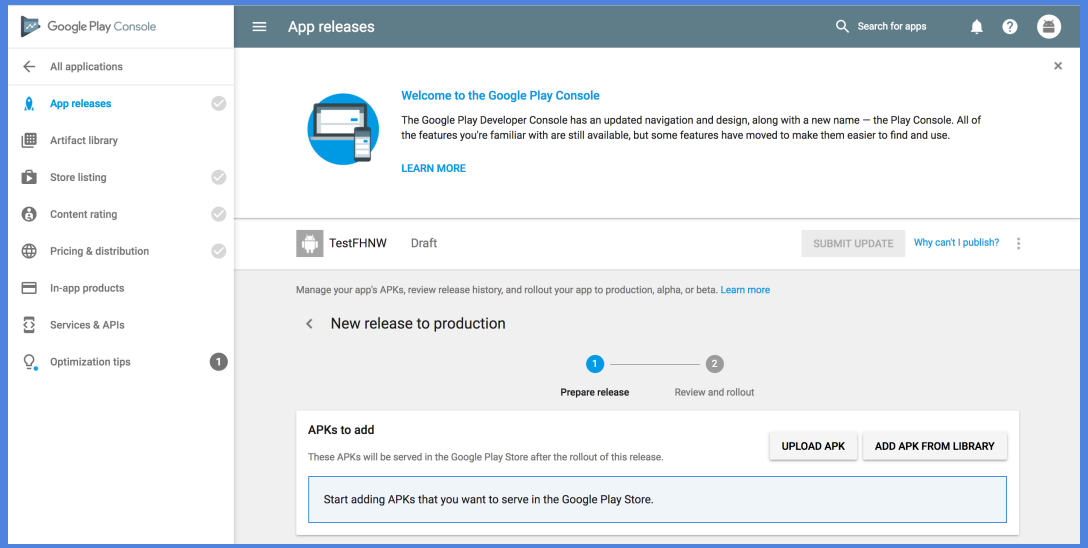
Release for Google Play Store

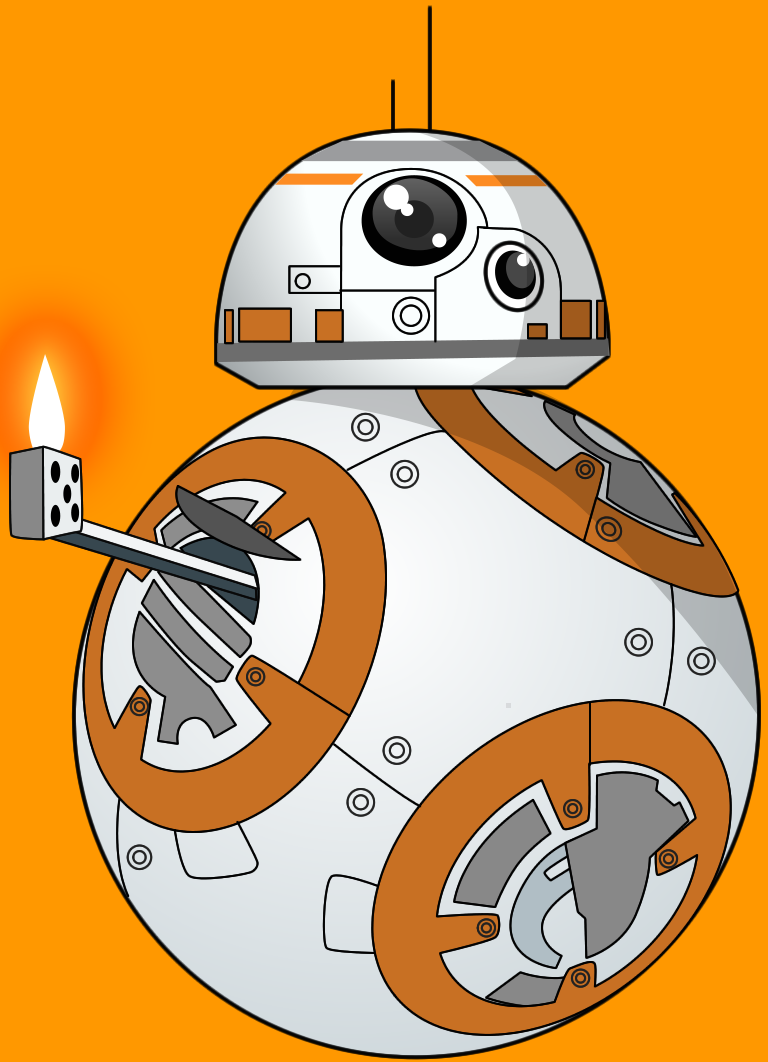
- Create release version of your android app
  - corodva build --release android
- This generates an unsigned apk -> platforms/android/build/outputs/apk/android-release-unsigned.apk
- Sign apk with
  - jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore android-release-unsigned.apk alias_name
- Next optimize your apk with zipalign
  - zipalign –v 4 android-release-unsigned.apk rhyschwimme.apk

Ready for Store & Push Notification

Release for Google Play Store

- Uplaod you apk to https://play.google.com/apps
- Create Application > App releases > Manage Production > Upload APK
- And also fill out further infos





Tips

Change status bar color
app.components.ts

```
initializeApp() {
  this.platform.ready().then(() => {
    // Okay, so the platform is ready and our plugins are available.
    // Here you can do any higher level native things you might need.
    //this.statusBar.styleDefault();
   // this.statusBar.styleLightContent();

    this.statusBar.overlaysWebView(false);
    this.statusBar.backgroundColorByHexString('#995f72');

    this.splashScreen.hide();
```

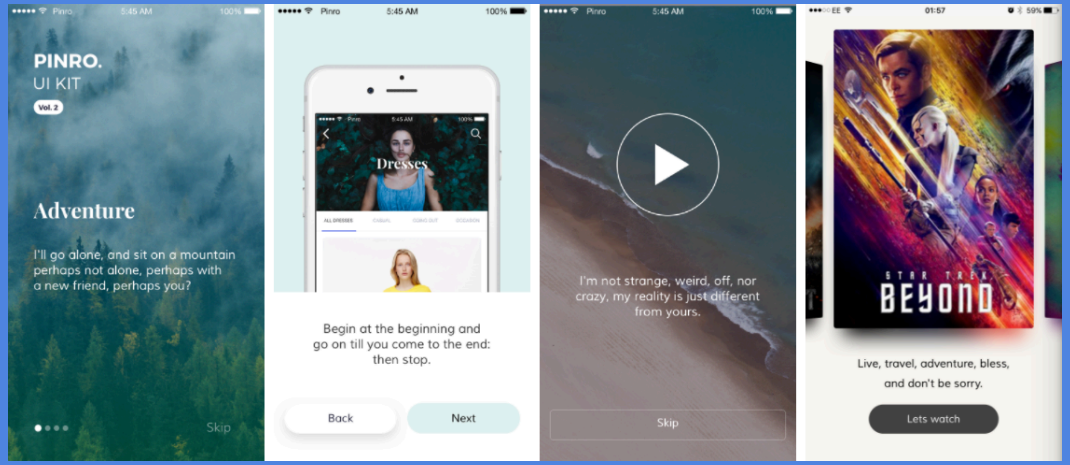
Change content background color
variables.scss

```
$background-ios-color: rgba(244, 244, 244, 1);
```

Change tab bar icon and text color
variables.scss
fun is just a predefined color like primary or danger

```
$tabs-ios-tab-text-color-active: color($colors, fun);
$tabs-ios-tab-icon-color-active: color($colors, fun);
```

Tips



https://market.ionic.io/

- Plugins & Templates
- Try them with Ionic View
  - A202129d
  - 73be3cec
  - A247ECC0
  - https://market.ionic.io/plugins/ionic-stretchy-header
  - https://market.ionic.io/plugins/swipedtabs-ionic-2