



ionic

Building Hybrid Apps with Angular and Ionic

Use Web Technologies

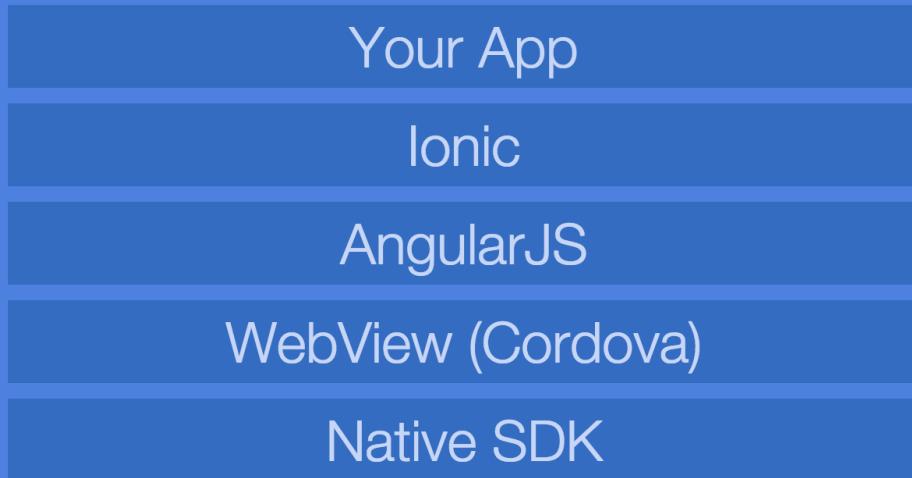


Ionic

- 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
- Ionic 3.0 in April 2017
- High performance
- iOS > 7, Adnroid > 4.1
- Open source
- Many Frameworks and libraries built-in
 - Angular, Cordova, Gulp, Sass, NodeJS, Bower, hammer.js
- Custom directives/components -> <ion-list>
- Drifty Co



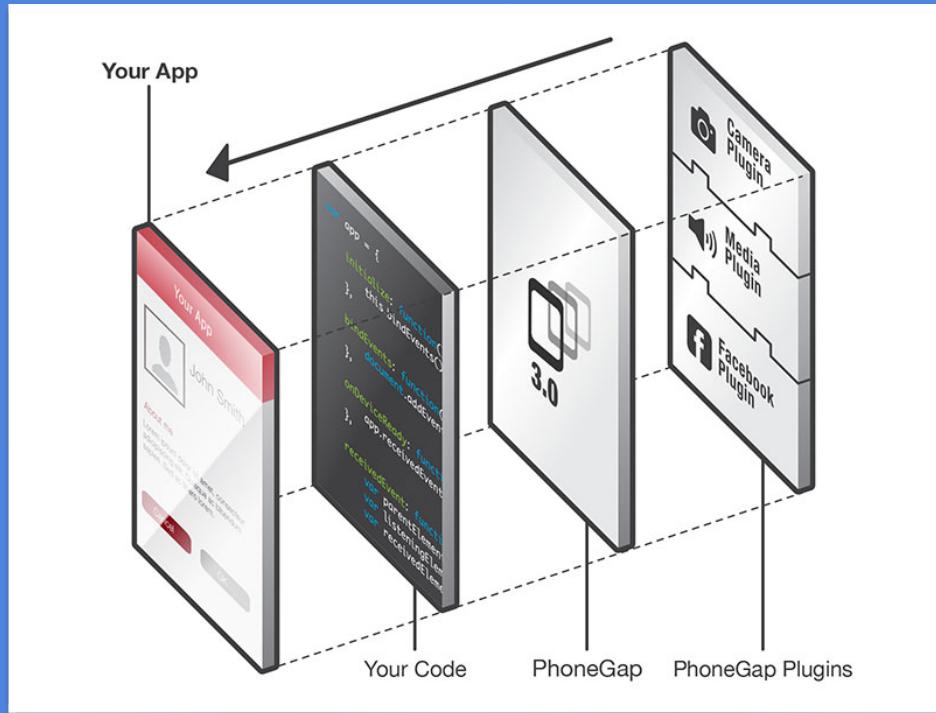
Architecture





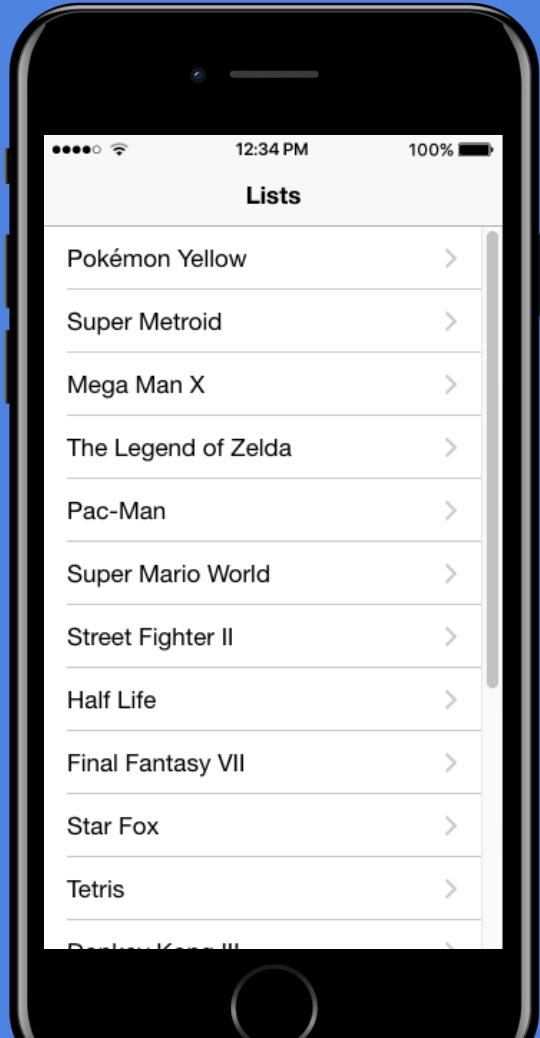
- Wraps app into a native app
- App uses WebView (browser of the phone without url bar and other stuff)
- Offers APIs to access camera, contacts, accelerometer finger print scanner etc.
- Develop a single code base for iOS and Android...
- PhoneGap/Cordova - Adobe





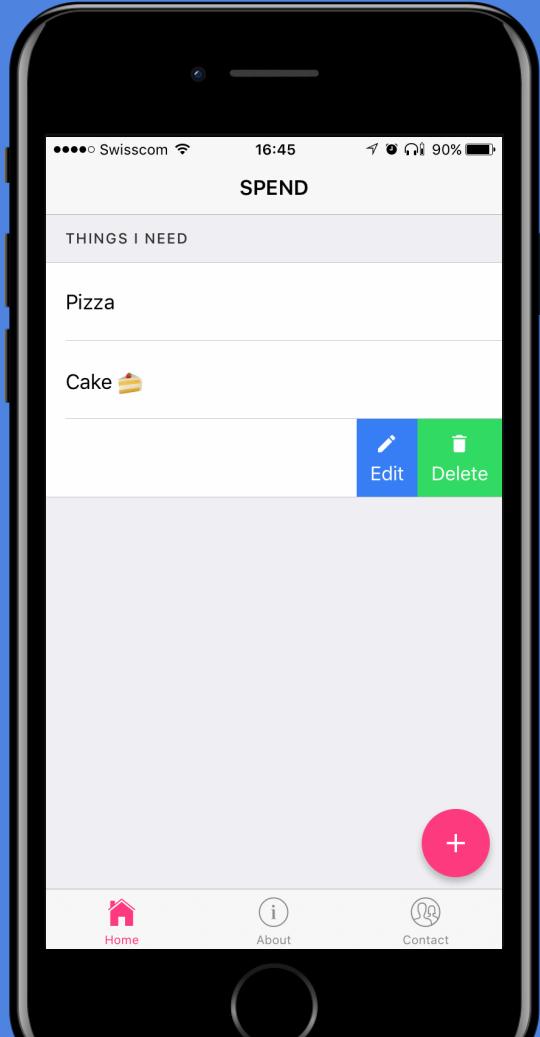
Pro | Cons

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



Lists

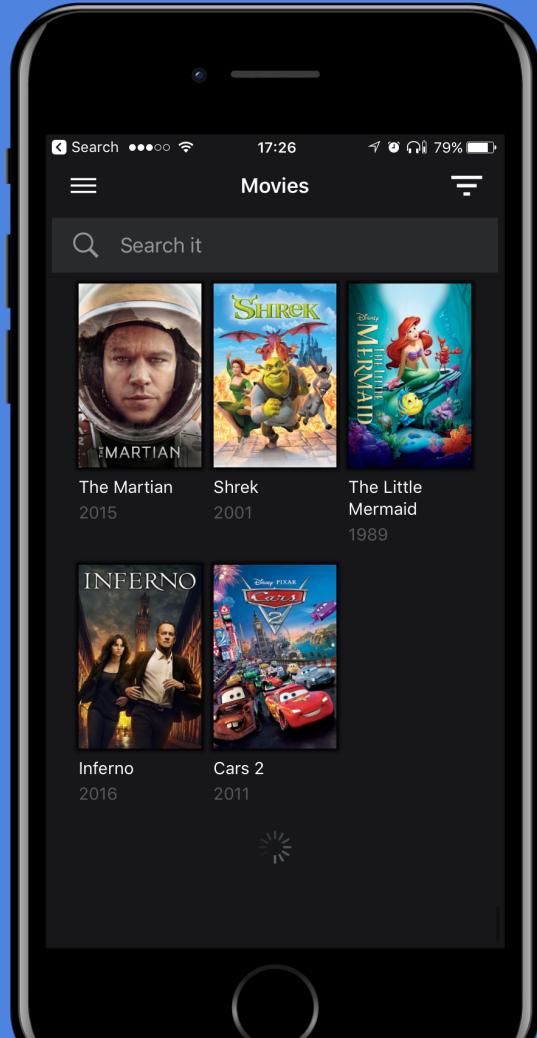
```
<ion-list>
  <button ion-item *ngFor="let item of items" (click)="itemSelected(item)">
    {{ item }}
  </button>
</ion-list>
```



Sliding List

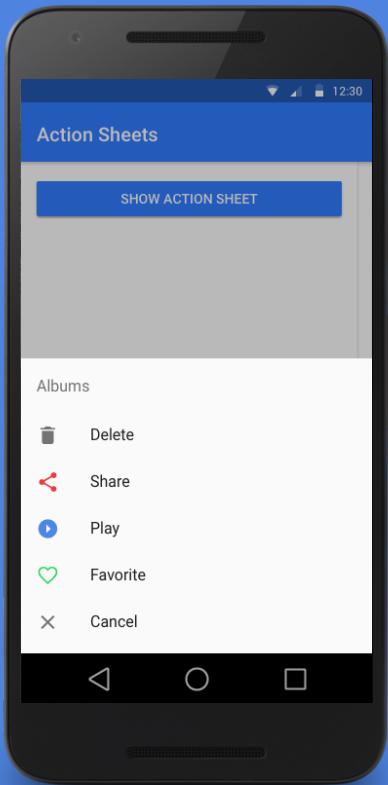
```
<ion-list>
  <ion-list-header>
    Things I need
  </ion-list-header>
  <ion-item-sliding *ngFor="let item of items | async">
    <ion-item (tap)="viewItem(item.$key, item.name)">
      {{item.name}}
    </ion-item>
    <ion-item-options side="right">
      <button ion-button color="primary" (click)="updateItem(item.$key,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

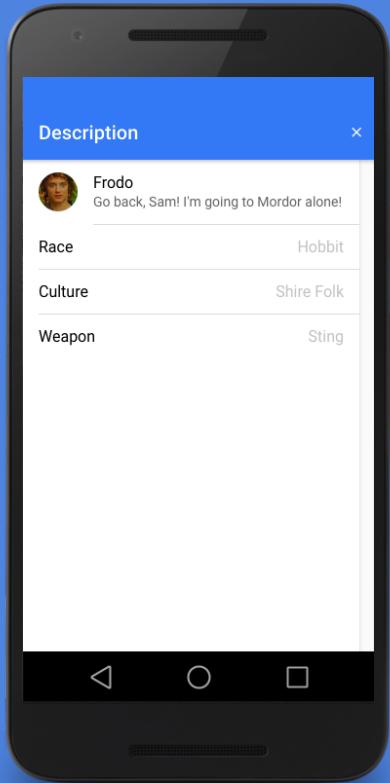


Action Sheet

- 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

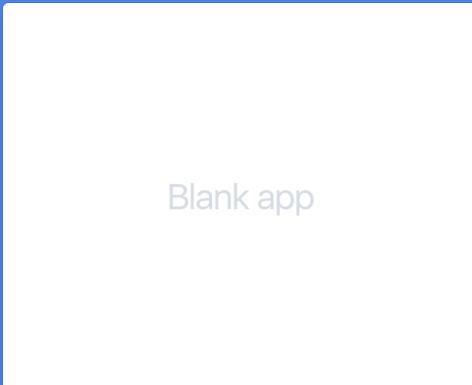


```
import { ModalController } from 'ionic-angular';
import { ModalPage } from './modal-page';

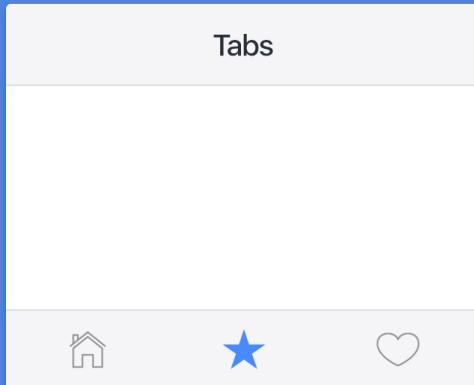
export class MyPage {
  constructor(public navCtrl: NavController) {}

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

Templates

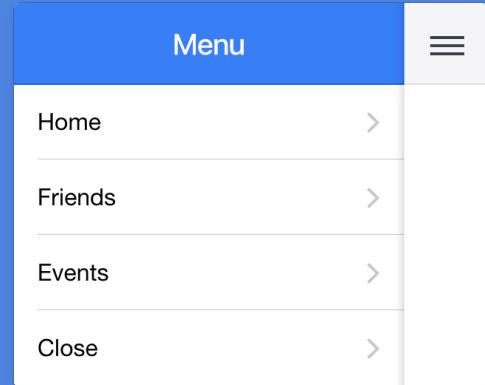


`ionic start myApp blank`



`ionic start myApp tabs`

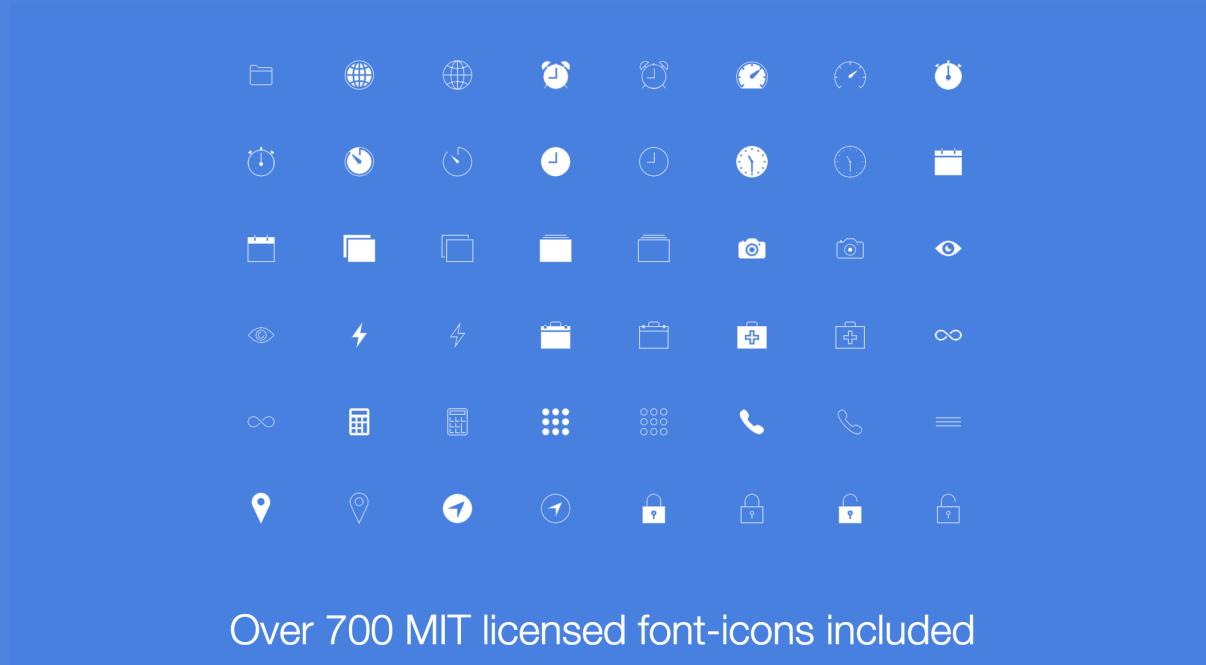
- Nested views
- Each tab has its own nav history



`ionic start myApp sidemenu`

Ionicons

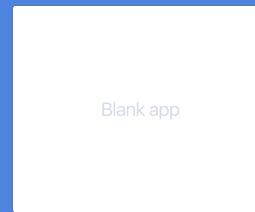
<ion-icon name="md-trash"></ion-icon>



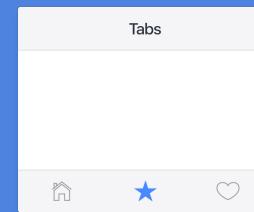
Over 700 MIT licensed font-icons included

Installation

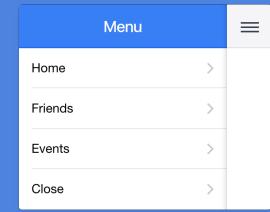
- Install Node.js (<https://nodejs.org/>)
- \$ npm install -g cordova ionic
- \$ sudo ionic start myApp tabs
- \$ cd myApp
- (change access rights if necessary on mac)
- \$ sudo ionic cordova platform add ios | android
- \$ sudo ionic serve
- \$ sudo ionic cordova build ios | android
- \$ sudo ionic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



ionic start myApp
blank --v2



ionic start myApp
tabs --v2



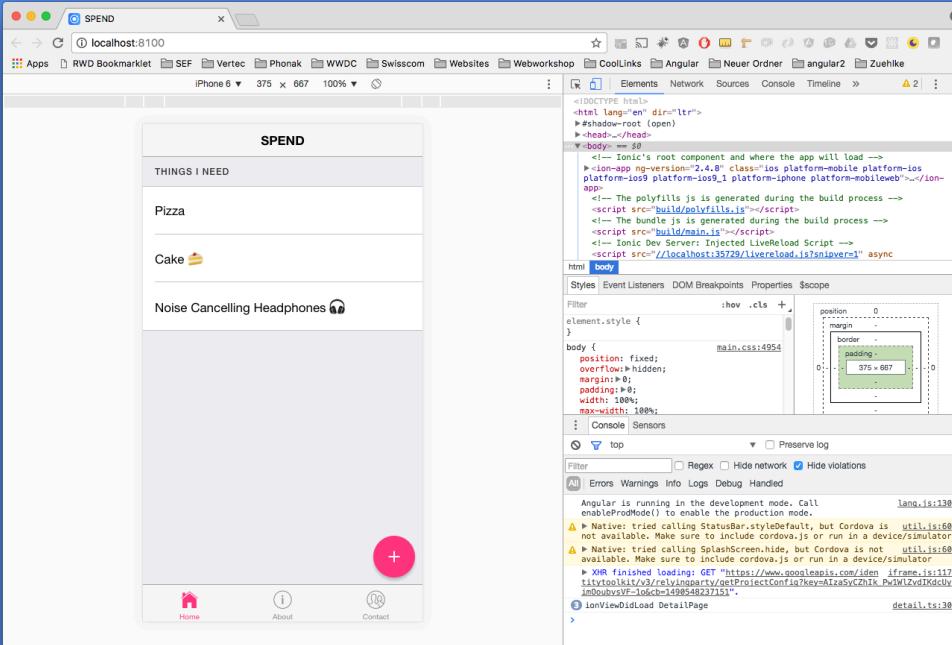
ionic start myApp
sidemenu --v2

Keep Xcode and iOS versions always up to date!

Debugging & Testing

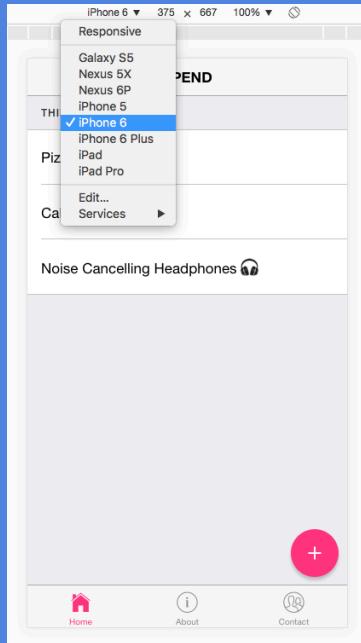
- Use Google Chrome!
- Developer Tools

Chrome Developer Tools

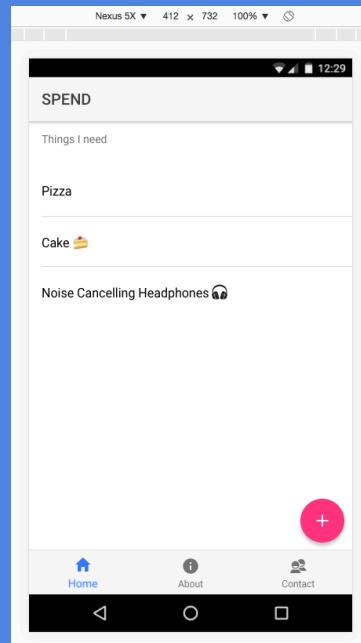


View > Developer > Developer Tools
Shortcut Mac alt+cmd+i

Chrome Developer Tools

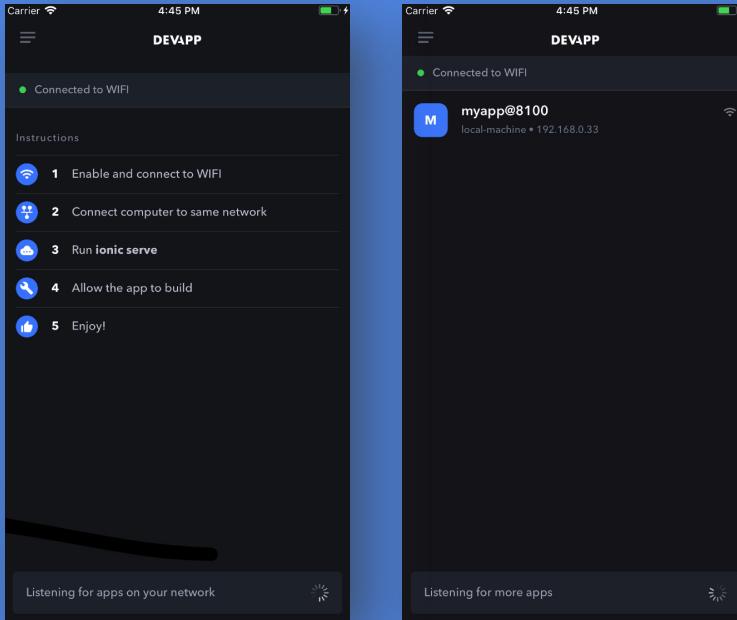


iOS Design



Android Design

Ionic DevApp



Plugins & Ionic Native

- Out of the box
 - cordova-plugin-splashscreen
 - cordova-plugin-statusbar
 - ionic-plugin-keyboard
- More on Ionic Native
 - <http://ionicframework.com/docs/native/>
 - Camera, Barcode Scanner, Geolocation, Geofence, Gyroscope,...

App Structure

- Work in the **src** folder on the **root** of your app
- App is served on `www/index.html`
- The code will be compiled and copied to `platforms/ios` or `platforms/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 cordova run android --device`

App Structure

```
▸ SPEND
  ▶ .vscode
    ▶ node_modules
    ▶ platforms
    ▶ plugins
    ▶ resources
  ▲ src
    ▲ app
      app.component.ts
      app.html
      app.module.ts
      app.scss
      main.ts
    ▶ assets
    ▲ pages
      ▶ about
      ▶ contact
      ▶ detail
    ▲ home
      home.html
      home.scss
      home.ts
    ▶ tabs
    ▲ theme
      variables.scss
      declarations.d.ts
      index.html
      manifest.json
      service-worker.js
  ▶ www
    .editorconfig
    .gitignore
    config.xml
    ionic.config.json
    package.json
    tsconfig.json
    tslint.json
```

Prepare Your Device

- Android - activate developer mode
 - Settings>about phone>Software info> tip on Build number until message appears (7 times) „You are now a devloper!“
 - Go back to Settings, then Developer options and check USB debugging
 - Then in your console: ionic cordova run android --device
- iPhone
 - Open Xcode and just plug in an press play – yay!
 - You might need to pick your team (most likely your Apple ID) in the Signing section under Gereral.

CLI

```
$ cd myApp
```

```
# Add iOS and Android platform for cordova
```

```
$ ionic cordova platform add ios | $ ionic cordova platform add android
```

```
# Build Debug version
```

```
$ ionic cordova build ios | $ionic cordova build android
```

```
# Build Prod version
```

```
$ ionic cordova build ios --release
```

```
# Run in Browser
```

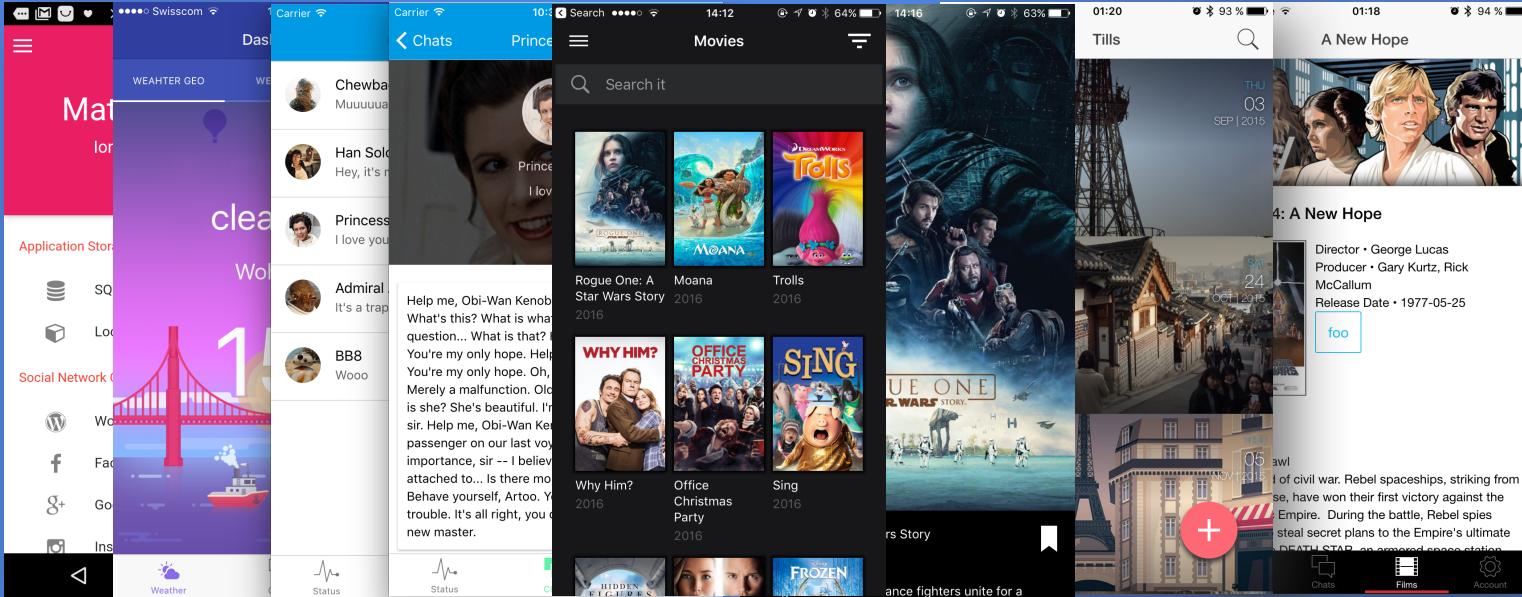
```
$ ionic serve
```

```
$ ionic serve --lab
```

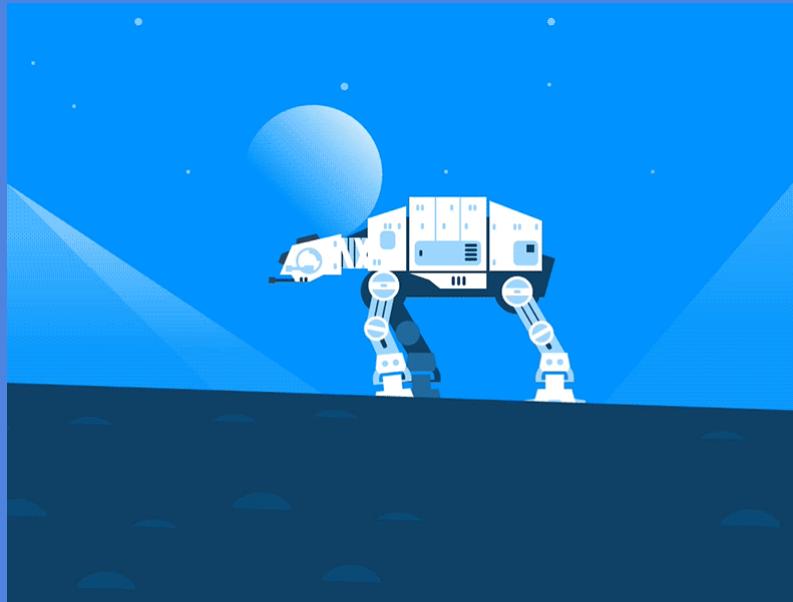
```
# Run on real device if connected
```

```
Android: $ ionic cordova run android --device
```

Demo Time

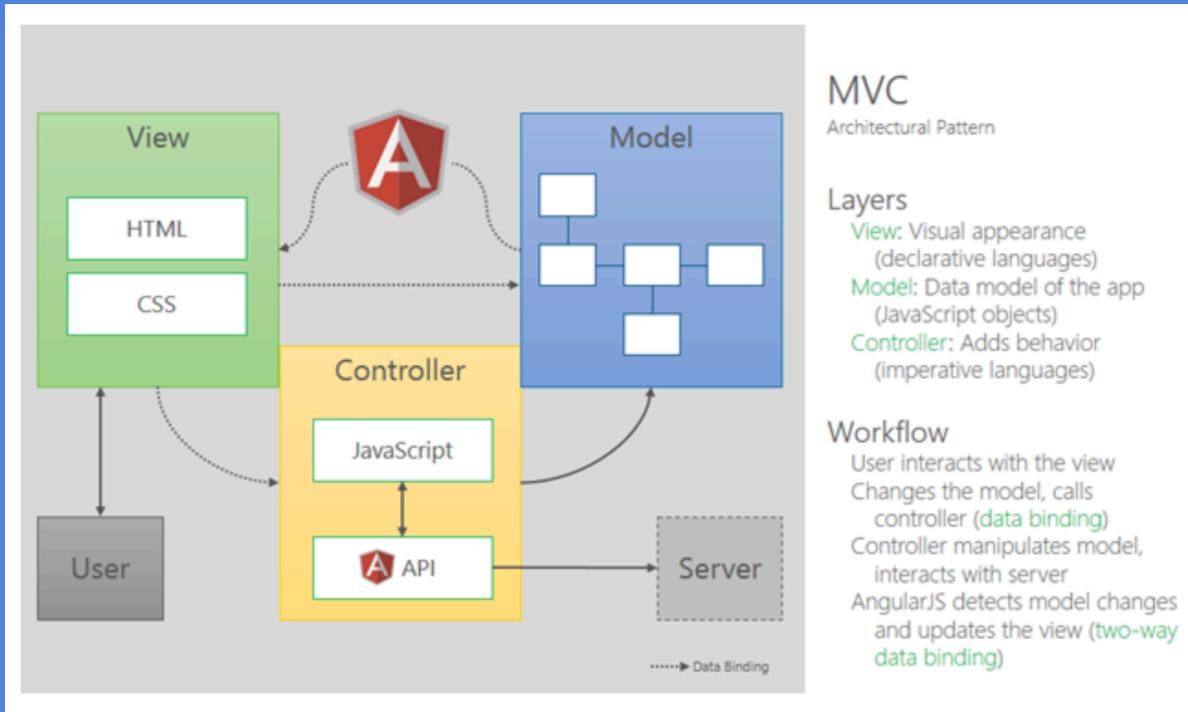


Coding Time



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

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="bordered-shadow lazyimg" 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 (page1.html)

```

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("foo " + this.movies);
    });
  }

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

```

Controller (page1.ts)

```

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);
        });
    });
  }
}

```

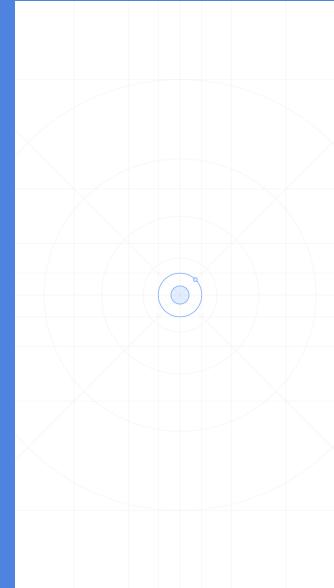
Model (movie-data.ts)

Exercise: Weather App



App icon and splash screen

- Place your icon and splash (png, psd or ai) in the resources directory of the root of your app
 - > myapp/resources/icon.png
 - > myapp/resources/splash.png
 - (name must be icon.png and splash.png)
- Icon min 192x192px and **NO** rounded corners
 - Tip: use 1024x1024px
- Splash 2732x2732px
- For different icons/splash per platform place icon into resources/android/icon.png and resources/ios/icon.png



App icon and splash screen

- Generate app icons
 \$ ionic cordova resources --icon
- Generate splash screen
 \$ ionic cordova resources --splash
- Generate ionc & splash
 \$ Ionic cordova resources

