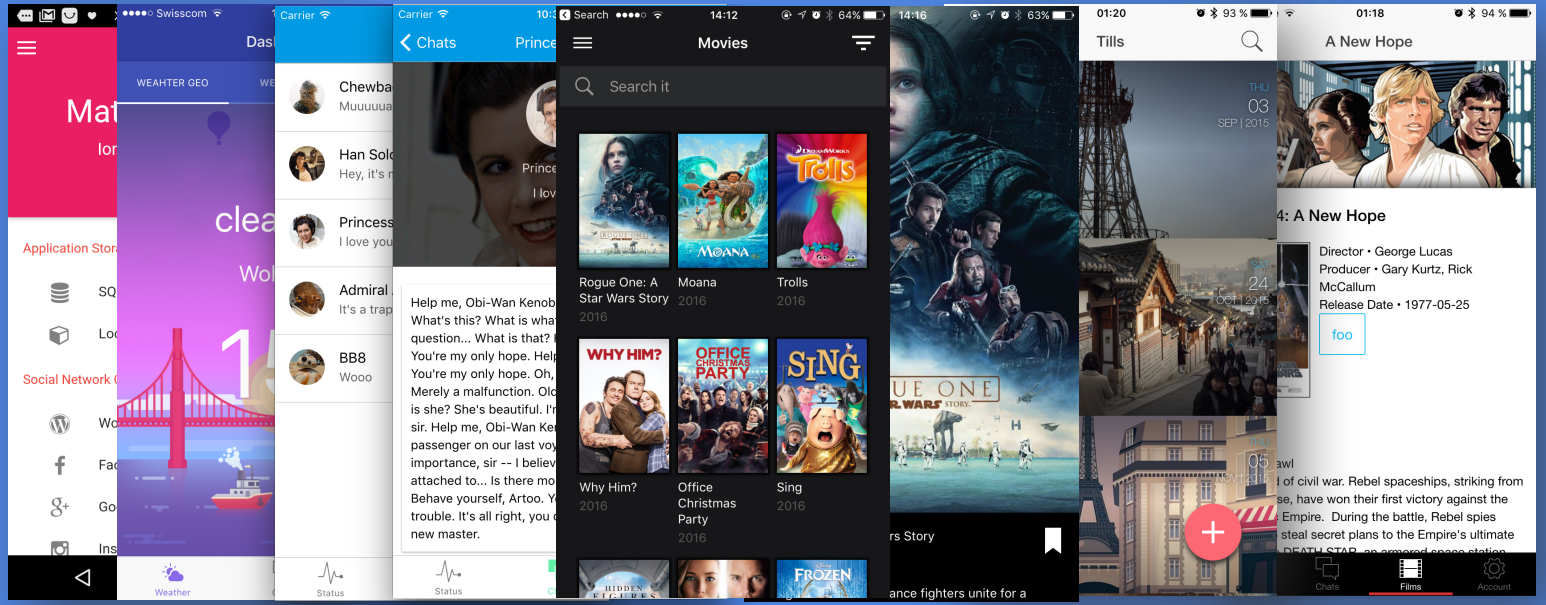
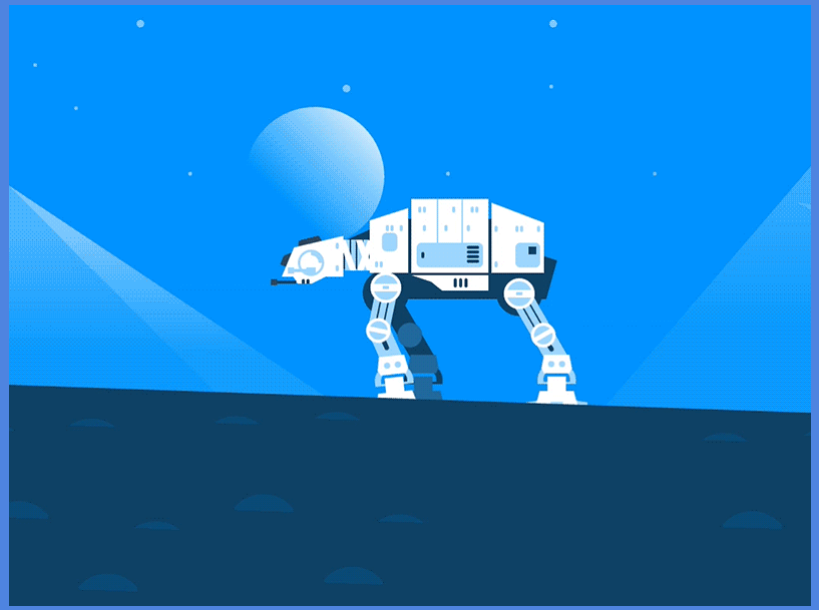
ionic

Building Hybrid Apps with Angular and Ionic

Demo Time


Swisscom
WEAHTER GEO
Weather
Carrier
Chewba
Han Solo
Princess
Admiral
BB8
Wooo
Status
Chats
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
Help me, Obi-Wan Kenob
What's this? What is wha
question... What is that?
You're my only hope. Hel
You're my only hope. Oh,
Merely a malfunction. Old
is she? She's beautiful. I'
sir. Help me, Obi-Wan Ke
passenger on our last voy
importance, sir -- I believ
attached to... Is there mo
Behave yourself, Artoo. Y
trouble. It's all right, you
new master.
14:16
63%
01:20
93 %
Tills
THU
03
SEP | 2015
SAT
24
OCT | 2015
THU
05
01:18
94 %
A New Hope
4: A New Hope
Director • George Lucas
Producer • Gary Kurtz, Rick McCallum
Release Date • 1977-05-25
foo
Chats
Films
Account

Coding Time



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 use “the banana in a box” syntax

home.html

```
<ion-label>Name:</ion-label>
<ion-input [(ngModel)]="name"
(ngModelChange)= "showInput()"></ion-input>

<p>you are typing: {{ name }}</p>
```

home.ts

```
name: string;
showInput(){
  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 ion-button primary (**click**)="back()"> Click Me </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.

  *<button ion-button primary (**tap**)="viewItem(this.movie)"> Tap Me </button>*

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** (click)="viewItem(this.movie)">*

Generate a new page

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

  ionic generate page myView

- This will generate the html, ts and scss files for you
  - app/pages/my-page/my-view.html
  - app/pages/my-page/my-view.ts
  - app/pages/my-page/my-view.module.ts (for lazy loading)
  - app/pages/my-page/my-view.scss

Import a new page

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

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

Push & Pop Pages (1)

- To navigate to a new page make sure you have NavController imported on your starting page and injected to your constructor
  - import { NavController } from 'ionic-angular';
  - constructor(public navCtrl: NavController, ...){...}
- Import the destination page into your starting page
  - import { MyDestinationPage } from './../my-destination/my-destination';

Push & Pop Pages (2)

- Push to destination page from starting page

myDestination.ts

```
itemTapped(event, item) {
  this.navCtrl.push(MyDestinationPage, {
        item: item
  });
}
```

myDestination.html

```
<button ion-button secondary
(click)="itemTapped($event, name)">Send my
Param</button>
```

Push & Pop Pages (3)

- Receive at destination page
- Make sure that NavParams and NavController are imported
  - import { …, **NavController, NavParams** } from 'ionic-angular';
  - Also add them to your Constructor
  - Create a variable to save the NavParam – exp myParam:any;
  - Bind to the transferred NavParam:

myDestination.ts

```
constructor(public navCtrl: NavController, public navParams: NavParams) {
  this.myParam = navParams.get('item');
}
```

myDestination.html

```
<h1>{{myParam}}</h1>
```

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

!If you make a http call, don‘t forget to import HttpClientModule in your app.module.ts imports list!

//Ionic 3 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 use of variables:

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

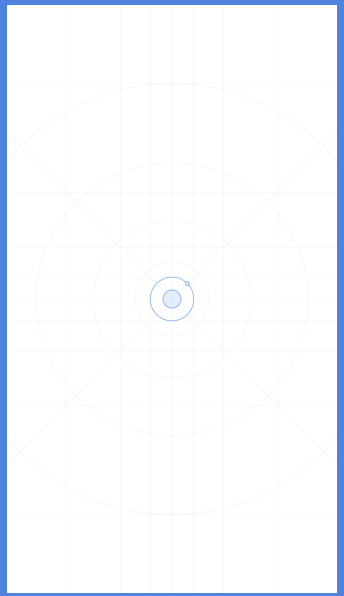
To change Ionics preset plattform variables, override them in your variables.sccs file.
You can find all variables on:
http://ionicframework.com/docs/theming/overriding-ionic-variables/

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 1014x1024px and **NO** rounded corners
- Splash min 2732x2732px

- 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