Crash Course in AngularJS + Ionic

(Deep dive)

Nolan Erck

(with assistance from Scott Steinbeck)

About Me

- Chief Consultant / Owner, South of Shasta Consulting
  - Software Development, Training, Design.
- Using ColdFusion since 1999 (4.x)
- Other stuff: C++, Java, jQuery, PHP, .NET, HTML5, Android, you get the idea...
- Manager of SacInteractive User Group.
- Reformed Video Game Developer (Grim Fandango, SimPark, StarWars Rogue Squadron, etc).
- Music Junkie.

I'M REAL PROUD OF MY RECORD COLLECTION

Today's Agenda

- Intro to MVC
- Intro to AngularJS
  - Some basic features, concepts and code samples.
- [break time]
- Intro to Ionic
  - More features, code samples, simulator, etc.
- Other resources.
- Questions.

Prerequisites

- Good understanding of “modern” JavaScript.
  - Scopes, anonymous functions, classes, objects.
- Previous MVC experience is helpful but not req'd.
- You'll also get exposed to:
  - Node.js, PhoneGap/Cordova, mobile SDKs (Android, Xcode, etc).

- And 1 more thing you need to know...

Prerequisites

- Know that...
  - Object Oriented Programming is hard.
  - This “hybrid mobile” stuff is different than what you may be used to.
  - And some of it is confusing at first.

- And that's NORMAL.

We're using Angular 1.x

- AngularJS 1.x is released.
- AngularJS 2.x is technically still in beta.
  - Moving target
  - Really different than 1.x
  - Bigger learning curve
    - Transpiling (boo, hiss)
    - TypeScript
  - Lots of smart people at this conference to ask about 2.x features.

How do I get started?

- Easy!
- https://angularjs.org/
- Click “download”.
  - (npm and bower options available too)
- Angular.js is all you need to start.
  - (Additional JS files for fancier features. We'll get to that later.)

How do I get started?

- <script src=”angular.js”></script>
- <html ng-app>
- {{variable name / expression}}
- [DEMO – Hello World]

What is MVC?

- Model View Controller
- Design Pattern
- Splits your app up into 3 logical sections.
  - Model – where your data and related logic goes.
  - View – the parts of your app you “view” (HTML, CSS).
  - Controller – determines what happens next in the app (some business logic maybe too).
- The “restaurant” analogy.

The “Model”

- Short for “data model” (kind of)
- Where your data lives
- Can be a local JS variable, localStorage, REST API, some other ajax request, etc.
- The app doesn't care
- Restaurant: the kitchen

[DEMO - Model]

The “View”

- The part of the app you “view”.
  - HTML, CSS, JavaScript

```
<html>
<body>
        this part here is the “view”
</body>
</html>
```

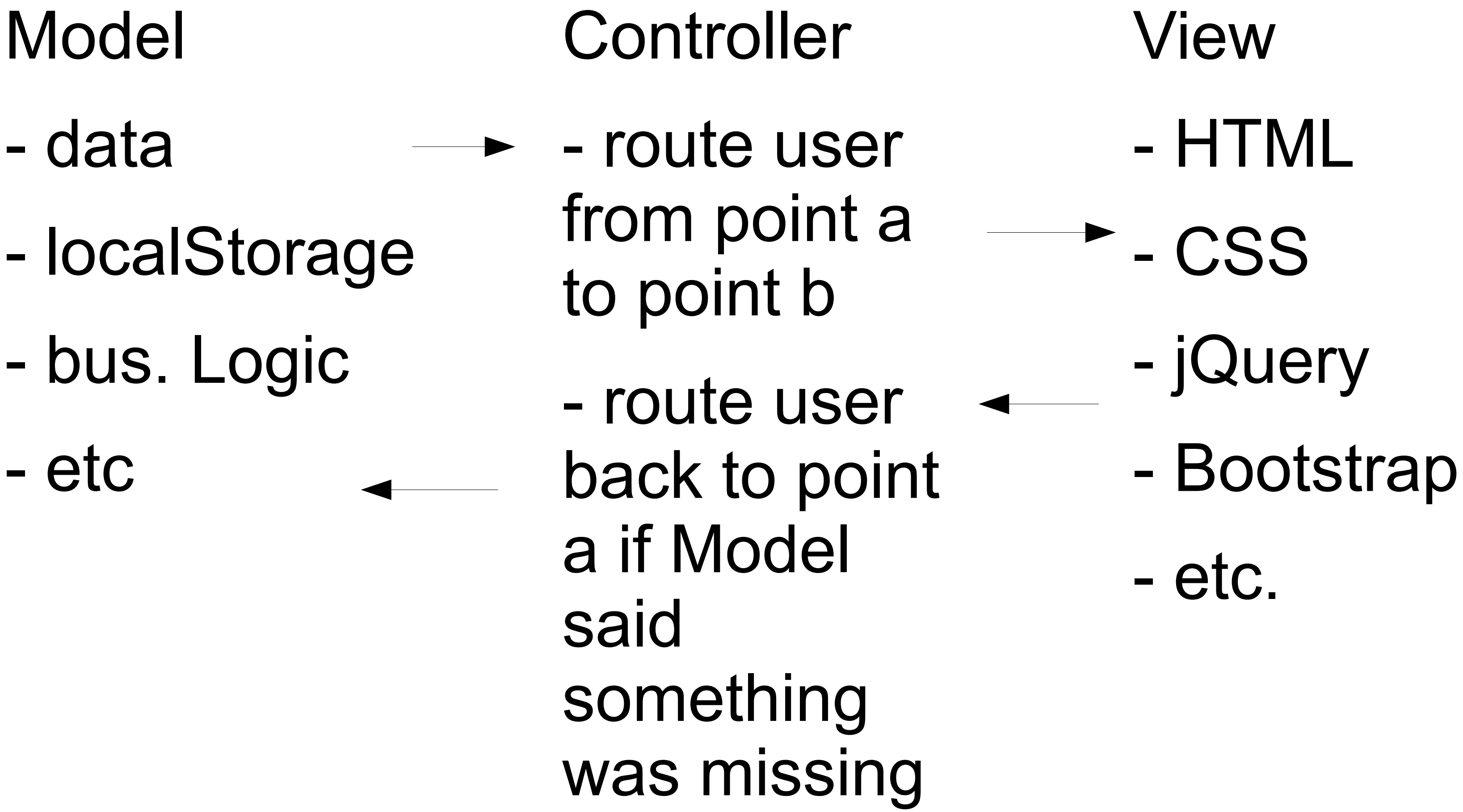
- Restaurant: the menu and plates of food

The “Controller”

- Controls what happens next.
  - Send info to the model.
  - Update the View with an error/success message.
  - Move the User to a new section of the app.
  - Etc.
- Restaurant: the waiter.
- AngularJS: $scope
- $scope is the waiter

[DEMO - Controller]

MVC – All Together



Model

- data
- localStorage
- bus. Logic
- etc

Controller

- route user from point a to point b
- route user back to point a if Model said something was missing

View

- HTML
- CSS
- jQuery
- Bootstrap
- etc.

What is AngularJS?

- MVC framework for JavaScript SPAs.
- Very few moving parts to get started.
  - Include 1 JS file, add “ng-app” to your HTML, done!
- Some nice features.
  - Dependency Injection
  - Routes
  - 2-way Data Binding
  - Various other things...

Single Page Application (SPA)

- Instead of contact.html, about.html, products.html
- You have these guys:
  - Sitename.com/#/contact
  - Sitename.com/#/products
- The whole site is routed through *index.html* via some internal machinery.
- Entire site is downloaded at once, then accessed via the browser cache.
- Only new *data* is retrieved from the server.

Looping

- In Controller / Model
  - Standard JavaScript

```
for( i = 0; i < 5; i++ ) { … }
```

  - angular.forEach()

```
var values = { name: 'Martin Gore', instrument: 'guitar' };
var log = [];
angular.forEach(values, function(value, key) {
  this.push(key + ': ' + value);
} );
```

Looping

- In View
  - ng-repeat
    - <div data-ng-repeat="i in [1,2,3,4,5]">
    - <div data-ng-repeat="item in aryGroceries">

[DEMO - Looping]

Filters

- Select a subset of items from an array.
- When you don't want to loop over *all* the elements in a given array.
- Like a built-in if() statement.
  - For sorting data on the fly.

- [DEMO - Filters]

Routes

- URLs in an Angular app look like so:
  - /index.html#/home
- Everything is loaded via index.html
- Each #section as it's own “view” that's dynamically injected as the app is running.

[DEMO - Routes]

Getting data via AJAX

- $http.get() method
  - Performs HTTP GET to retrieve data in JSON form.
  - Could be a local file, REST, anything that returns JSON.

[DEMO - JSONdata]

Custom Directives

- Basically custom tags written in AngularJS.
  - <my-custom-widget />
- Works just like regular HTML tags.
- Similar to Polymer.
- Like extended tags in HTML5, or custom tags in ColdFusion.
- Can be custom “tags”, “classes” or “attributes”.

[Demo - Directives]

Services

- AKA modules, utility libraries, etc
- Wired together using Dependency Injection
- “$name” for built-in core Angular services
  - $http, $resource, etc
- “name” for user-developed services
  - MusicianService, etc

...and that's AngularJS

- That (in an extreme nutshell) is AngularJS.
  - (There are LOTS more features we didn't cover.)
- Can be used by itself for SPA web apps.
- Not mobile-specific.

- Break time, then on to Ionic...

Ionic

- Hybrid mobile app platform
- iOS and Android in 1 codebase
- JavaScript, HTML, CSS
- No Swift, Objective-C, Java (Android)

Ionic

- Uses PhoneGap (aka Cordova)
  - HTML, CSS, JavaScript gets compiled
  - Combined with a WebView control
  - Code → WebView (chrome-less browser)
  - Packaged into an app
    - iPhone binary, Android APK file.
  - The device is “running an app, that happens to be a web browser”.

Ionic

- Is (essentially) mobile-specific.
- Sits on top of PhoneGap and Angular.
- Built on Angular's “Directives” and “Services”.
- Also uses PhoneGap's “magic”.
- Gives a Bootstrap-esque set of functionality for hybrid mobile apps.

Ionic

- Performance “obsessed”.
  - Minimal DOM manipulation
  - Removed 300ms tap delay
  - Etc
- Lots of other neat features.
- We'll barely scratch the surface today.

Ionic – Getting Started

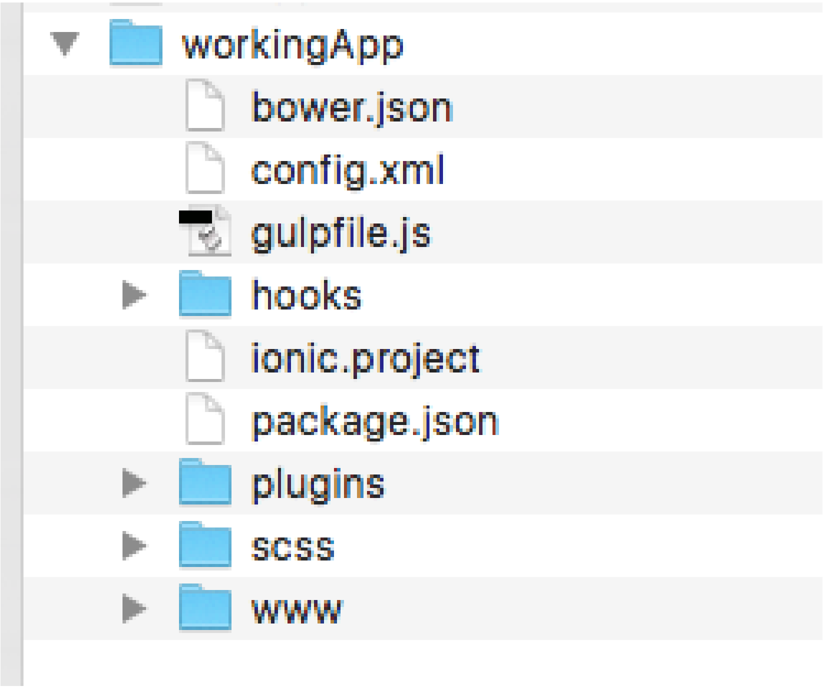
- Need Node.js, Ionic packages, Cordova packages.
- If building apps locally, need Android and iPhone SDKs (or use PhoneGapBuild).
- Initial installation can be a bit of a pain.
  - Especially if you don't run bleeding edge OS, dev tools, etc.

Ionic – Getting Started

- To start:
  - ionic start my-app [blank]
  - Generates all files needed to start an app.
  - “blank” is the template name. Several to pick from, we're starting with a bare-bones app.

    [DEMO 1]
  - ionic start my-app tabs
  - [DEMO 2]
  - ionic start my-app sidemenu
  - [DEMO 3]

Project Structure



```
▼ workingApp
    bower.json
    config.xml
    gulpfile.js
  ▶ hooks
    ionic.project
    package.json
  ▶ plugins
  ▶ scss
  ▶ www
```

Bower, Gulp, etc

- build things (optional)

/www

- main code folder

/scss

- Optional SASS things

/plugins

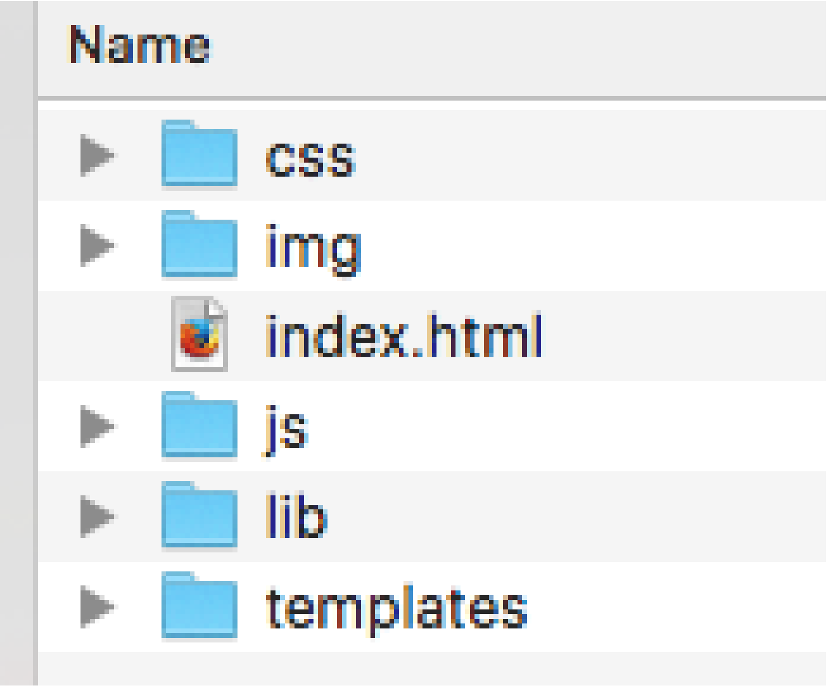
- Cordova / PhoneGap Plugins

Let's Make An App!

- Warning:
  - Live coding about to happen!
  - Dependency on the wifi about to happen!
  - Please stop streaming cat videos on YouTube for the next few minutes.
  - Thanks. :)

[DEMO ionic start DevObjective2016 tabs]

Project Structure – www folder



Name

▶ css

▶ img

index.html

▶ js

▶ lib

▶ templates

/css

- empty by default

/img

- Auto-resized by ionic build

/js

- My code for the app

/lib

- JS libraries, angular, ionic
- Ionic CSS, SVG fonts, etc

/templates

- My views for the app

Project Structure

- <ion-*> tags
- Technically Angular directives and services under the hood.
- The stuff that looks like Angular *is* Angular.
- The <ion-*> tags are the Ionic part.
- All the AngularJS you already know works the exact same way!

  [DEMO 4]

  [DEMO 5]

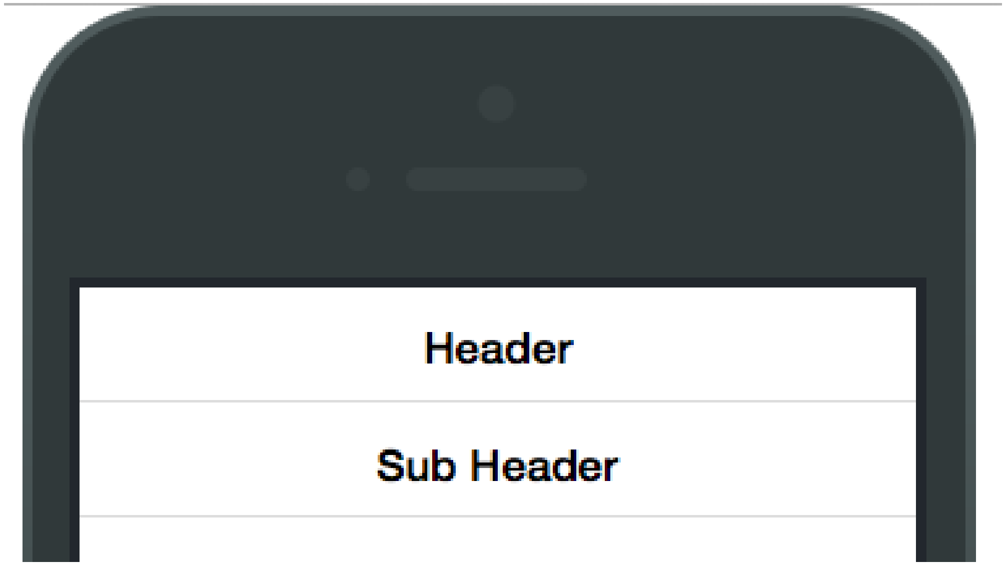
Project Structure

- Lots of CSS classes
- Very Bootstrap-esque
  - Add class or classes to HTML tags for most of the base functionality.

Code Samples

- Header and Subheader

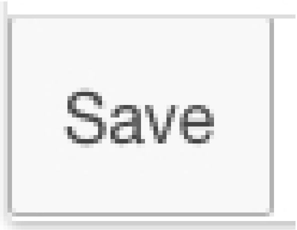
```
<div class="bar bar-header">
  <h1 class="title">Header</h1>
</div>
<div class="bar bar-subheader">
  <h2 class="title">Sub Header</h2>
</div>
```



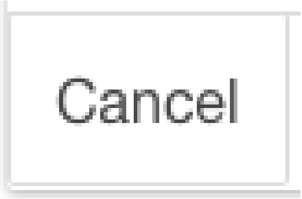
Code Samples

- Buttons

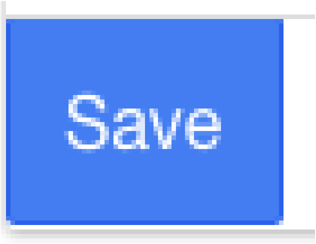
```
<button class="button"> Default </button>
```



```
<button class="button button-light"> Cancel </button>
```



```
<button class="button button-positive"> Save </button>
```



Code Samples

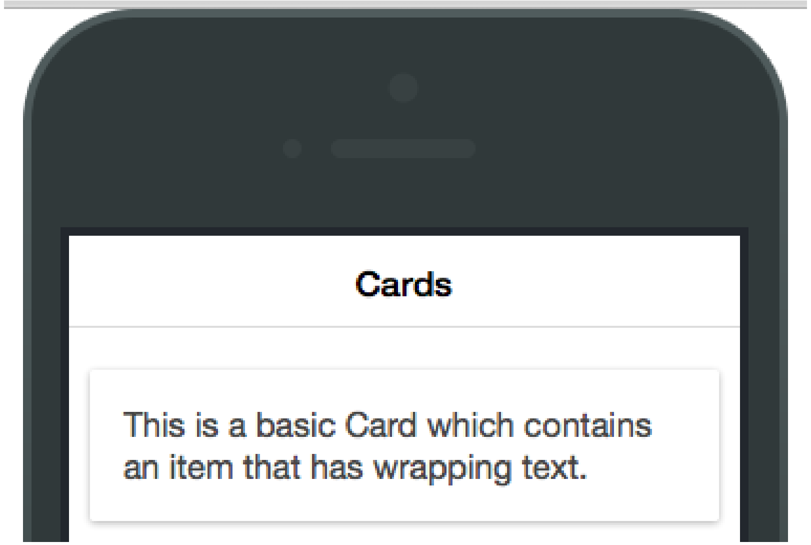
- Lists

```
<div class="list">
  <div class="item item-divider"> Guitar Players </div>
  <a class="item" href="#"> Steve Vai </a>
  <a class="item" href="#"> Eric Johnson </a>
  <a class="item" href="#"> Stanley Jordan </a>
</div>
```

Code Samples

- Cards

```
<ion-content>
  <div class="card">
    <div class="item item-text-wrap">
      Card content goes here.
    </div>
  </div>
</ion-content>
```

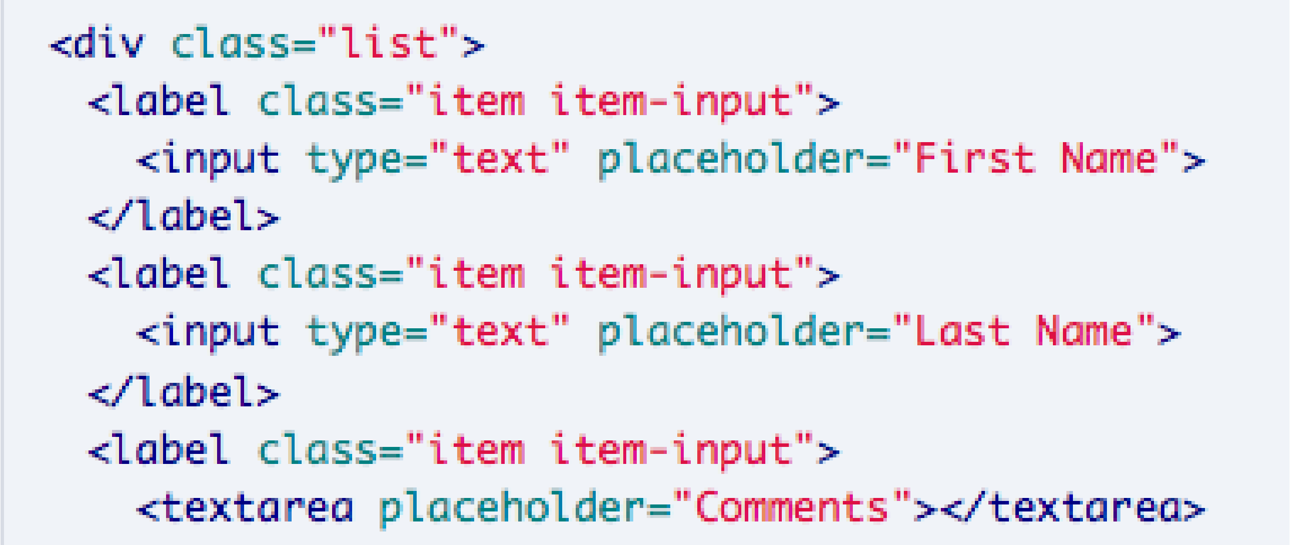


Cards

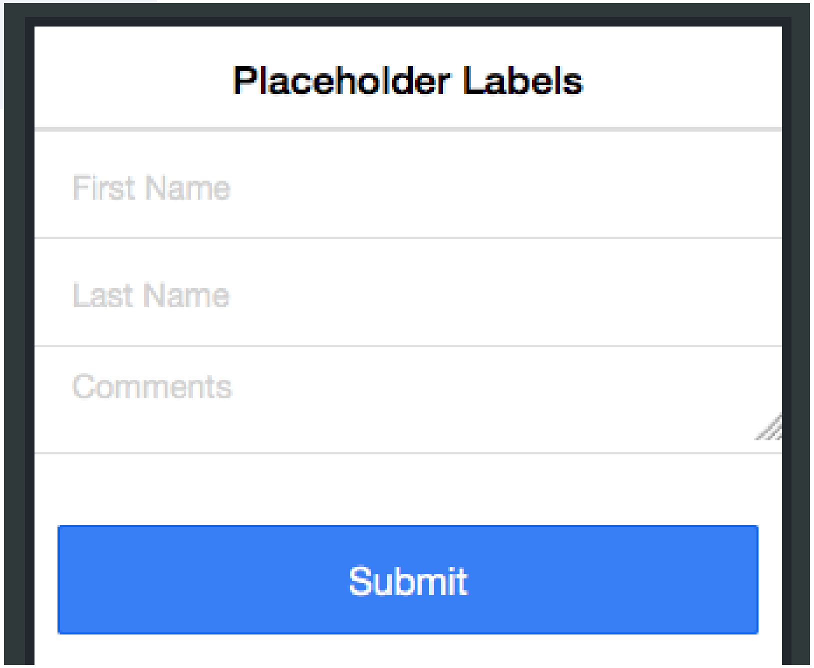
This is a basic Card which contains an item that has wrapping text.

[DEMO - cards]

Code Samples – Form Controls



```
<div class="list">
  <label class="item item-input">
    <input type="text" placeholder="First Name">
  </label>
  <label class="item item-input">
    <input type="text" placeholder="Last Name">
  </label>
  <label class="item item-input">
    <textarea placeholder="Comments"></textarea>
  </label>
</div>
```



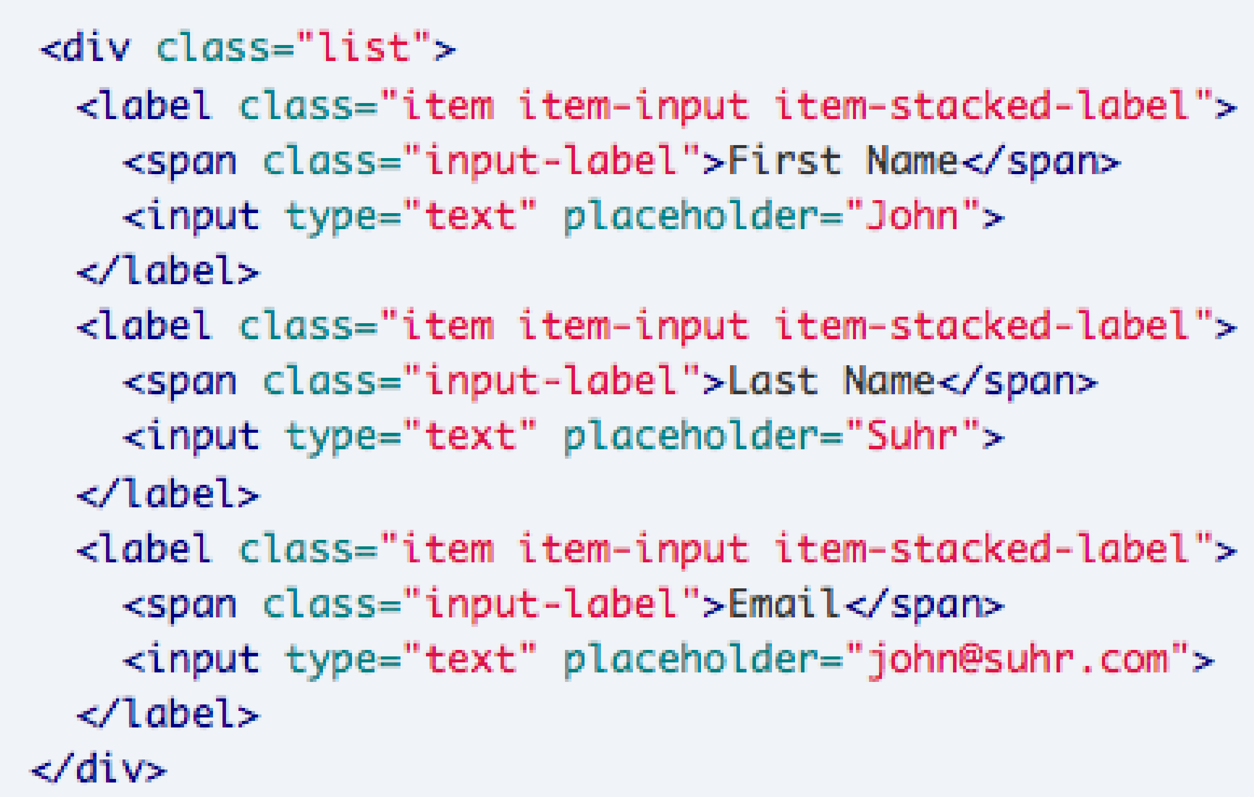
Placeholder Labels

First Name

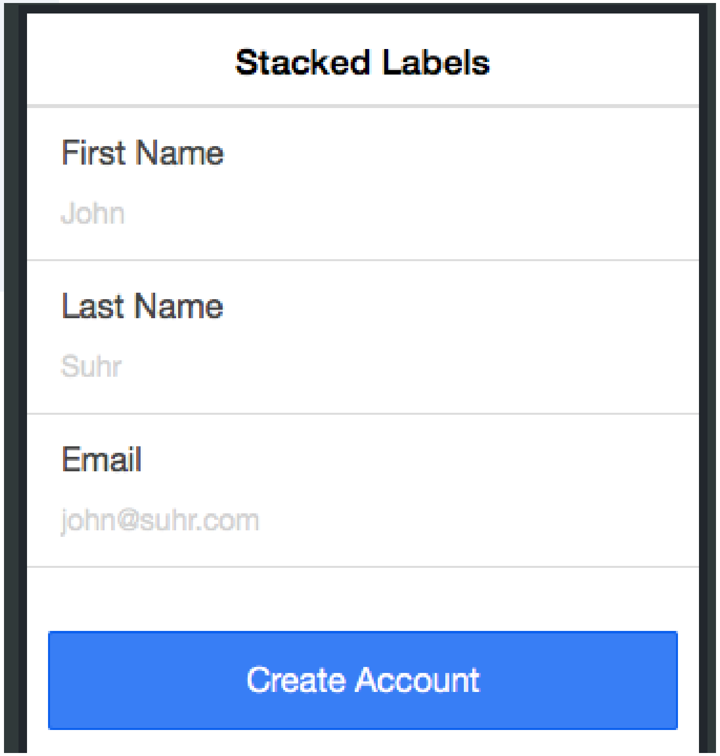
Last Name

Comments

Code Samples – Form Controls



```
<div class="list">
  <label class="item item-input item-stacked-label">
    <span class="input-label">First Name</span>
    <input type="text" placeholder="John">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Last Name</span>
    <input type="text" placeholder="Suhr">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Email</span>
    <input type="text" placeholder="john@suhr.com">
  </label>
</div>
```



Stacked Labels

First Name

John

Last Name

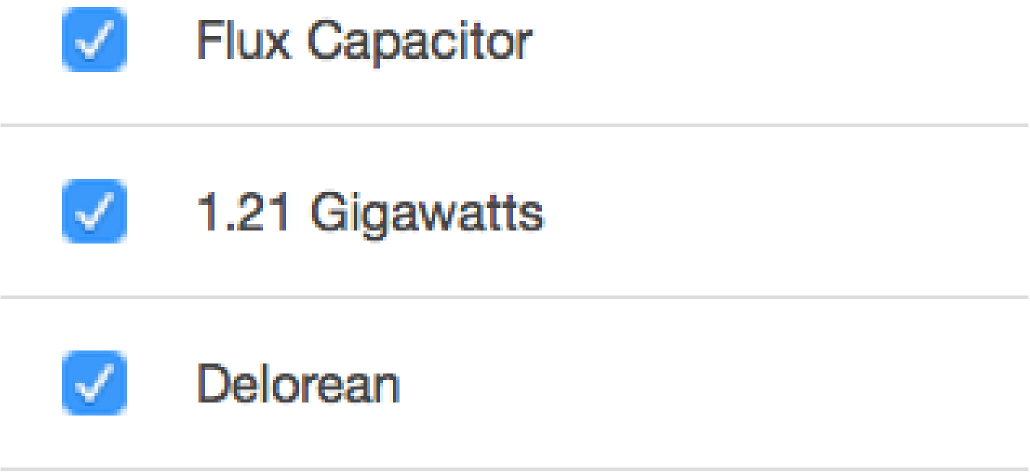
Suhr

Email

john@suhr.com

Form Controls - Checkboxes

```
<ion-list>
  <ion-checkbox ng-model="power.flux">Flux Capacitor
                </ion-checkbox>
  <ion-checkbox ng-model="power.gigawatt">
          1.21 Gigawatts</ion-checkbox>
  <ion-checkbox ng-model="power.delorean">
                Delorean</ion-checl
</ion-list>
```

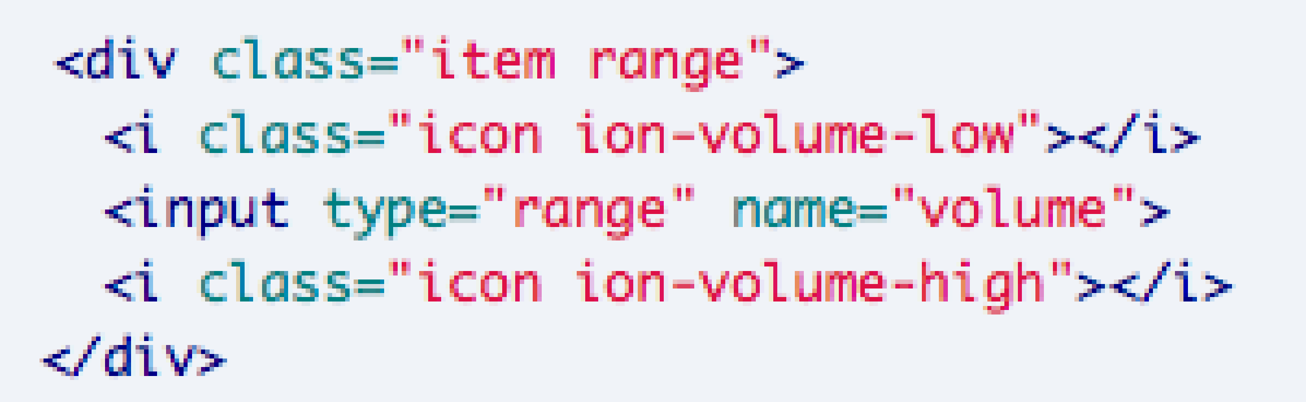


☑ Flux Capacitor

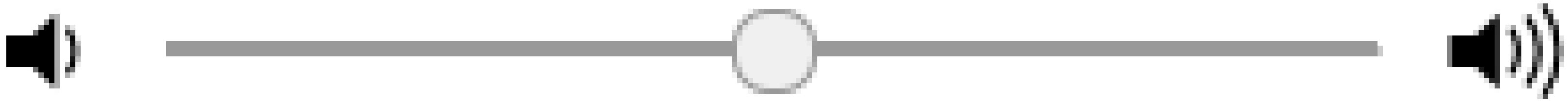
☑ 1.21 Gigawatts

☑ Delorean

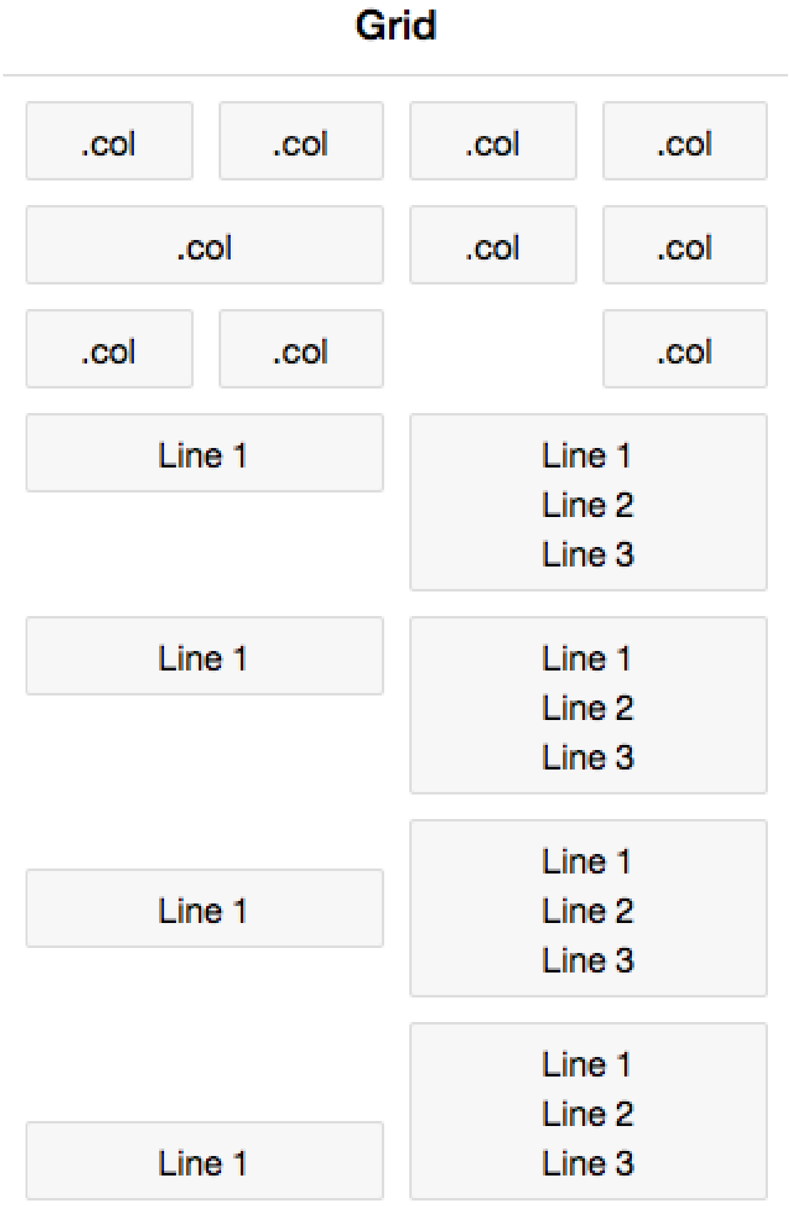
Range Control



```
<div class="item range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>
```



Grid Layout



Grid

.col .col .col .col

.col .col .col

.col .col .col

Line 1 | Line 1 Line 2 Line 3

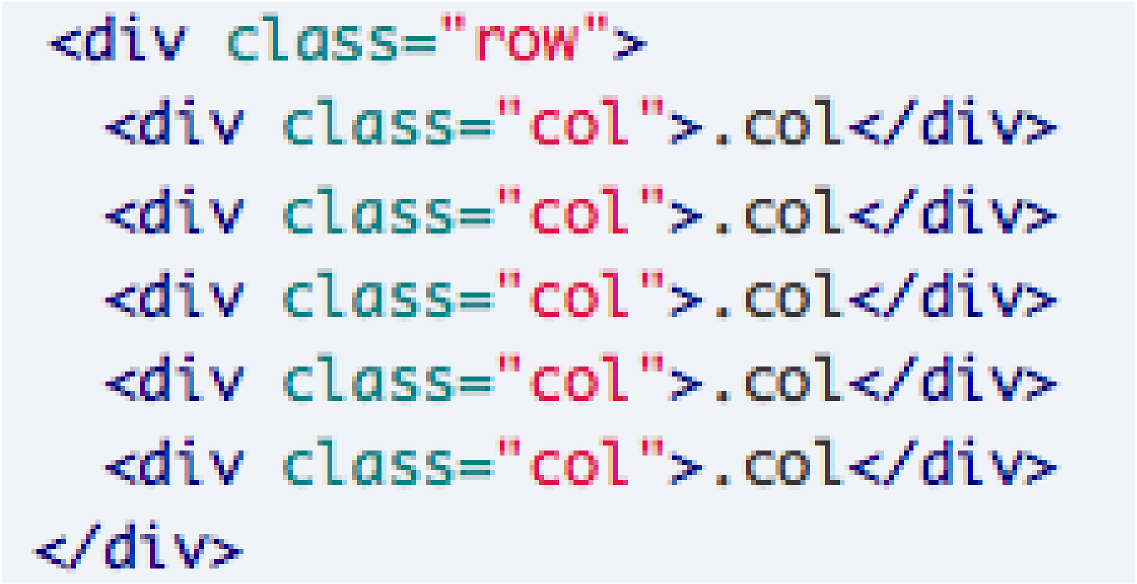
Line 1 | Line 1 Line 2 Line 3

Line 1 | Line 1 Line 2 Line 3

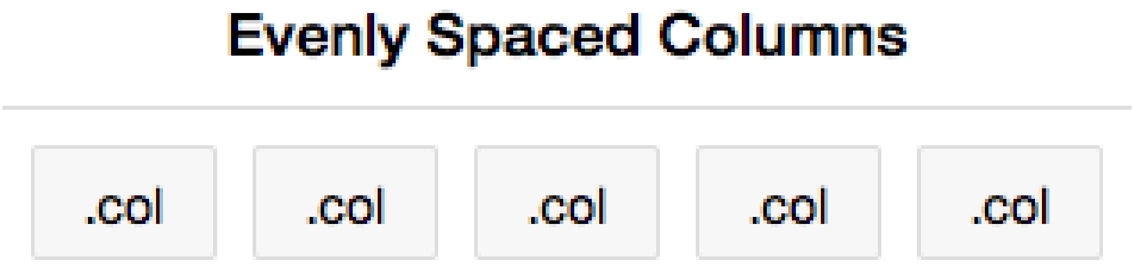
Line 1 | Line 1 Line 2 Line 3

Based on FlexBox

Add a column and Ionic will figure out the “layout math” for you.



```
<div class="row">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```



Evenly Spaced Columns

.col .col .col .col .col

And lots more!

- Utility classes
  - Colors, icons, padding
  - Platform-specific classes
    - Style iPhone differently than Android, etc
- User created add-ons
- Plugins

Testing Your App

- ionic serve
  - Runs in the browser
- ionic serve -- lab
  - Examples of iOS and Android layouts
- Code is sync'd
  - No reloading the browser needed
- ionic emulate iOS
  - Starts up the iOS simulator

Testing Your App

- Ionic View
  - http://view.ionic.io/
  - Easy private Beta testing
  - No iPhone Developer License needed
  - No “registering devices with Apple” needed
  - Install Ionic View app, add the serial #, done.

  [DEMO – Ionic View upload, install]

Publishing Your App

- Same as any other mobile app
  - Google Play Store (Android)
    - Make your APK and submit it
  - Apple App Store
    - Jump thru all of Apple's hoops
    - Subject yourself to their pain
- Build.phonegap.com
  - Helps with build process
  - Still have to deal with Apple's submission process. No way around that.

What's the catch?

- Same as any PhoneGap/Cordova project.
- Need to install the SDKs for each platform.
  - Takes time, hassles, etc.
  - Android isn't super intuitive
  - iOS requires Xcode, latest OSX, etc.

Other Resources

- AngularJS.org
- Book “Pro AngularJS”
- Book “PhoneGap Mobile Application Development Cookbook” – Matt Gifford
- IonicFramework.com
- Ionic Creator IDE
- Ionic View – easy “private QA” app
- Ray Camden's Ionic preso from CF.Objective last year.

Questions? Comments?

- Ways to reach me:
  - Southofshasta.com
  - nolan@southofshasta.com
  - Twitter @southofshasta

Free music trivia instead of swag giveaways.

You're welcome.

(Thanks again to Scott Steinbeck for assistance!)

Thanks!

Crash Course in AngularJS + Ionic

(Deep dive)

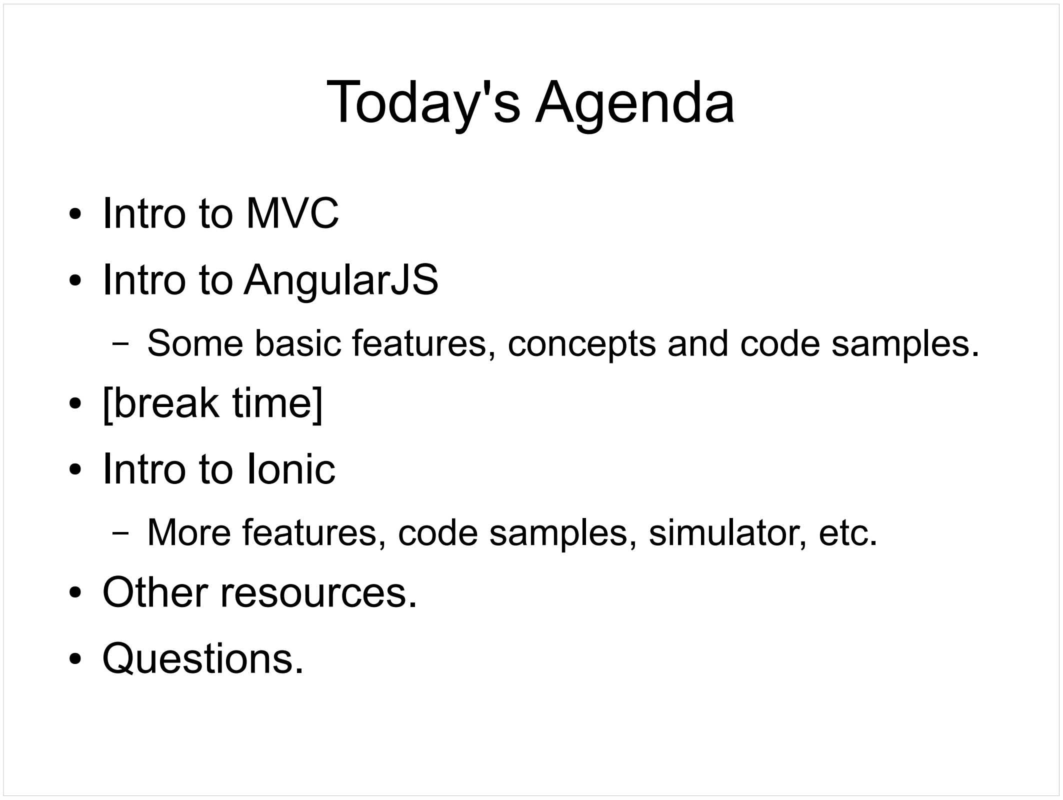
Nolan Erck

(with assistance from Scott Steinbeck)

About Me

- Chief Consultant / Owner, South of Shasta Consulting
  - Software Development, Training, Design.
- Using ColdFusion since 1999 (4.x)
- Other stuff: C++, Java, jQuery, PHP, .NET, HTML5, Android, you get the idea...
- Manager of SacInteractive User Group.
- Reformed Video Game Developer (Grim Fandango, SimPark, StarWars Rogue Squadron, etc).
- Music Junkie.

I'M REAL PROUD OF MY RECORD COLLECTION



Today's Agenda

- Intro to MVC
- Intro to AngularJS
  - Some basic features, concepts and code samples.
- [break time]
- Intro to Ionic
  - More features, code samples, simulator, etc.
- Other resources.
- Questions.

Prerequisites

- Good understanding of “modern” JavaScript.
  - Scopes, anonymous functions, classes, objects.
- Previous MVC experience is helpful but not req'd.
- You'll also get exposed to:
  - Node.js, PhoneGap/Cordova, mobile SDKs (Android, Xcode, etc).
- And 1 more thing you need to know...

Prerequisites

- Know that...
  - Object Oriented Programming is hard.
  - This “hybrid mobile” stuff is different than what you may be used to.
  - And some of it is confusing at first.

- And that's NORMAL.

We're using Angular 1.x

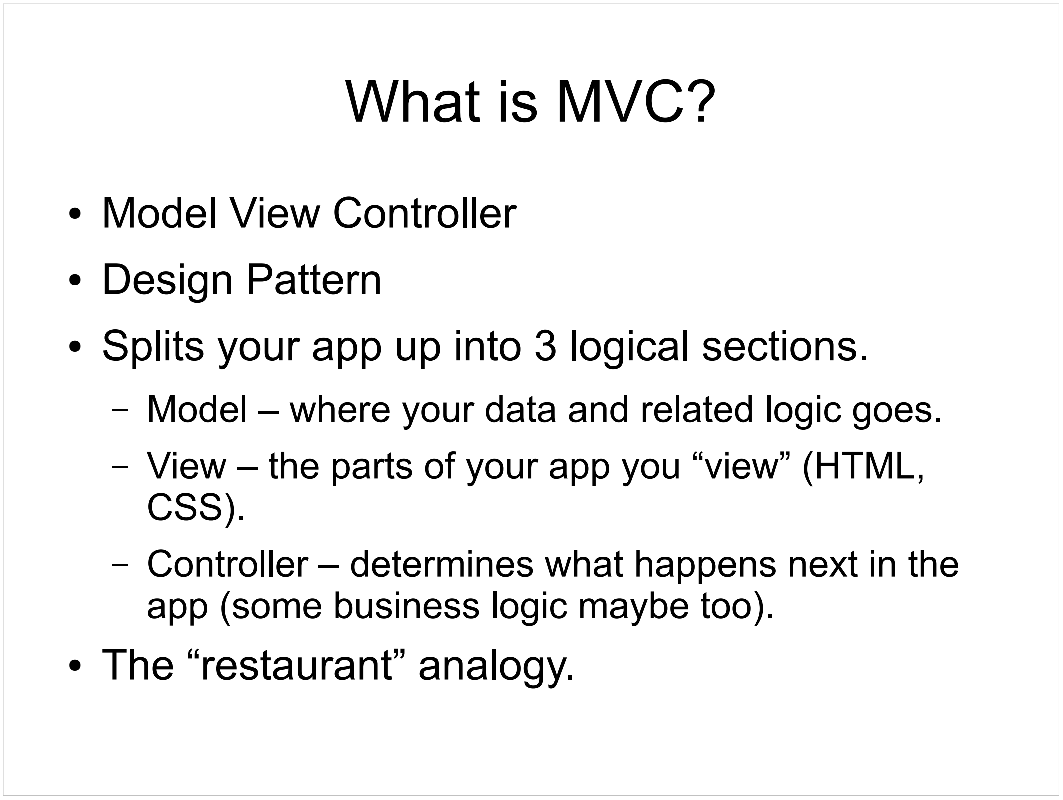
- AngularJS 1.x is released.
- AngularJS 2.x is technically still in beta.
  - Moving target
  - Really different than 1.x
  - Bigger learning curve
    - Transpiling (boo, hiss)
    - TypeScript
  - Lots of smart people at this conference to ask about 2.x features.

How do I get started?

- Easy!
- https://angularjs.org/
- Click “download”.
  - (npm and bower options available too)
- Angular.js is all you need to start.
  - (Additional JS files for fancier features. We'll get to that later.)

How do I get started?

- <script src=”angular.js”></script>
- <html ng-app>
- {{variable name / expression}}
- [DEMO – Hello World]



What is MVC?

- Model View Controller
- Design Pattern
- Splits your app up into 3 logical sections.
  - Model – where your data and related logic goes.
  - View – the parts of your app you “view” (HTML, CSS).
  - Controller – determines what happens next in the app (some business logic maybe too).
- The “restaurant” analogy.

The “Model”

- Short for “data model” (kind of)
- Where your data lives
- Can be a local JS variable, localStorage, REST API, some other ajax request, etc.
- The app doesn't care
- Restaurant: the kitchen

[DEMO - Model]

The “View”

- The part of the app you “view”.
  - HTML, CSS, JavaScript

```
<html>
<body>
        this part here is the “view”
</body>
</html>
```

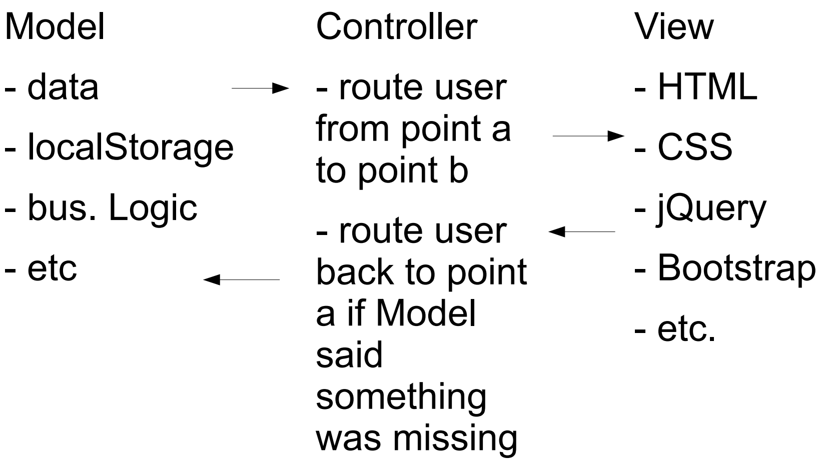
- Restaurant: the menu and plates of food

The “Controller”

- Controls what happens next.
  - Send info to the model.
  - Update the View with an error/success message.
  - Move the User to a new section of the app.
  - Etc.
- Restaurant: the waiter.
- AngularJS: $scope
- $scope is the waiter

[DEMO - Controller]

MVC – All Together



Model

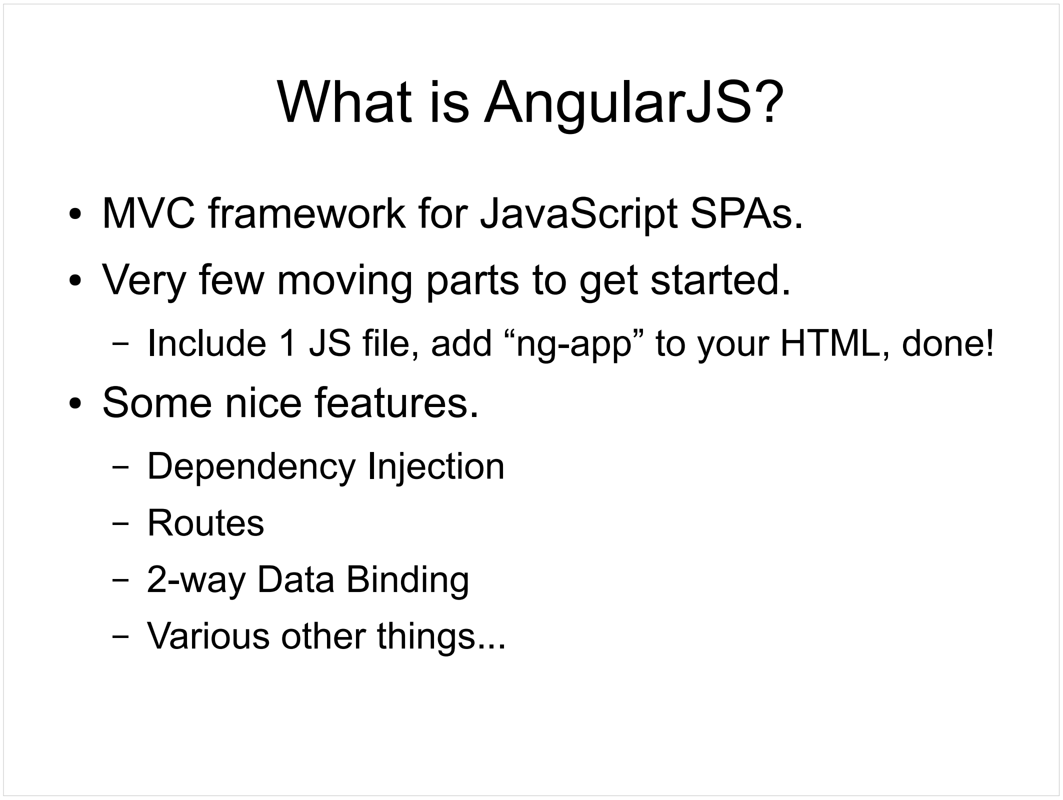
- data
- localStorage
- bus. Logic
- etc

Controller

- route user from point a to point b
- route user back to point a if Model said something was missing

View

- HTML
- CSS
- jQuery
- Bootstrap
- etc.



What is AngularJS?

- MVC framework for JavaScript SPAs.
- Very few moving parts to get started.
  - Include 1 JS file, add “ng-app” to your HTML, done!
- Some nice features.
  - Dependency Injection
  - Routes
  - 2-way Data Binding
  - Various other things...

Single Page Application (SPA)

- Instead of contact.html, about.html, products.html
- You have these guys:
  - Sitename.com/#/contact
  - Sitename.com/#/products
- The whole site is routed through *index.html* via some internal machinery.
- Entire site is downloaded at once, then accessed via the browser cache.
- Only new *data* is retrieved from the server.

Looping

- In Controller / Model
  - Standard JavaScript

```
for( i = 0; i < 5; i++ ) { … }
```

  - angular.forEach()

```
var values = { name: 'Martin Gore', instrument: 'guitar' };
var log = [];
angular.forEach(values, function(value, key) {
  this.push(key + ': ' + value);
} );
```

Looping

- In View
  - ng-repeat
    - <div data-ng-repeat="i in [1,2,3,4,5]">
    - <div data-ng-repeat="item in aryGroceries">

[DEMO - Looping]

Filters

- Select a subset of items from an array.
- When you don't want to loop over *all* the elements in a given array.
- Like a built-in if() statement.
  - For sorting data on the fly.
- [DEMO - Filters]

Routes

- URLs in an Angular app look like so:
  - /index.html#/home
- Everything is loaded via index.html
- Each #section as it's own “view” that's dynamically injected as the app is running.

[DEMO - Routes]

Getting data via AJAX

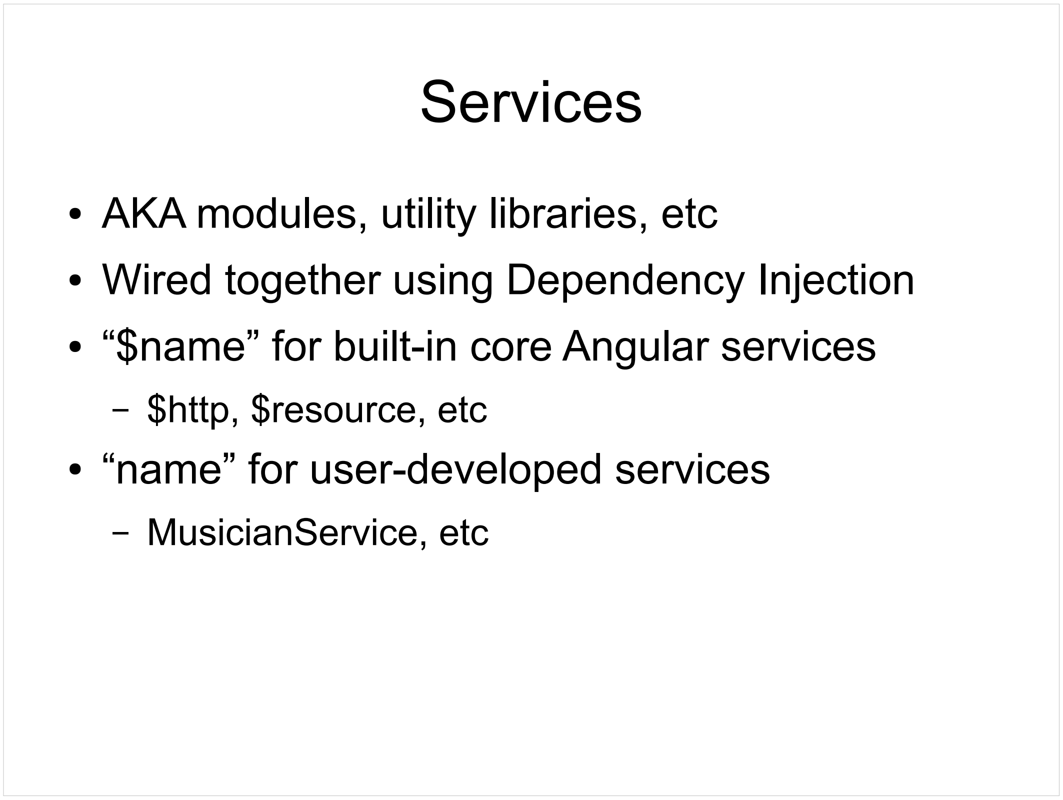
- $http.get() method
  - Performs HTTP GET to retrieve data in JSON form.
  - Could be a local file, REST, anything that returns JSON.

[DEMO - JSONdata]

Custom Directives

- Basically custom tags written in AngularJS.
  - <my-custom-widget />
- Works just like regular HTML tags.
- Similar to Polymer.
- Like extended tags in HTML5, or custom tags in ColdFusion.
- Can be custom “tags”, “classes” or “attributes”.

[Demo - Directives]



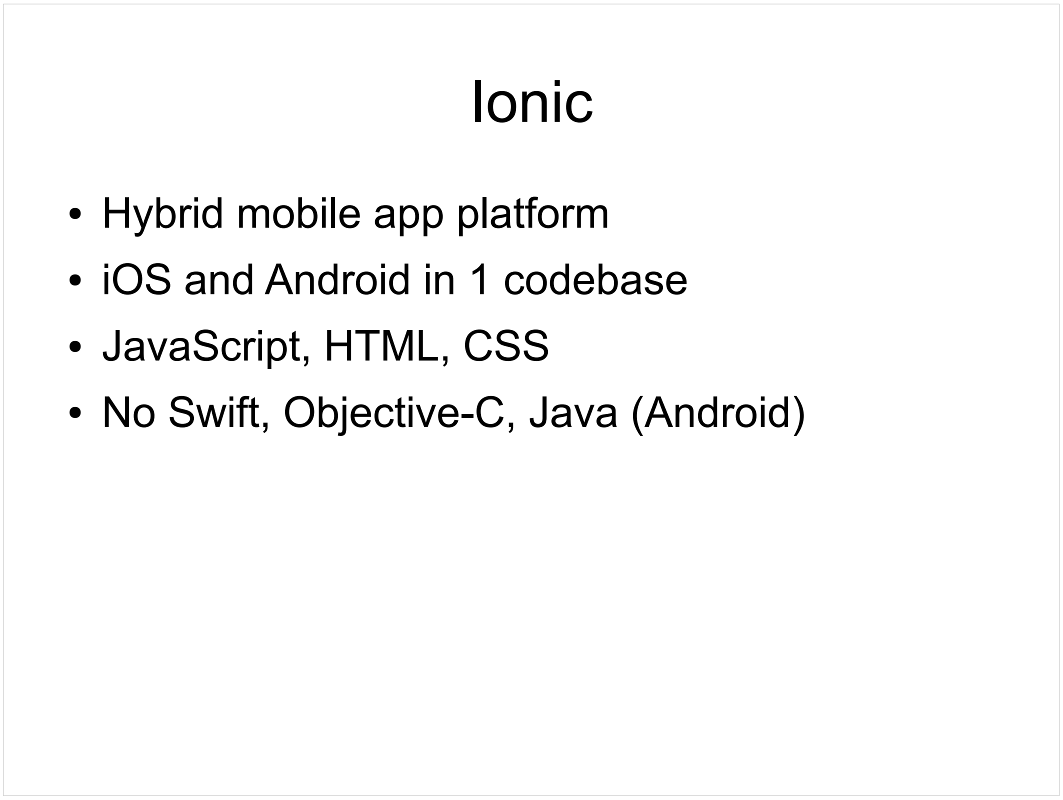
Services

- AKA modules, utility libraries, etc
- Wired together using Dependency Injection
- “$name” for built-in core Angular services
  - $http, $resource, etc
- “name” for user-developed services
  - MusicianService, etc

...and that's AngularJS

- That (in an extreme nutshell) is AngularJS.
  - (There are LOTS more features we didn't cover.)
- Can be used by itself for SPA web apps.
- Not mobile-specific.

- Break time, then on to Ionic...



Ionic

- Hybrid mobile app platform
- iOS and Android in 1 codebase
- JavaScript, HTML, CSS
- No Swift, Objective-C, Java (Android)

Ionic

- Uses PhoneGap (aka Cordova)
  - HTML, CSS, JavaScript gets compiled
  - Combined with a WebView control
  - Code → WebView (chrome-less browser)
  - Packaged into an app
    - iPhone binary, Android APK file.
  - The device is “running an app, that happens to be a web browser”.

Ionic

- Is (essentially) mobile-specific.
- Sits on top of PhoneGap and Angular.
- Built on Angular's “Directives” and “Services”.
- Also uses PhoneGap's “magic”.
- Gives a Bootstrap-esque set of functionality for hybrid mobile apps.

Ionic

- Performance “obsessed”.
  - Minimal DOM manipulation
  - Removed 300ms tap delay
  - Etc
- Lots of other neat features.
- We'll barely scratch the surface today.

Ionic – Getting Started

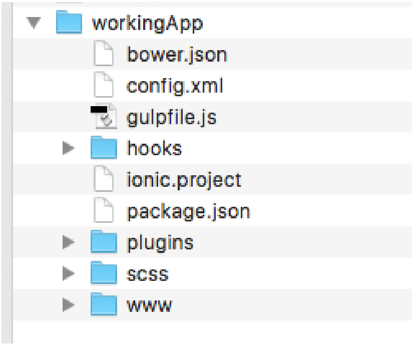
- Need Node.js, Ionic packages, Cordova packages.
- If building apps locally, need Android and iPhone SDKs (or use PhoneGapBuild).
- Initial installation can be a bit of a pain.
  - Especially if you don't run bleeding edge OS, dev tools, etc.

Ionic – Getting Started

- To start:
  - ionic start my-app [blank]
  - Generates all files needed to start an app.
  - “blank” is the template name. Several to pick from, we're starting with a bare-bones app.

    [DEMO 1]
  - ionic start my-app tabs
  - [DEMO 2]
  - ionic start my-app sidemenu
  - [DEMO 3]

Project Structure



```
▼ workingApp
    bower.json
    config.xml
    gulpfile.js
  ▶ hooks
    ionic.project
    package.json
  ▶ plugins
  ▶ scss
  ▶ www
```

Bower, Gulp, etc

- build things (optional)

/www

- main code folder

/scss

- Optional SASS things

/plugins

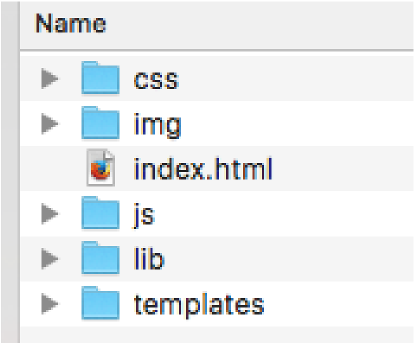
- Cordova / PhoneGap Plugins

Let's Make An App!

- Warning:
  - Live coding about to happen!
  - Dependency on the wifi about to happen!
  - Please stop streaming cat videos on YouTube for the next few minutes.
  - Thanks. :)

[DEMO ionic start DevObjective2016 tabs]

Project Structure – www folder



Name

- css
- img
- index.html
- js
- lib
- templates

/css

- empty by default

/img

- Auto-resized by ionic build

/js

- My code for the app

/lib

- JS libraries, angular, ionic
- Ionic CSS, SVG fonts, etc

/templates

- My views for the app

Project Structure

- <ion-*> tags
- Technically Angular directives and services under the hood.
- The stuff that looks like Angular *is* Angular.
- The <ion-*> tags are the Ionic part.
- All the AngularJS you already know works the exact same way!

  [DEMO 4]

  [DEMO 5]

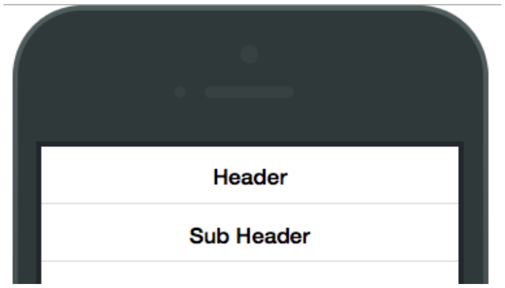
Project Structure

- Lots of CSS classes
- Very Bootstrap-esque
  - Add class or classes to HTML tags for most of the base functionality.

Code Samples

- Header and Subheader

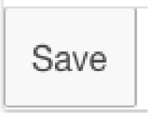
```
<div class="bar bar-header">
  <h1 class="title">Header</h1>
</div>
<div class="bar bar-subheader">
  <h2 class="title">Sub Header</h2>
</div>
```



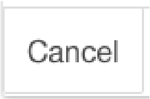
Code Samples

- Buttons

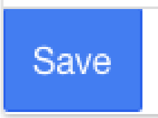
```
<button class="button"> Default </button>
```



```
<button class="button button-light"> Cancel </button>
```



```
<button class="button button-positive"> Save </button>
```



Code Samples

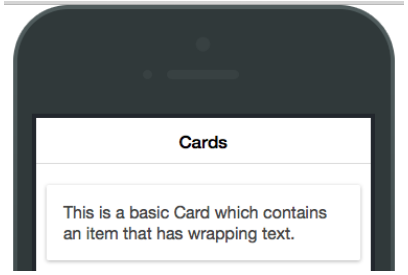
- Lists

```
<div class="list">
  <div class="item item-divider"> Guitar Players </div>
  <a class="item" href="#"> Steve Vai </a>
  <a class="item" href="#"> Eric Johnson </a>
  <a class="item" href="#"> Stanley Jordan </a>
</div>
```

Code Samples

- Cards

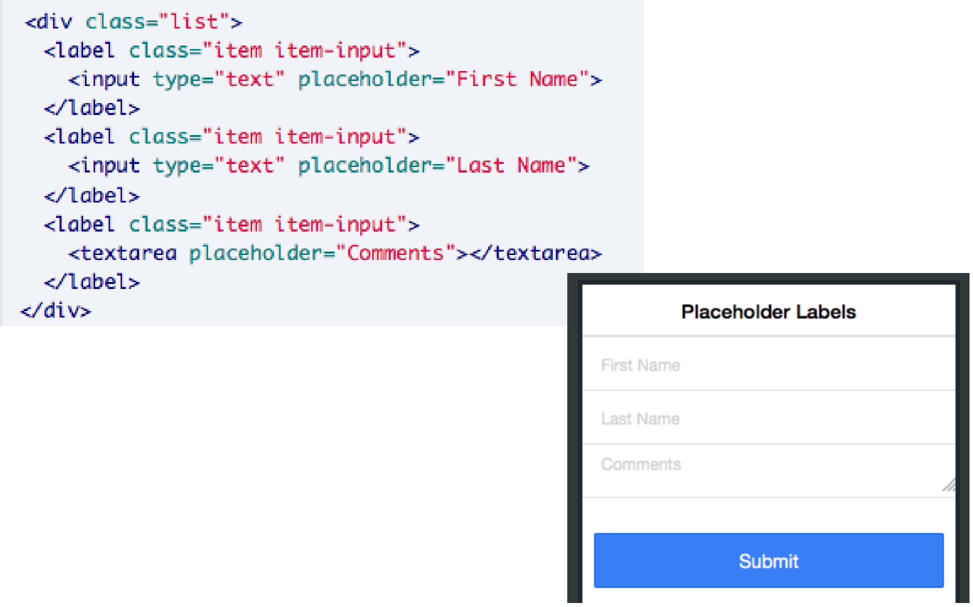
```
<ion-content>
  <div class="card">
    <div class="item item-text-wrap">
      Card content goes here.
    </div>
  </div>
</ion-content>
```



This is a basic Card which contains an item that has wrapping text.

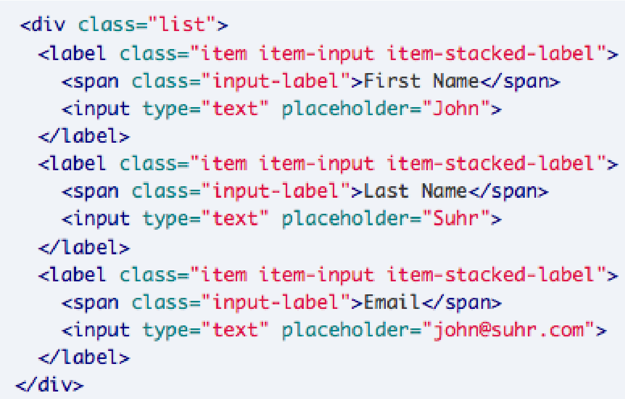
[DEMO - cards]

Code Samples – Form Controls

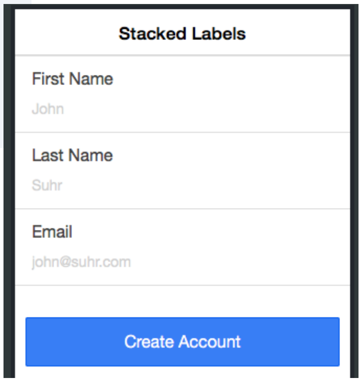


```
<div class="list">
  <label class="item item-input">
    <input type="text" placeholder="First Name">
  </label>
  <label class="item item-input">
    <input type="text" placeholder="Last Name">
  </label>
  <label class="item item-input">
    <textarea placeholder="Comments"></textarea>
  </label>
</div>
```

Code Samples – Form Controls

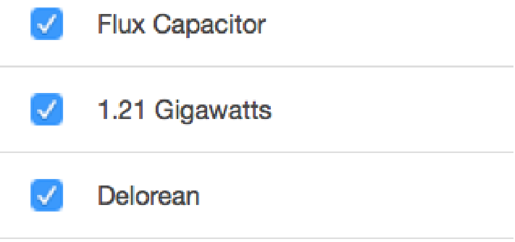


```
<div class="list">
  <label class="item item-input item-stacked-label">
    <span class="input-label">First Name</span>
    <input type="text" placeholder="John">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Last Name</span>
    <input type="text" placeholder="Suhr">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Email</span>
    <input type="text" placeholder="john@suhr.com">
  </label>
</div>
```



Form Controls - Checkboxes

```
<ion-list>
  <ion-checkbox ng-model="power.flux">Flux Capacitor
                </ion-checkbox>
  <ion-checkbox ng-model="power.gigawatt">
          1.21 Gigawatts</ion-checkbox>
  <ion-checkbox ng-model="power.delorean">
                Delorean</ion-checl
</ion-list>
```

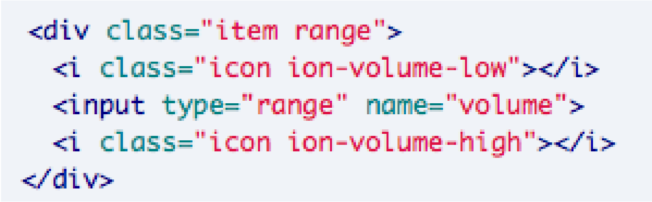


☑ Flux Capacitor

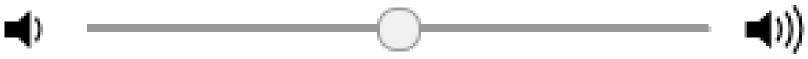
☑ 1.21 Gigawatts

☑ Delorean

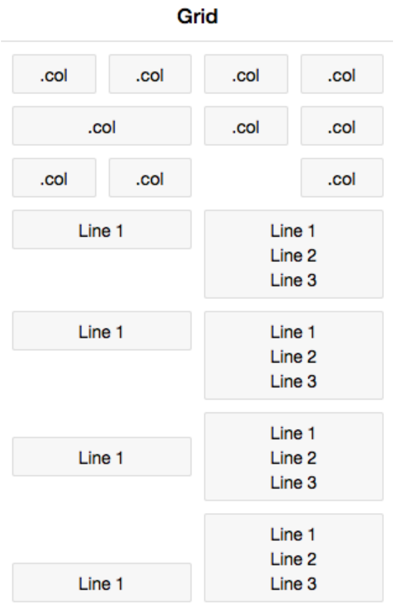
Range Control



```
<div class="item range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>
```



Grid Layout



Grid

.col .col .col .col

.col .col .col

.col .col .col

Line 1 | Line 1
Line 2
Line 3

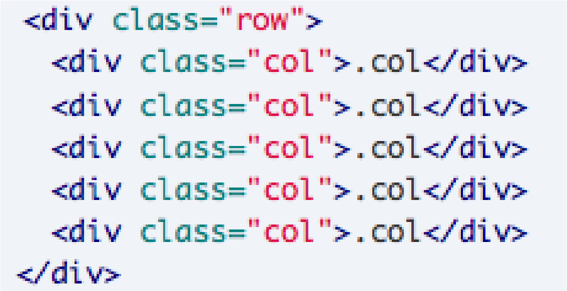
Line 1 | Line 1
Line 2
Line 3

Line 1 | Line 1
Line 2
Line 3

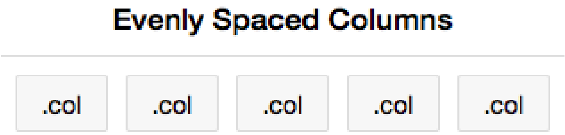
Line 1 | Line 1
Line 2
Line 3

Based on FlexBox

Add a column and Ionic will figure out the “layout math” for you.



```
 <div class="row">
   <div class="col">.col</div>
   <div class="col">.col</div>
   <div class="col">.col</div>
   <div class="col">.col</div>
   <div class="col">.col</div>
</div>
```



Evenly Spaced Columns

.col .col .col .col .col

And lots more!

- Utility classes
  - Colors, icons, padding
  - Platform-specific classes
    - Style iPhone differently than Android, etc
- User created add-ons
- Plugins

Testing Your App

- ionic serve
  - Runs in the browser
- ionic serve -- lab
  - Examples of iOS and Android layouts
- Code is sync'd
  - No reloading the browser needed
- ionic emulate iOS
  - Starts up the iOS simulator

Testing Your App

- Ionic View
  - http://view.ionic.io/
  - Easy private Beta testing
  - No iPhone Developer License needed
  - No “registering devices with Apple” needed
  - Install Ionic View app, add the serial #, done.

    [DEMO – Ionic View upload, install]

Publishing Your App

- Same as any other mobile app
  - Google Play Store (Android)
    - Make your APK and submit it
  - Apple App Store
    - Jump thru all of Apple's hoops
    - Subject yourself to their pain
- Build.phonegap.com
  - Helps with build process
  - Still have to deal with Apple's submission process. No way around that.

What's the catch?

- Same as any PhoneGap/Cordova project.
- Need to install the SDKs for each platform.
  - Takes time, hassles, etc.
  - Android isn't super intuitive
  - iOS requires Xcode, latest OSX, etc.

Other Resources

- AngularJS.org
- Book “Pro AngularJS”
- Book “PhoneGap Mobile Application Development Cookbook” – Matt Gifford
- IonicFramework.com
- Ionic Creator IDE
- Ionic View – easy “private QA” app
- Ray Camden's Ionic preso from CF.Objective last year.

Questions? Comments?

- Ways to reach me:
  - Southofshasta.com
  - nolan@southofshasta.com
  - Twitter @southofshasta

Free music trivia instead of swag giveaways.
You're welcome.

(Thanks again to Scott Steinbeck for assistance!)

Thanks!