



The Art of ANGULARJS in 2015

Matt Raible • <http://raibledesigns.com>

Photos by  McGinity Photo

 RAIBLE DESIGNS
ENTERPRISE OPEN SOURCE CONSULTING



Who is **Matt Raible**?

Father, Skier, Mountain
Biker, Whitewater Rafter

Web Framework Connoisseur

Founder of AppFuse



Blogger on raibledesigns.com

Devoxx4Kids Denver

- Teaching Kids to Program
- Java, Minecraft, robots, oh my!
- Non profit, looking for donations and speakers



www.devoxx4kids.org

<http://www.meetup.com/Devoxx4Kids-Denver/>

How to Become an Artist

Part 1 of 3: Learn the Basics on Your Own

Take some time and try various mediums of art

Recognize your strengths

Do your research and learn the basics

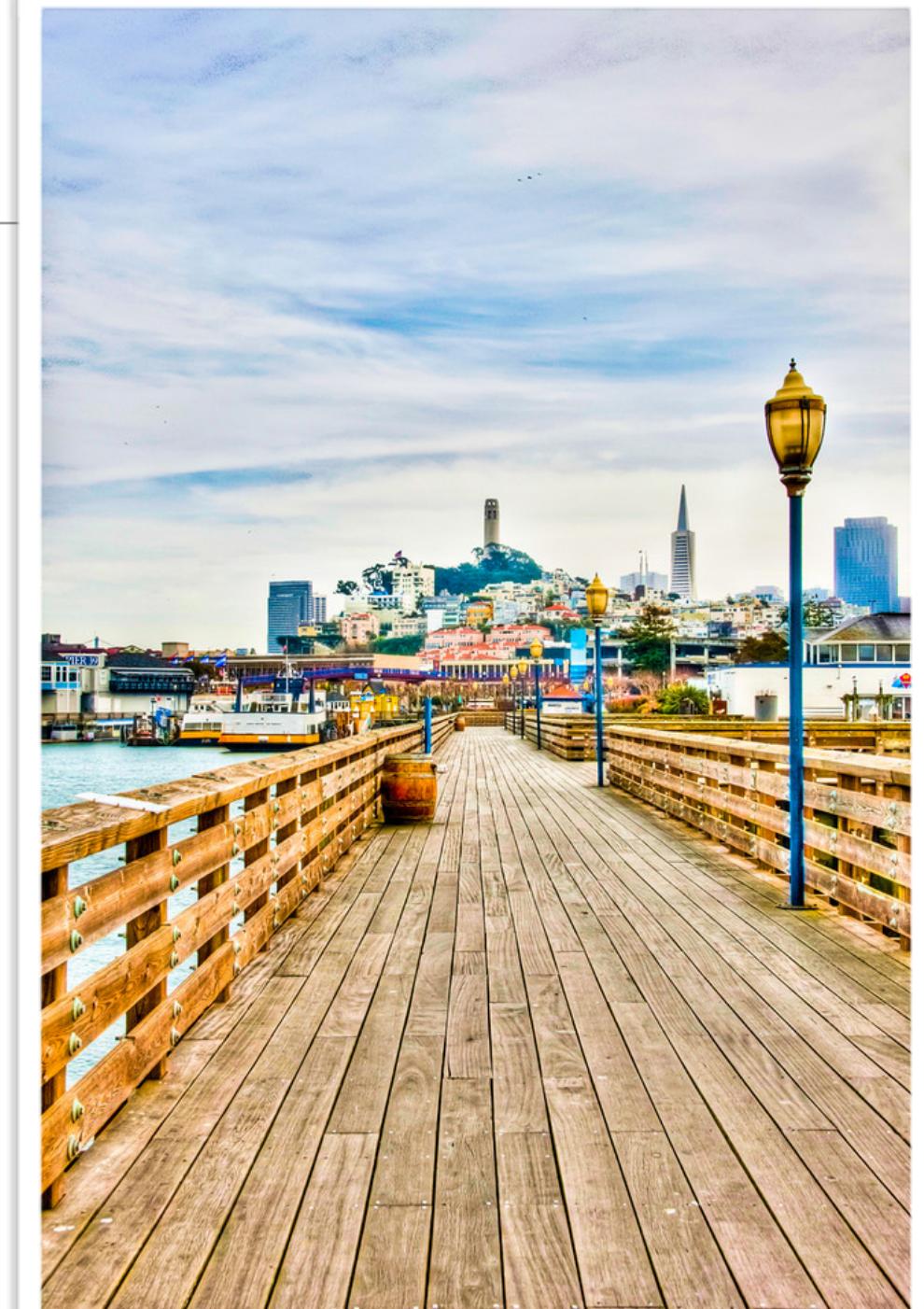
Get the supplies you will need

Observe the world around you

Make time for your art every day

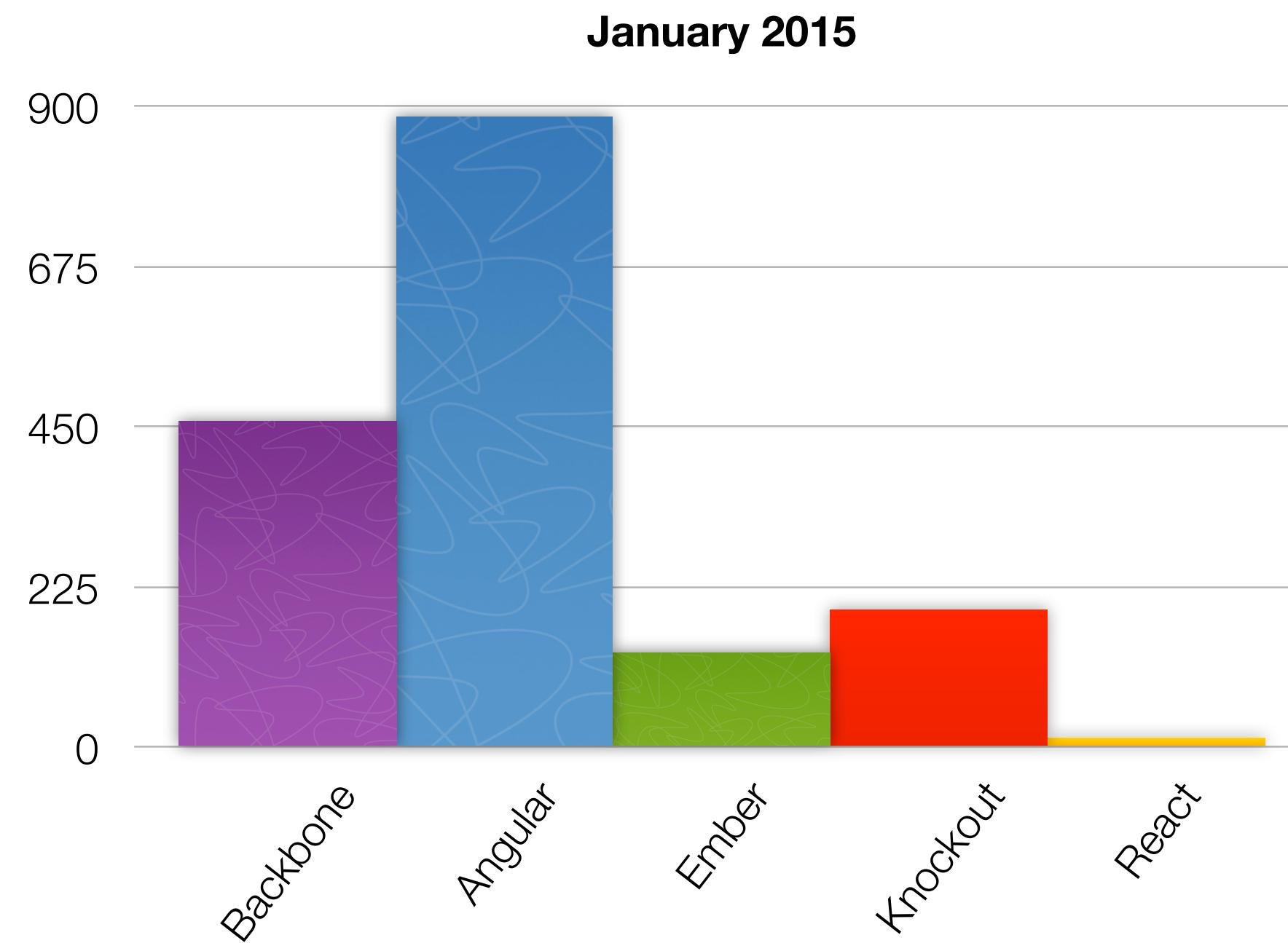
Seek out the opinions of others

Develop your own style



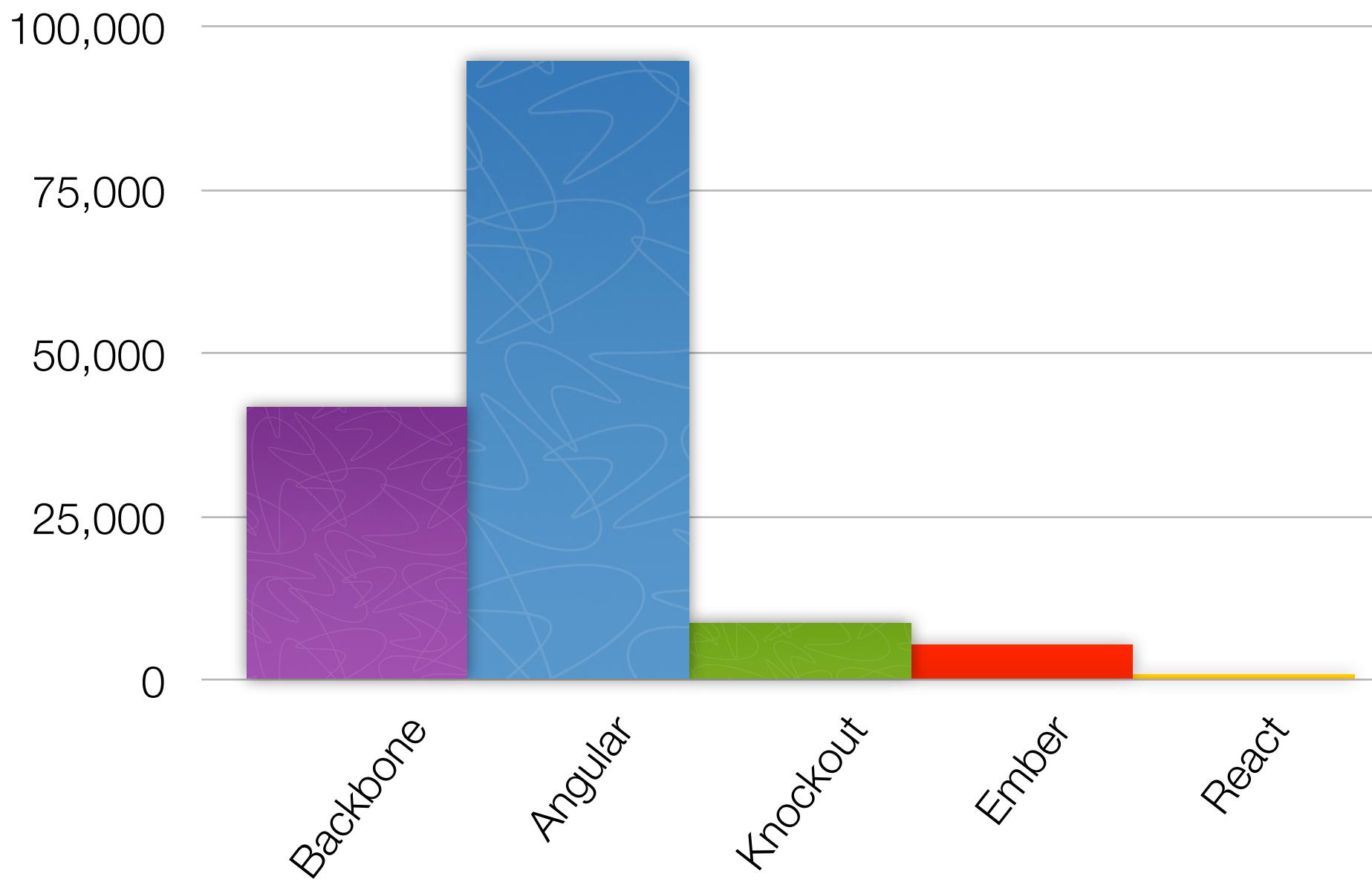
<http://www.wikihow.com/Become-an-Artist>

Jobs on Dice.com

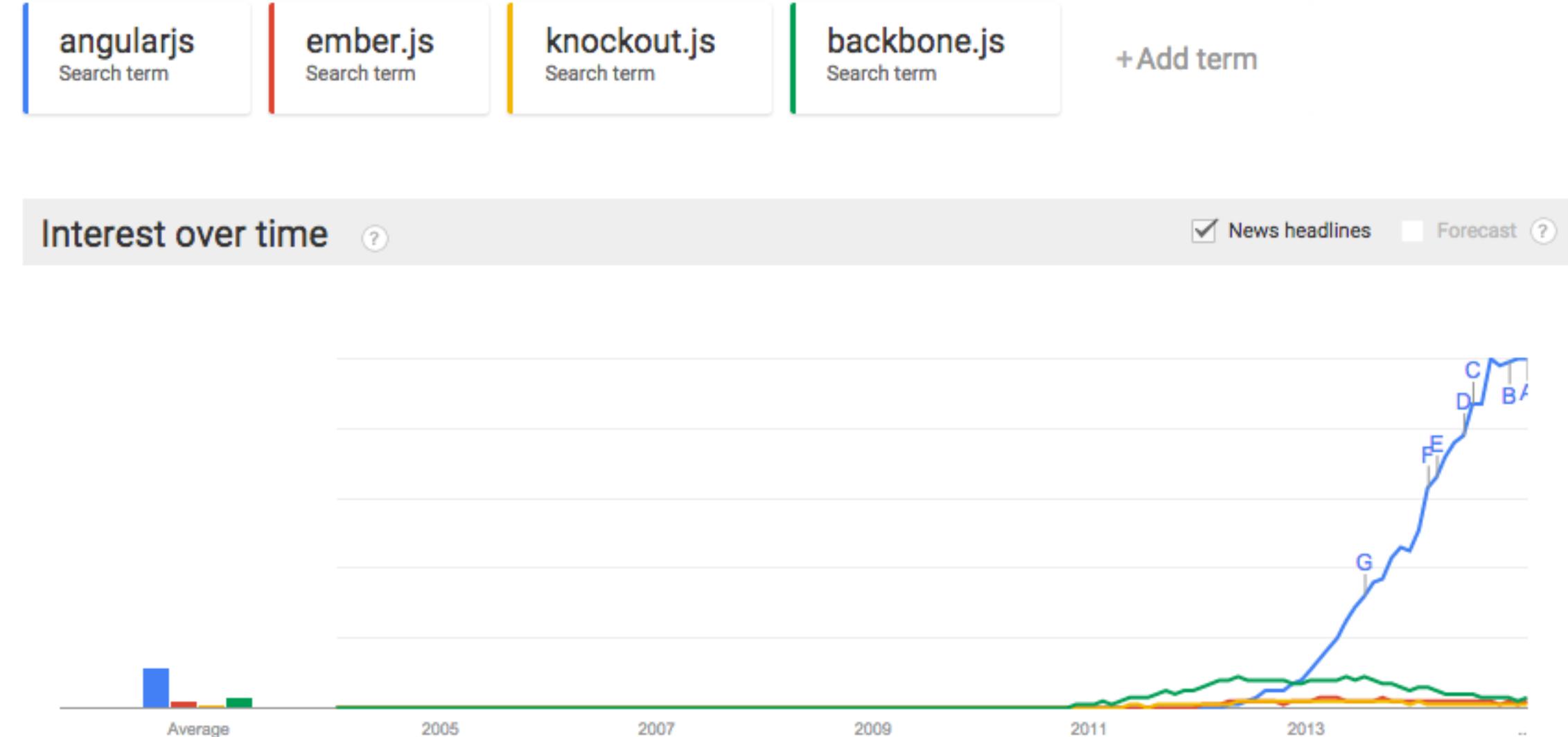


LinkedIn Skills

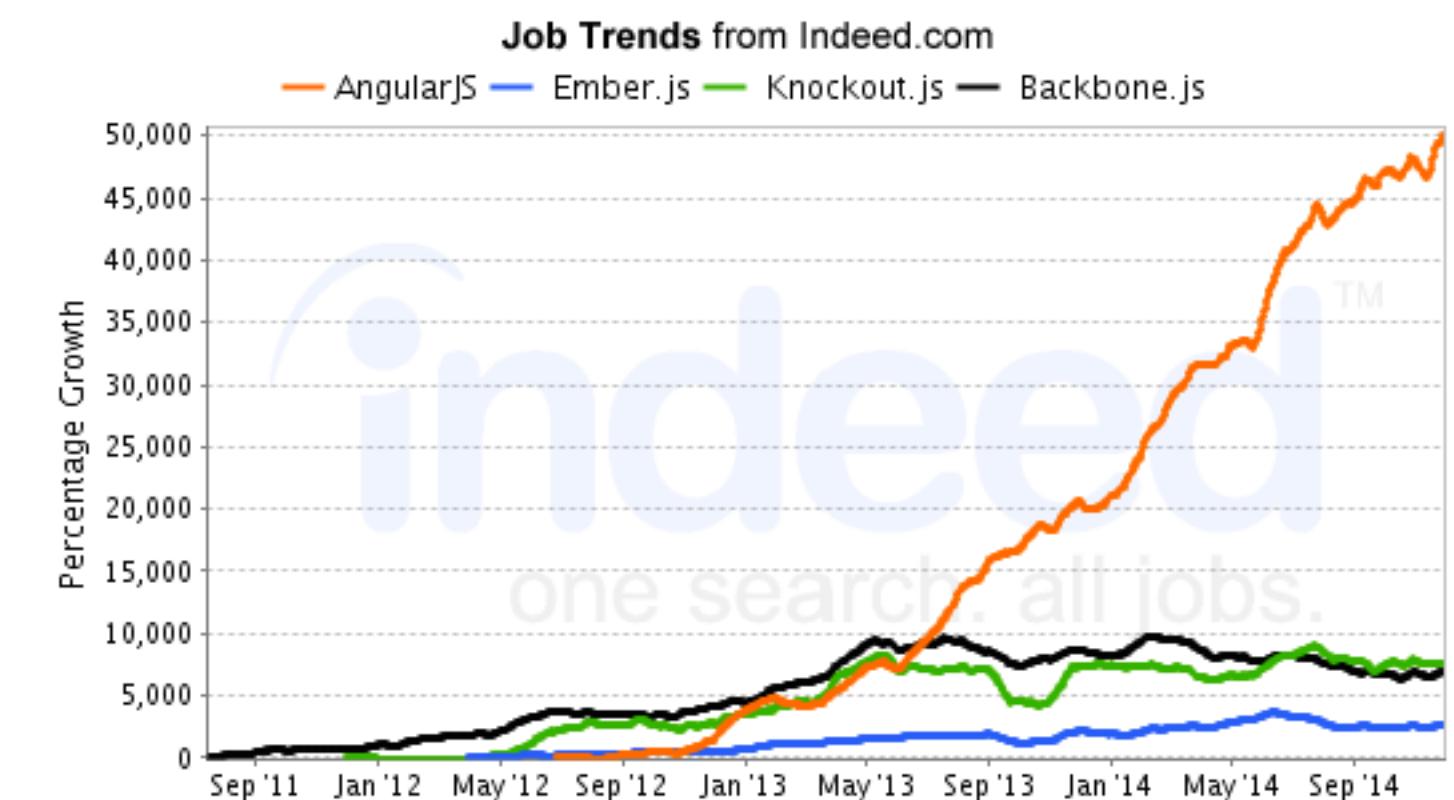
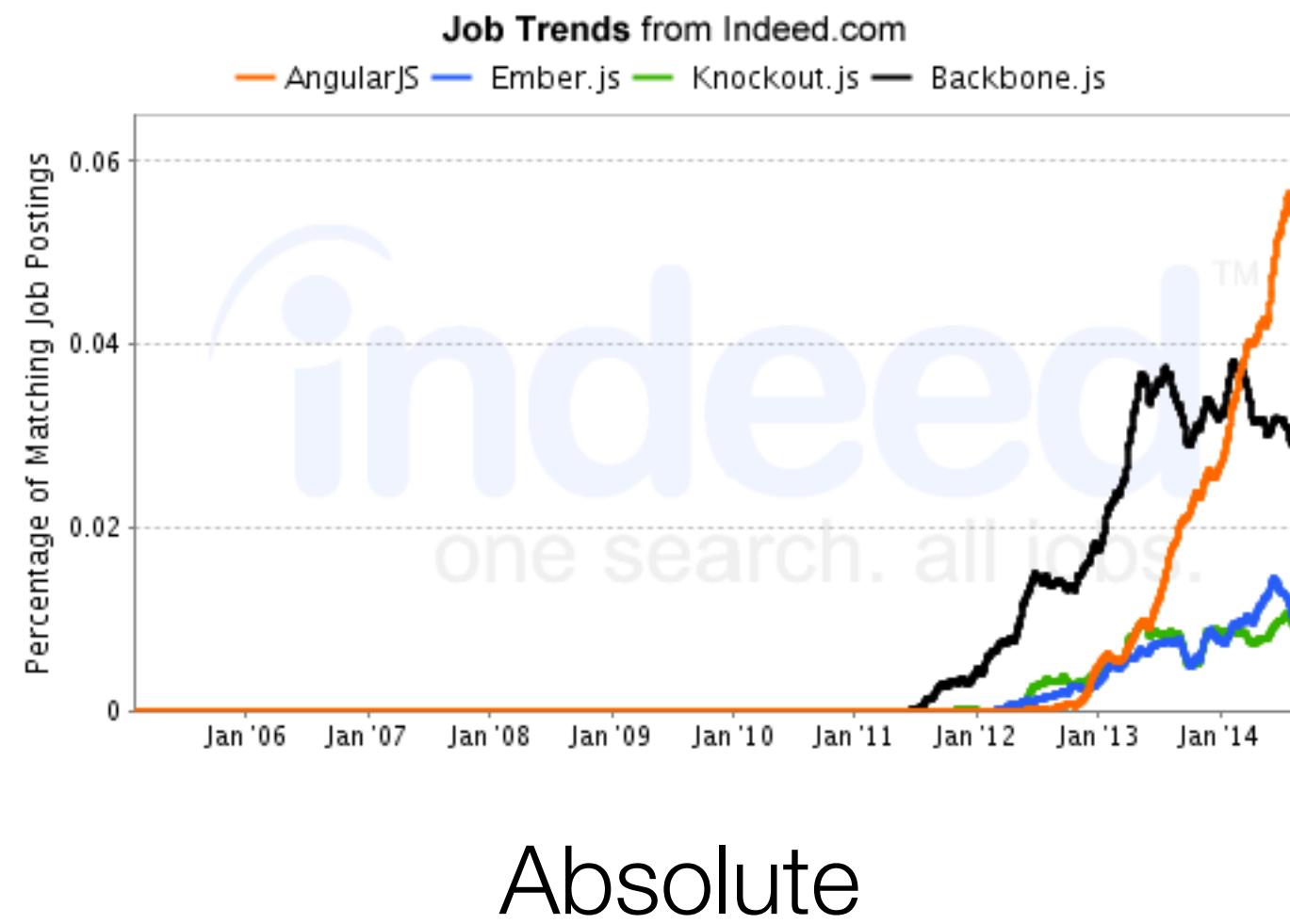
January 2015



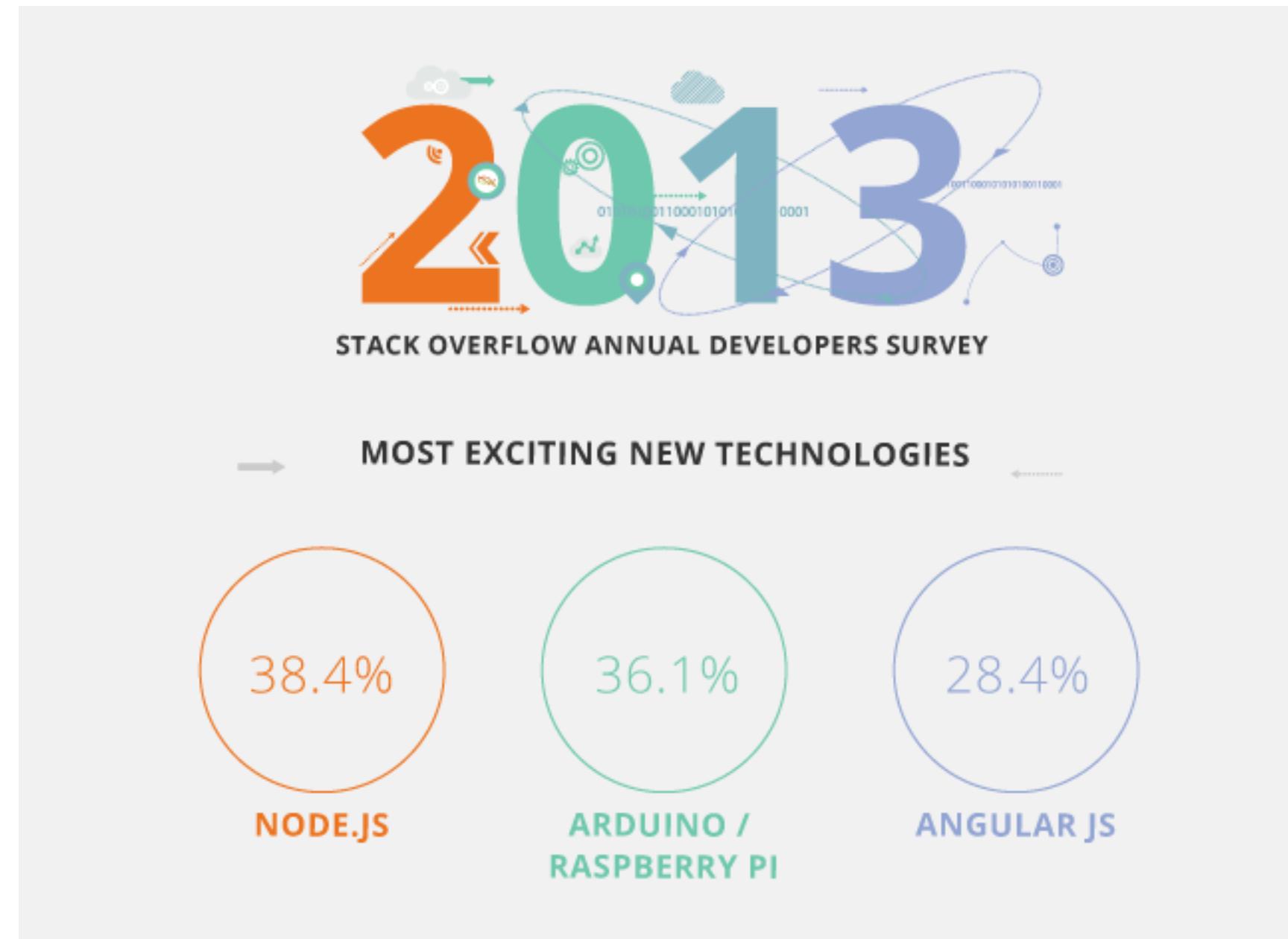
Google Trends



Indeed Job Trends



Stack Overflow





Who wants to learn  ANGULARJS?

The History of AngularJS

Started by Miško Hevery in 2009

GWT = 3 developers, 6 months

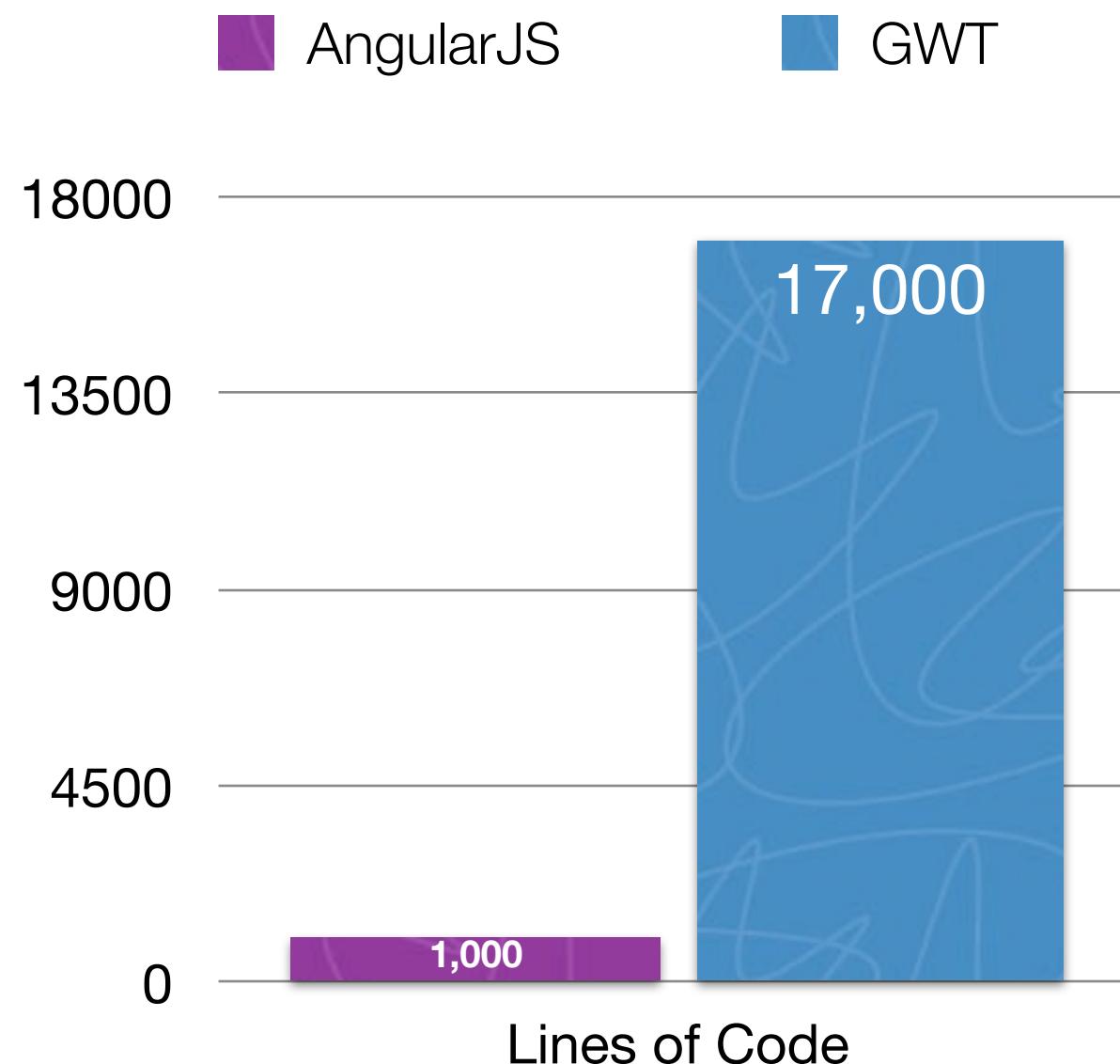
AngularJS = 1 developer, 3 weeks

Learn more:



<https://www.youtube.com/watch?v=X0VsStcCCM8>

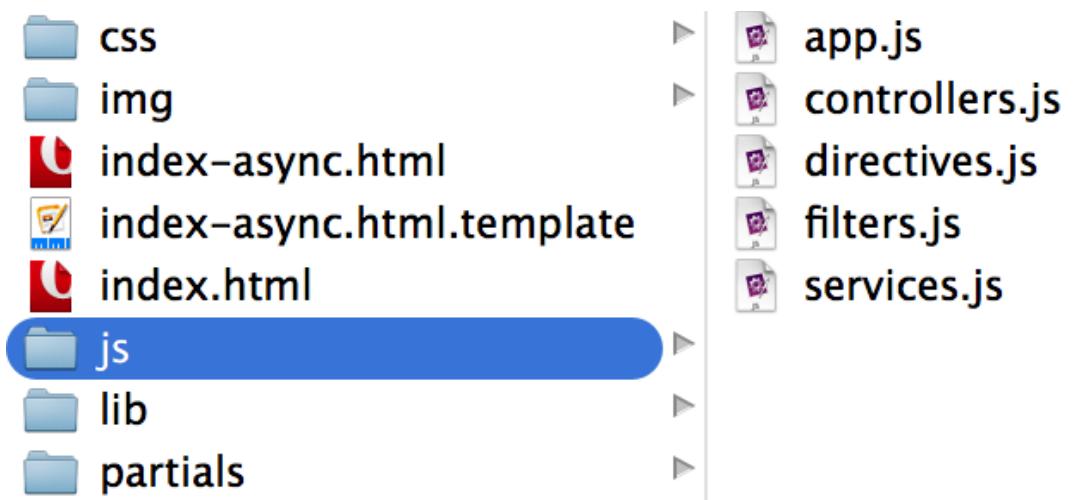
The History of AngularJS



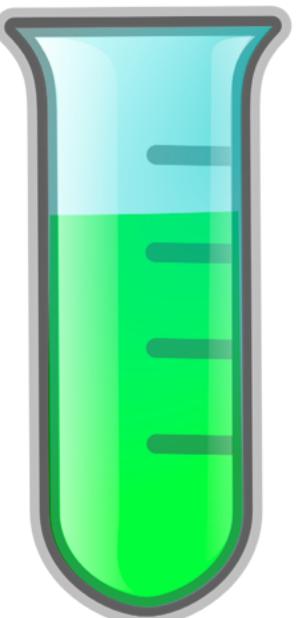
Hello World

```
<!doctype html>
<html ng-app>
<head>
  <title>Hello World</title>
</head>
<body>
<div>
  <label>Name:</label>
  <input type="text" ng-model="name" placeholder="Enter a name here">
  <hr>
  <h1>Hello {{name}}!</h1>
</div>
<script src="http://code.angularjs.org/1.3.11/angular.min.js"></script>
</body>
</html>
```

Architecture Principles



Structure



Testability



D.R.Y.

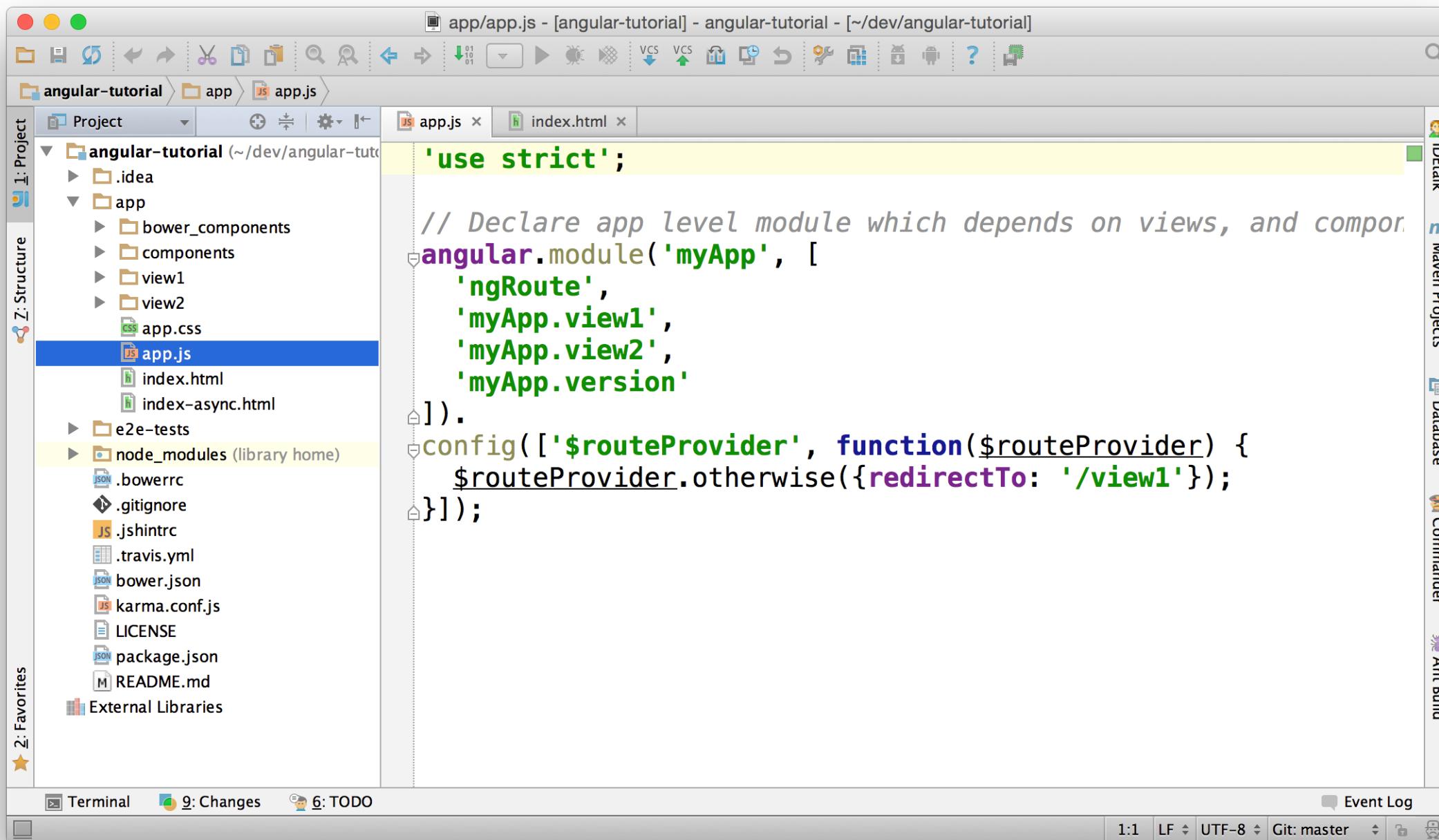
Code Organization

Start with Angular Seed*

```
git clone https://github.com/angular/angular-seed.git
```

* more options to be discussed later...

Quick Demo



The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** app/app.js - [angular-tutorial] - angular-tutorial - [/dev/angular-tutorial]
- Toolbars:** Standard Java-like toolbar with icons for file operations, search, and navigation.
- Project Structure:** On the left, the project tree shows the directory structure:
 - angular-tutorial (~/dev/angular-tutorial)
 - .idea
 - app
 - bower_components
 - components
 - view1
 - view2
 - app.css
 - app.js
 - index.html
 - index-async.html
 - e2e-tests
 - node_modules (library home)
 - .bowerrc
 - .gitignore
 - jshintrc
 - .travis.yml
 - bower.json
 - karma.conf.js
 - LICENSE
 - package.json
 - README.md
- Code Editor:** The main window displays the contents of app.js:

```
'use strict';

// Declare app level module which depends on views, and components
angular.module('myApp', [
  'ngRoute',
  'myApp.view1',
  'myApp.view2',
  'myApp.version'
]).
config(['$routeProvider', function($routeProvider) {
  $routeProvider.otherwise({redirectTo: '/view1'});
}]);
```
- Bottom Navigation:** Includes tabs for Terminal, Changes, TODO, and Event Log, along with encoding and Git status indicators.

App Definition

```
var app = angular.module('myApp', []);
```

```
<!DOCTYPE html>
<html ng-app="myApp">
```

App Definition with separate files

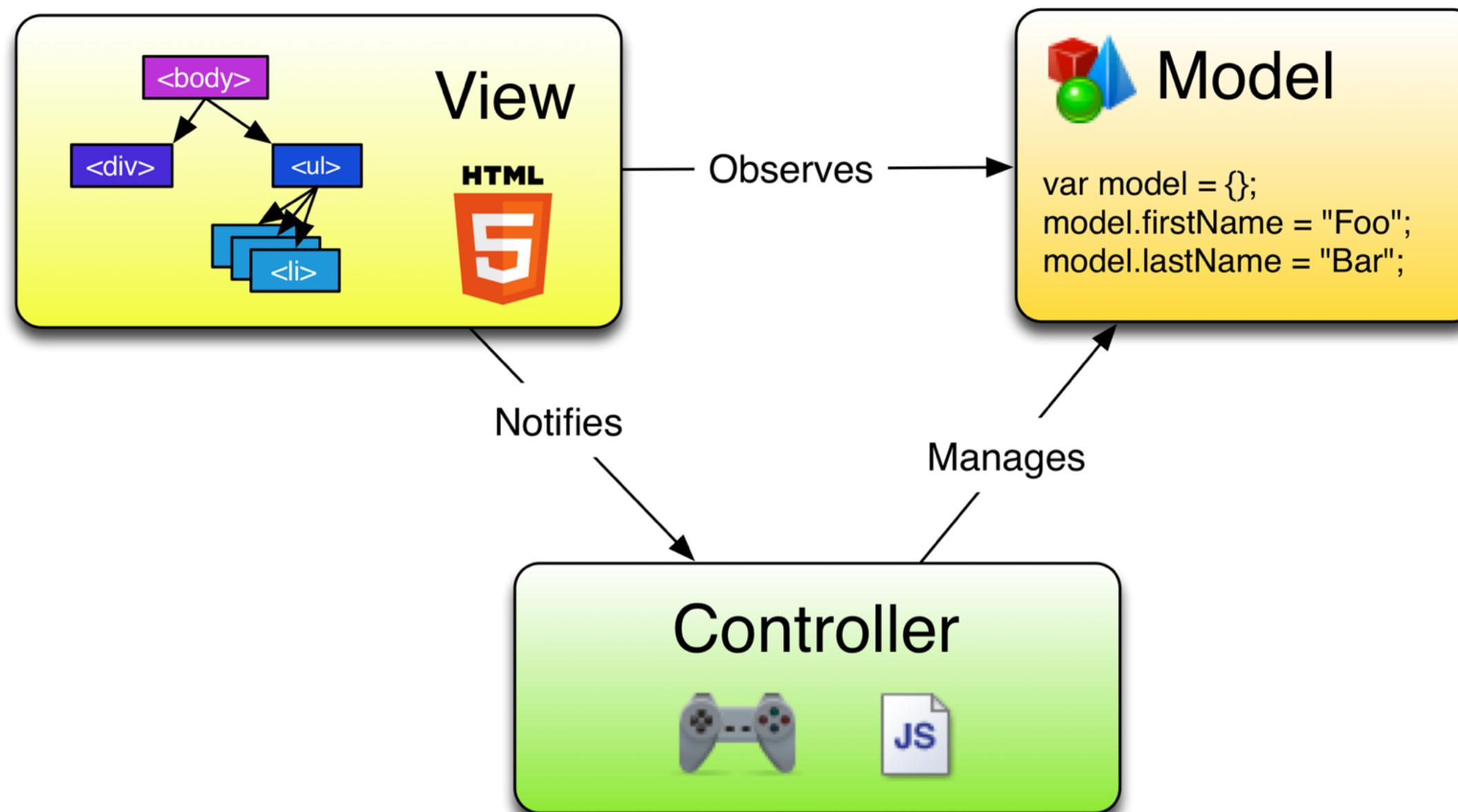
app.js

```
angular.module('myApp', ['ngRoute',
  'myApp.filters',
  'myApp.services',
  'myApp.directives',
  'myApp.controllers'
])
```

controllers.js

```
angular.module('myApp.controllers', []).
  controller('MyCtrl1', [function() {
}])
```

Model View Controller



Data Binding

friend.js

```
$scope.friend = {  
  name: "Fernand"  
};
```

friend.html

```
{{friend.name}}          // 1-way  
<input ng-model="friend.name"> // 2-way
```

Solving FOUC

This will work just fine — if it's not on the first page:

```
<p>{{friend.name}}</p>
```

Use **ng-cloak** or **ng-bind** attribute:

```
<p ng-cloak>{{friend.name}}</p>
```

```
<p ng-bind="friend.name"></p>
```

Directives

```
<div ng-repeat="entry in news.entries">
  <span ng-bind="entry.title"></span>
  <button ng-click="delete($index)">
    Delete
  </button>
</div>
```

Directives with valid HTML5

```
<div data-ng-repeat="entry in news.entries">
  <span data-ng-bind="entry.title"></span>
  <button data-ng-click="delete($index)">
    Delete
  </button>
</div>
```

```
<div data-ng:repeat="entry in news.entries">
  <span data-ng:bind="entry.title"></span>
  <button data-ng:click="delete($index)">
    Delete
  </button>
</div>
```

Custom Directives

```
$scope.customer = {  
  name: 'Franklin',  
  address: '1830 Blake'  
};  
  
<div ng-controller="MyController">  
  <my-customer></my-customer>  
</div>  
  
.directive('myCustomer', function() {  
  return {  
    template: 'Name: {{customer.name}} \  
    Address: {{customer.address}}'  
  };  
});
```

Built-In Directives

ng-href

ng-src

ng-disabled

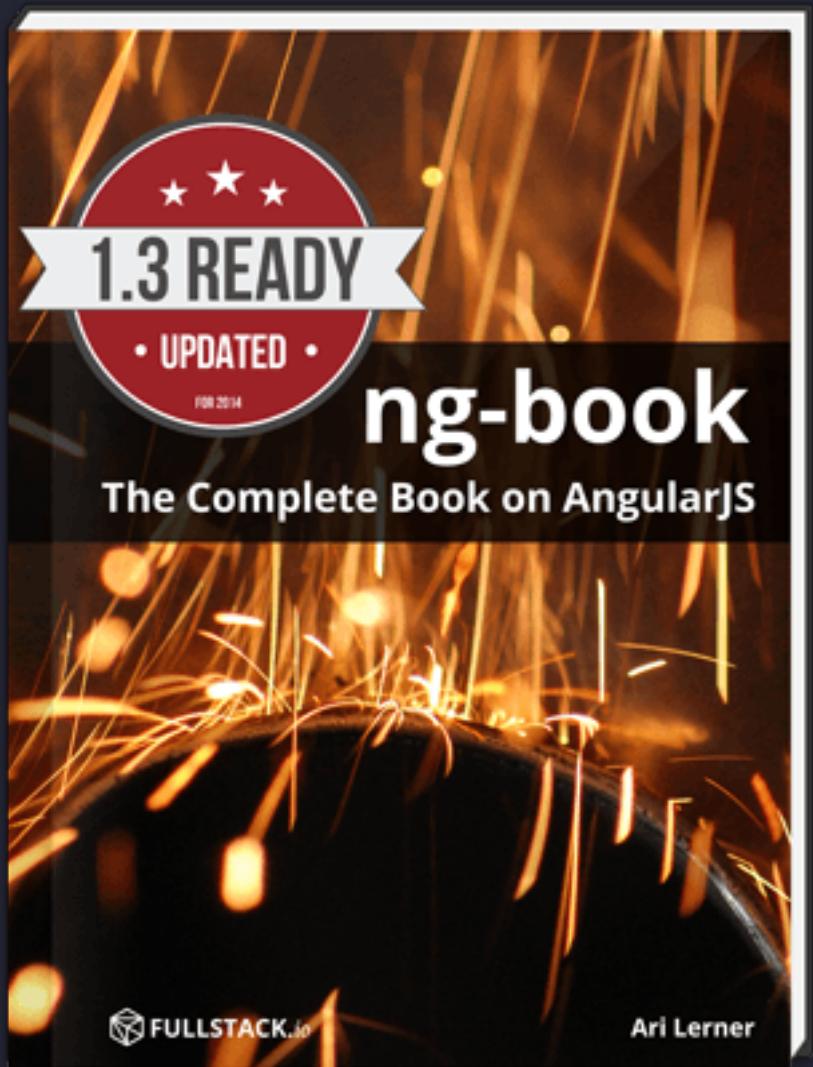
ng-checked

ng-readonly

ng-selected

ng-class

ng-style



Services

```
var services = angular.module('myApp.services', ['ngResource']);

services.factory('LoginService', function($resource) {
  return $resource(':action', {}, {
    authenticate: {
      method: 'POST',
      params: {'action': 'authenticate'},
      headers: {'Content-Type': 'application/x-www-form-urlencoded'}
    }
  );
});

services.factory('NewsService', function($resource) {
  return $resource('news/:id', {id: '@id'});
});
```

\$http

```
$http({method: 'GET', url: '/news'}).  
  success(function(data, status, headers, config) {  
    // this callback will be called asynchronously  
    // when the response is available  
  }).  
  error(function(data, status, headers, config) {  
    // called asynchronously if an error occurs  
    // or server returns response with an error status.  
});  
  
$http.get('/news').success(successCallback);  
$http.post('/news', data).success(successCallback);
```

\$q

```
myApp.factory('HelloWorld', function($q, $timeout) {  
  
    var getMessages = function() {  
        var deferred = $q.defer();  
  
        $timeout(function() {  
            deferred.resolve(['Hello', 'world!']);  
        }, 2000);  
  
        return deferred.promise;  
    };  
  
    return {  
        getMessages: getMessages  
    };  
});
```

\$q

```
myApp.controller('HelloCtrl', function($scope, HelloWorld) {  
  HelloWorld.getMessages().then(function(messages) {  
    $scope.messages = messages;  
  });  
});
```

Dependency Injection

```
.controller('LoginController', function($scope, $rootScope, $location,
                                         $http, $cookieStore, LoginService) {
  $scope.login = function () {
    LoginService.authenticate($.param({username: $scope.username,
                                       password: $scope.password}),
    function (user) {
      $rootScope.user = user;
      $http.defaults.headers.common[xAuthTokenHeaderName] = user.token;
      $cookieStore.put('user', user);
      $location.path("/");
    });
  };
})
```

Filters

```
{ { name | uppercase } }
```

```
<!-- Displays: 123.46 -->  
{ { 123.456789 | number:2 } }
```

```
<!-- In en-US locale, '$1000.00' will be shown -->  
{ { 1000 | currency } }
```

```
<!-- all of the words with e in them ["Lerner","Likes","Eat"] -->  
{ { ['Ari', 'Lerner', 'Likes', 'To', 'Eat', 'Pizza'] | filter:'e' } }
```

also: lowercase, limitTo, orderBy

Routes

```
.config(['$routeProvider', '$locationProvider', '$httpProvider',
  function ($routeProvider, $locationProvider, $httpProvider) {
    $routeProvider.when('/create', {
      templateUrl: 'partials/create.html', controller: 'CreateController'
    });
    $routeProvider.when('/edit/:id', {
      templateUrl: 'partials/edit.html', controller: 'EditController'
    });
    $routeProvider.when('/login', {
      templateUrl: 'partials/login.html', controller: 'LoginController'
    });
    $routeProvider.otherwise({
      templateUrl: 'partials/index.html', controller: 'IndexController'
    });

    $locationProvider.hashPrefix('!');
  }
])
```

Routing: Navigation

```
$rootScope.logout = function () {
  delete $rootScope.user;
  delete $http.defaults.headers.common[xAuthTokenHeaderName];
  $cookieStore.remove('user');
  $location.path("/login");
};
```

Routing: Navigation

```
$rootScope.logout = function () {  
    delete $rootScope.user;  
    delete $http.defaults.headers.common[xAuthTokenHeaderName];  
    $cookieStore.remove('user');  
    $location.path("/login");  
};
```

Code Organization Revisited

Lineman helps you build fat-client JavaScript apps



It produces happiness by building assets, mocking servers, and running specs on every file change

```
git clone https://github.com/linemanjs/lineman-angular-template.git my-app
cd my-app
sudo npm install -g lineman
npm install
lineman run
```

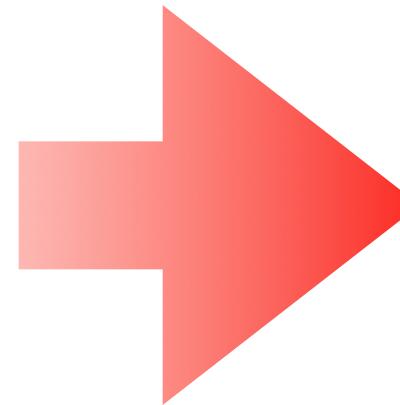
```
. └── app
    ├── js                                # <-- JS & CoffeeScript
    ├── img                               # <-- images (are merged into the 'img' folder inside of generated & dist)
    ├── static                            # <-- any other static files that need to be included in your built app
    └── pages
        └── index.us                      # <-- a template used to produce the application's index.html
    └── templates
        ├── other.us                     # <-- other templates will be compiled to a window.JST object
        ├── thing.hb                      # <-- underscore & handlebars are both already set up
        └── _partial.hb                   # <-- a handlebars partial, usable from within other handlebars templates
    └── config
        ├── application.js            # <-- Override application configuration
        ├── files.js                  # <-- Override named file patterns
        ├── server.js                 # <-- Define custom server-side endpoints to aid in development
        └── spec.json                 # <-- Override spec run configurations
    └── dist
        # <-- Generated, production-ready app assets
    └── generated
        # <-- Generated, pre-production app assets
    └── grunt.js                         # <-- gruntfile defines app's task config
    └── package.json                     # <-- Project's package.json
    └── tasks                            # <-- Custom grunt tasks can be defined here
    └── spec
        ├── helpers                  # <-- Spec helpers (loaded before other specs)
        └── some-spec.coffee          # <-- All the Jasmine specs you can write (JS or Coffee)
    └── vendor
        ├── js                        # <-- 3rd-party Javascript
        └── underscore.js             # <-- Underscore, because underscore is fantastic.
        └── img                      # <-- 3rd-party images (are merged into the 'img' folder inside of generated & dist)
        └── css                        # <-- 3rd-party CSS
```



Feature	Lineman	Yeoman
Generators	1 builtin scaffold, clonable templates, no sub-generators	No builtin scaffolds, installable generators, many sub-generators
Command Line Interface	Wrapped utilities that are accessed via CLI ie: <code>lineman grunt</code>	Unwrapped utilities that are accessed via their CLI ie: <code>grunt</code>
HTML5 pushState Simulator	builtin and enabled via config/application	-
API Stubbing Prototyping	builtin and enabled via config/server	-
API Proxy	builtin and enabled via config/application	-
Test Runner	testem, configured for dev and ci mode with <code>lineman spec</code> and <code>lineman spec-ci</code>	-
Grunt Task Lifecycle	preconfigured phases: common, dev, dist, split into config/files and config/application	generator specific
Dependency Graph	intentionally shallow and lightweight	generator specific, often deep and heavy
Cute ASCII Text Greeter	-	snazzy wizard based CLI
Package Management	none builtin but supports various, including bower, via extensions	bower
SourceMap Generation	builtin, defaults to <code>grunt-concat-sourcemap</code>	-
Dev Server	<code>lineman run</code> , spins up express.js	generator specific
Directory Structure	common across project templates	generator specific
License	MIT	BSD



```
app/
  css/
    app.css
    bootstrap.css
  img/
    phones/
      glyphicons-halflings.png
  js/
    app.js
    controllers.js
    filters.js
    services.js
  partials/
    phone-detail.html
    phone-list.html
  phones/
    nexus-s.json
    phones.json
    index.html
  test/
    e2e/
      runner.html
      scenarios.js
    unit/
      controllersSpec.js
      directivesSpec.js
      filtersSpec.js
      servicesSpec.js
```



```
app/
  app.css
  app.js
  bootstrap.css
  components/
    checkmark-filter/
      checkmark-filter.js
      checkmark-filter_test.js
  phonecat/
    phonecat.js
    phonecat-service.js
    phonecat-service_test.js
  detail/
    phone-detail.html
    phone-detail-controller.js
    phone-detail-controller_test.js
  index.html
  list/
    phone-list.html
    phone-list-controller.js
    phone-list-controller_test.js
```

Google's Recommendations for Angular App Structure

Testing Strategies for Angular JS

David Mosher · 11 videos

Subscribe 3,782

16,318

142 2

The image shows a YouTube video player window with the title "Testing Strategies for Angular JS" by David Mosher. The video player has a dark theme. In the center, there is a Sublime Text 2 window showing code for testing an AngularJS controller. The code includes a describe block for the LoginController and a function for successfully logging in. Below the code editor, a terminal window shows the command "git branch: master, index: 2?". At the bottom of the video player, there is a control bar with a play button, volume controls, a timestamp (15:38 / 1:01:45), and other video settings like HD.

```
describe("controller: LoginController", function() {
  describe("successfully loggin in", function() {
    });
  });
angular.module("app").controller('LoginController', function($scope, $location, AuthenticationService) {
  $scope.credentials = { username: "", password: "" };
  var onLoginSuccess = function() {
    $location.path('/home');
  };
  $scope.login = function() {
    AuthenticationService.login($scope.credentials).success(onLoginSuccess);
  };
});
```

Testing

Karma - test runner, framework agnostic

Jasmine - unit tests, framework agnostic

Protractor - integration tests, angular specific

Lineman - productivity, framework agnostic



Testing: Controllers

```
describe("controller: LoginController", function() {  
  beforeEach(function() {  
    module("app");  
  });  
  
  beforeEach(inject(function($controller, $rootScope, $location,  
                        AuthenticationService, $httpBackend) {  
    this.$location = $location;  
    this.$httpBackend = $httpBackend;  
    this.scope = $rootScope.$new();  
    this.redirect = spyOn($location, 'path');  
    $controller('LoginController', {  
      $scope: this.scope,  
      $location: $location,  
      AuthenticationService: AuthenticationService  
    });  
  }));  
});
```



Jasmine

Testing: Controllers



Jasmine

```
afterEach(function() {
  this.$httpBackend.verifyNoOutstandingRequest();
  this.$httpBackend.verifyNoOutstandingExpectation();
});

describe("successfully logging in", function() {
  it("should redirect you to /home", function() {
    this.$httpBackend.expectPOST('/login',
      this.scope.credentials).respond(200);
    this.scope.login();
    this.$httpBackend.flush();
    expect(this.redirect).toHaveBeenCalledWith('/home');
  });
});
});
```

Testing: Directives

```
beforeEach(inject(function($rootScope, $compile) {
  this.directiveMessage = 'ralph was here';
  this.html = "<div shows-message-when-hovered message='"
    + this.directiveMessage + "'></div>";
  this.scope = $rootScope.$new();
  this.scope.message = this.originalMessage = 'things are looking grim';
  this.elem = $compile(this.html)(this.scope);
}));

describe("when a user mouses over the element", function() {
  it("sets the message on the scope to the message attribute", function() {
    this.elem.triggerHandler('mouseenter');
    expect(this.scope.message).toBe(this.directiveMessage);
  });
});
```

Testing: Directives with CoffeeScript

```
describe "directive: shows-message-when-hovered (coffeescript)", ->
```

```
  Given -> module("app")
```

```
  Given inject ($rootScope, $compile) ->
    @directiveMessage = 'ralph was here'
    @html = "<div shows-message-when-hovered
              message='#{@directiveMessage}'></div>"
    @scope = $rootScope.$new()
    @scope.message = @originalMessage = 'things are looking grim'
    @elem = $compile(@html)(@scope)
```

```
  describe "when a user mouses over the element", ->
```

```
    When -> @elem.triggerHandler('mouseenter')
```

```
    Then "the message on the scope is set to the message attribute", ->
```

```
      @scope.message == @directiveMessage
```

Testing: End-to-End

```
protractor = require("protractor")
require "protractor/jasminewd"
require 'jasmine-given'

describe "my angular app", ->
  ptor = protractor.getInstance()
  describe "visiting the login page", ->
    Given -> ptor.get "/"

    describe "when a user logs in", ->
      Given -> ptor.findElement(protractor.By.input("credentials.username")).sendKeys "Ralph"
      Given -> ptor.findElement(protractor.By.input("credentials.password")).sendKeys "Wiggum"
      When -> ptor.findElement(protractor.By.id("log-in")).click()
      Then -> ptor.findElement(protractor.By.binding("{{ message }}")).getText().then (text) ->
        expect(text).toEqual "Mouse Over these images to see a directive at work"
```

Building with Grunt

```
sudo npm install
```

```
sudo npm install -g grunt-cli
```

```
vi package.json
```

```
"grunt": "~0.4.1",
"grunt-contrib-concat": "~0.3.0",
"grunt-contrib-uglify": "~0.2.7",
"grunt-contrib-cssmin": "~0.7.0",
"grunt-usemin": "~2.0.2",
"grunt-contrib-copy": "~0.5.0",
"grunt-rev": "~0.1.0",
"grunt-contrib-clean": "~0.5.0",
"matchdep": "~0.3.0"
```

Gruntfile.js

```
module.exports = function (grunt) {

  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),

    clean: ["dist", '.tmp'],

    copy: {
      main: {
        expand: true,
        cwd: 'app/',
        src: ['**', '!js/**', '!lib/**', '!**/*.css'],
        dest: 'dist/'
      }
    },
    rev: {
      files: {
        src: ['dist/**/*.{js,css}']
      }
    },
  },
}
```

Gruntfile.js

```
useminPrepare: {
  html: 'app/index.html'
},
usemin: {
  html: ['dist/index.html']
},
uglify: {
  options: {
    report: 'min',
    mangle: false
  }
}
});
```

Gruntfile.js

```
require('matchdep').filterDev('grunt-*').forEach(grunt.loadNpmTasks);

// Tell Grunt what to do when we type "grunt" into the terminal
grunt.registerTask('default', [
  'copy', 'useminPrepare', 'concat', 'uglify', 'cssmin', 'rev', 'usemin'
]);
};
```

index.html comments

```
<head>
  <title>My AngularJS App</title>
  <!-- build:css css/seed.min.css -->
  <link rel="stylesheet" href="css/app.css"/>
  <link rel="stylesheet" href="css/app2.css"/>
  <!-- endbuild -->
</head>
<body>

  <!-- build:js js/seed.min.js -->
  <script src="lib/angular/angular.js"></script>
  <script src="lib/angular/angular-route.js"></script>
  <script src="js/app.js"></script>
  <script src="js/services.js"></script>
  <script src="js/controllers.js"></script>
  <script src="js/filters.js"></script>
  <script src="js/directives.js"></script>
  <!-- endbuild -->
</body>
```

dist/index.html

```
<head>
  <title>My AngularJS App</title>
  <link rel="stylesheet" href="css/f050d0dc.seed.min.css"/>
</head>
<body>

  <script src="js/8973cf0f.seed.min.js"></script>
</body>
```

After Grunt

	YSlow	Page Speed
No optimization	75 27 HTTP requests / 464K	55/100
Apache optimization (gzip and expires headers)	89 initial load: 26 requests / 166K primed cache: 4 requests / 40K	88/100
Apache + concat/minified/versioned files	98 initial load: 5 requests / 136K primed cache: 3 requests / 1.4K	93/100

http://raibledesigns.com/rd/entry/using_grunt_with_angularjs_for

You shouldn't have to worry about FEO

Matt Raible @mraible 15 Jan
Using Grunt with AngularJS for Front End Optimization
raibledesigns.com/rd/entry/using.... From 27 requests to 5 on initial page load. #angularjs #grunt

Mark @markj9 Follow
@mraible that might be a bad thing! you should go listen to Igor..
rubyrogues.com/135-rr-http-2-...

12:40 PM - 15 Jan 2014

← ↗ ★

[http://raibledesigns.com/rd/entry/you_shouldn't_have_to](http://raibledesigns.com/rd/entry/you_shouldn_t_have_to)

HTTP/2 Performance Anti-Patterns?

Split dominant content domains

Reduce requests

Merging

Sprites

DataURIs



<http://www.slideshare.net/andydavies>

UI Bootstrap

<http://angular-ui.github.io/bootstrap>

UI Bootstrap

Bootstrap components written in pure **AngularJS** by the
AngularUI Team

```
<script src="lib/angular/ui-bootstrap-0.12.0.min.js"></script>
<script src="lib/angular/ui-bootstrap-tpls-0.12.0.min.js"></script>

angular.module('myApp', ['ui.bootstrap']);
```

UI Bootstrap: Carousel



UI Bootstrap: Carousel

```
<div ng-controller="CarouselDemoCtrl">
  <div style="height: 305px">
    <carousel interval="myInterval">
      <slide ng-repeat="slide in slides" active="slide.active">
        
        <div class="carousel-caption">
          <h4>Slide {{$index}}</h4>
          <p>{{slide.text}}</p>
        </div>
      </slide>
    </carousel>
  </div>
</div>
```

Foundation for Apps

<http://foundation.zurb.com/apps>



Templatizing Angular

Templatizing Angular Create simple Angular prototypes and apps without ever writing a line of Angular code. We create your Angular routes so you don't have to — just name your views and URLs.



Component Driven

There's no need to remember to call a div, ul or dl elements. Call our JS and CSS components with simple, custom HTML tags by prefixing your components' names with "zf-".



Add Motion to Views

Let users see when content changes by adding "enter" and "exit" motion classes to views. Simply define one of our predefined Motion UI classes at the top of your view and you're done.



Foundation

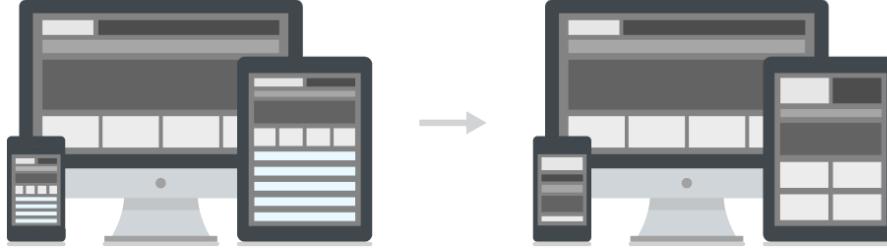
The screenshot shows a web browser window displaying the Foundation documentation at foundation.zurb.com/docs/. The page has a dark header with the word "Foundation". Below the header is a sidebar on the left containing sections like "Foundation Documentation", a search bar, and a large blue "Download Foundation" button. The main content area features a large heading "Getting Started" followed by a paragraph about the ease of getting started with Foundation. To the right of this is another section titled "What is Foundation?" with a description of the framework's advanced responsive design and its history. At the bottom of the page, there is a diagram showing a desktop monitor and a smartphone transitioning to a larger desktop monitor and a larger smartphone, symbolizing responsive design. A copyright notice at the bottom right reads "© 2015 Raible Designs".

Getting Started

Getting started with Foundation is easy. Whether you are a master of Sass, building a new app, or making your existing site responsive, we've got you covered.

What is Foundation?

Foundation is the most advanced, responsive front-end framework in the world. The framework is mobile friendly and ready for you to customize it any way you want to use it.



Millions of designers and engineers use Foundation as part of their workflows. It was the first framework to introduce the concepts of responsive design, semantics, mobile first and partials. It's also compatible with most browsers and

Foundation

The image shows two side-by-side screenshots of the Foundation documentation website, likely taken from a Mac OS X desktop environment. Both screenshots show the same content but with different scroll positions.

Screenshot 1 (Left): Shows the "Getting Started" page. The main content area contains the heading "Getting Started" and a paragraph of text: "Getting started with Foundation is easy. Whether you are a master of Sass, building a new app, or making your existing site responsive, we've got you covered." Below this is a section titled "What is Foundation?" with a paragraph: "Foundation is the most advanced, responsive front-end framework in the world. The framework is mobile friendly and ready for you to customize it any way you want to use it." At the bottom of the page are two small Foundation icon images.

Screenshot 2 (Right): Shows the "Getting Started" page with a longer scroll. The main content area now includes a sidebar on the left containing a navigation menu with items like "FOUNDATION", "Home", "SETUP", "Getting Started", "CSS", "Sass", "Sass Files", "Using Sass", "Applications", "Javascript", "Global Styles", "Kitchen Sink", "Upgrading", "STRUCTURE", and "Media Queries". The "Media Queries" item is highlighted with a red box. The main content area also contains the "What is Foundation?" section and its associated text.

Sass Mixins

```
<div class="row">
  <main class="medium-9 columns">
    <p>Main content</p>
  </main>
  <aside class="medium-3 columns">
    <p>Sidebar</p>
  </aside>
</div>
```

Sass Mixins

```
<div class="layout">
  <main class="layout-content">
    <p>Main content</p>
  </main>
  <aside class="layout-sidebar">
    <p>Sidebar</p>
  </aside>
</div>
```

Sass Mixins

```
@import "foundation/components/grid";

.layout {
    // `layout` container functions as a row
    @include grid-row();
}

.layout-content {
    // Mobile-first: make `layout-container` full-width
    @include grid-column(12);

    // On medium-up size, make `layout-container` 9 columns wide
    @media #{$medium-up} {
        @include grid-column(9);
    }
}

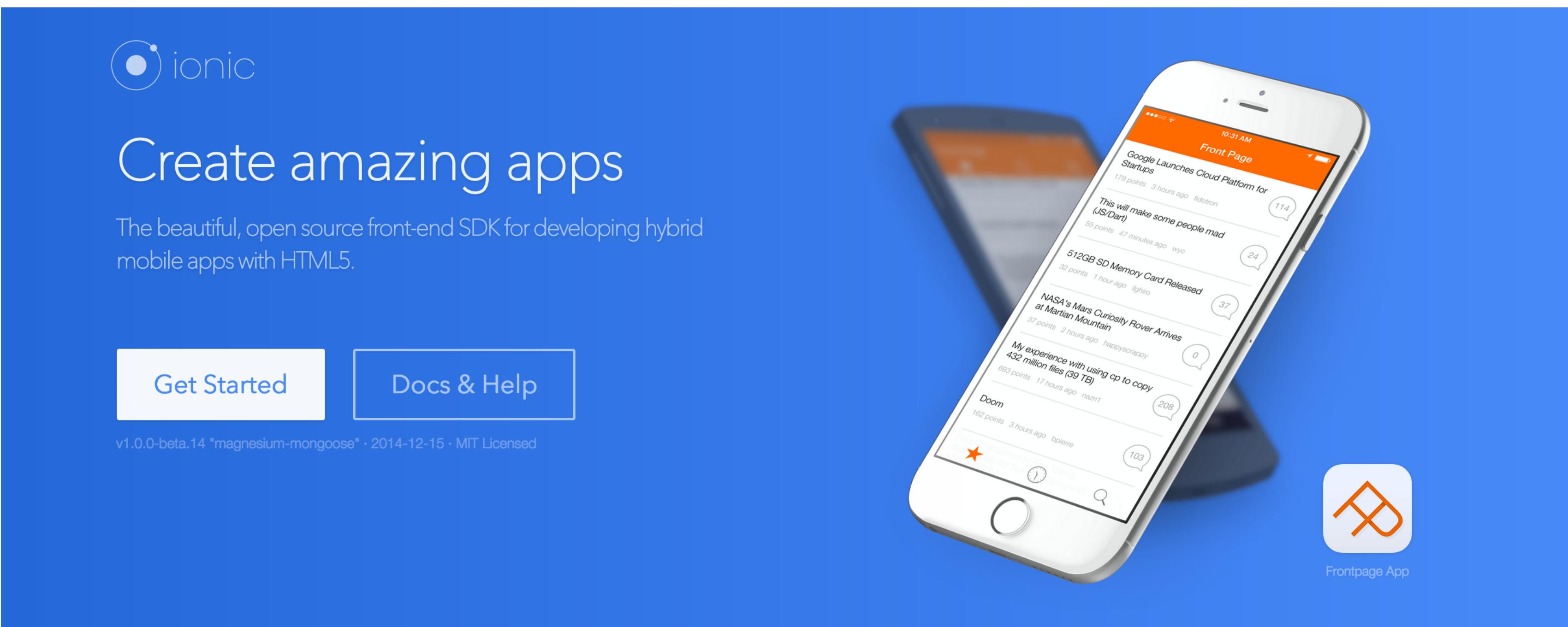
.layout-sidebar {
    // Mobile-first: make `layout-sidebar` full-width
    @include grid-column(12);

    // On medium-up size, make `layout-sidebar` 3 columns wide
    @media #{$medium-up} {
        @include grid-column(3);
    }
}
```

<http://bit.ly/1Kh1ha5>

Ionic Framework

<http://ionicframework.com>



The screenshot shows the Ionic Framework landing page. At the top left is the Ionic logo (a white circle with a dot). Below it is the tagline "Create amazing apps". Underneath that is a description: "The beautiful, open source front-end SDK for developing hybrid mobile apps with HTML5." Two prominent buttons are at the bottom left: "Get Started" (white background) and "Docs & Help" (blue background). At the very bottom, the text "v1.0.0-beta.14 "magnesium-mongoose" · 2014-12-15 · MIT Licensed" is visible.

Get Started Docs & Help

v1.0.0-beta.14 "magnesium-mongoose" · 2014-12-15 · MIT Licensed

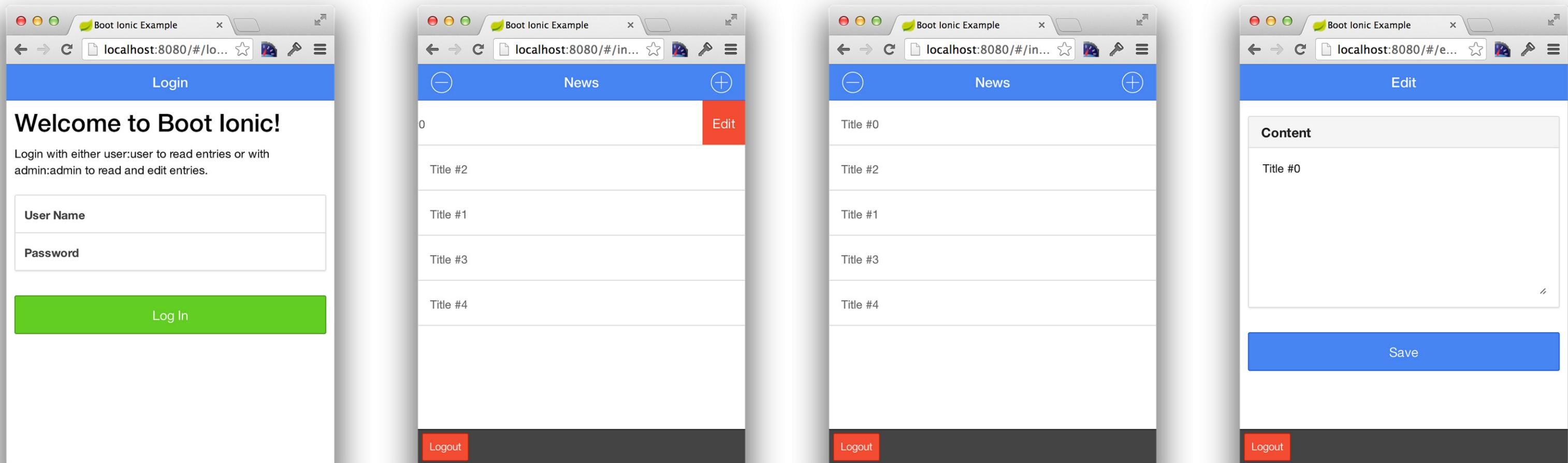


A silver smartphone is shown from a side-on perspective, tilted slightly. It displays a hybrid mobile application with a white header bar showing the time as 10:31 AM and the title "Front Page". The main content area lists several news items with small profile pictures, names, and interaction counts (e.g., 114, 24, 37, 0, 208, 103). A magnifying glass icon is at the bottom right of the screen.



Frontpage App

My Ionic Experience



http://raibledesigns.com/rd/entry/developing_an_ios_native_app

JHipster

<http://jhipster.github.io/>

Greetings, Java Hipster!



JHipster is a
Yeoman generator,



used to create a
Spring Boot + AngularJS
project.



JHipster

<http://jhipster.github.io/>

Foundational Frameworks

Spring Boot

Spring Security

AngularJS

Bootstrap

Bower

Metrics

Project Options

Java 7 or Java 8

Maven or Gradle

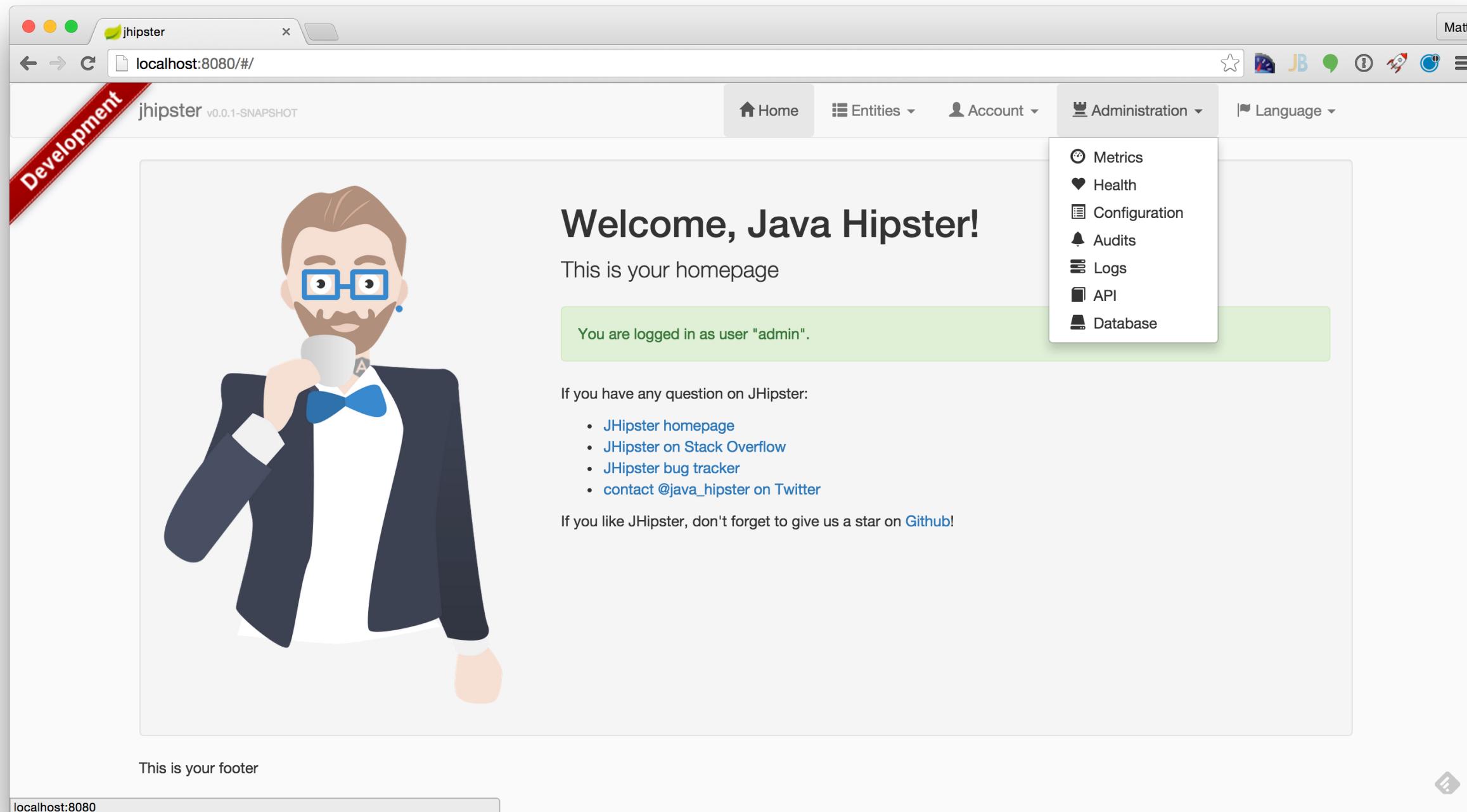
Authentication Type: cookie-based or OAuth2

Type of Database: SQL or NoSQL

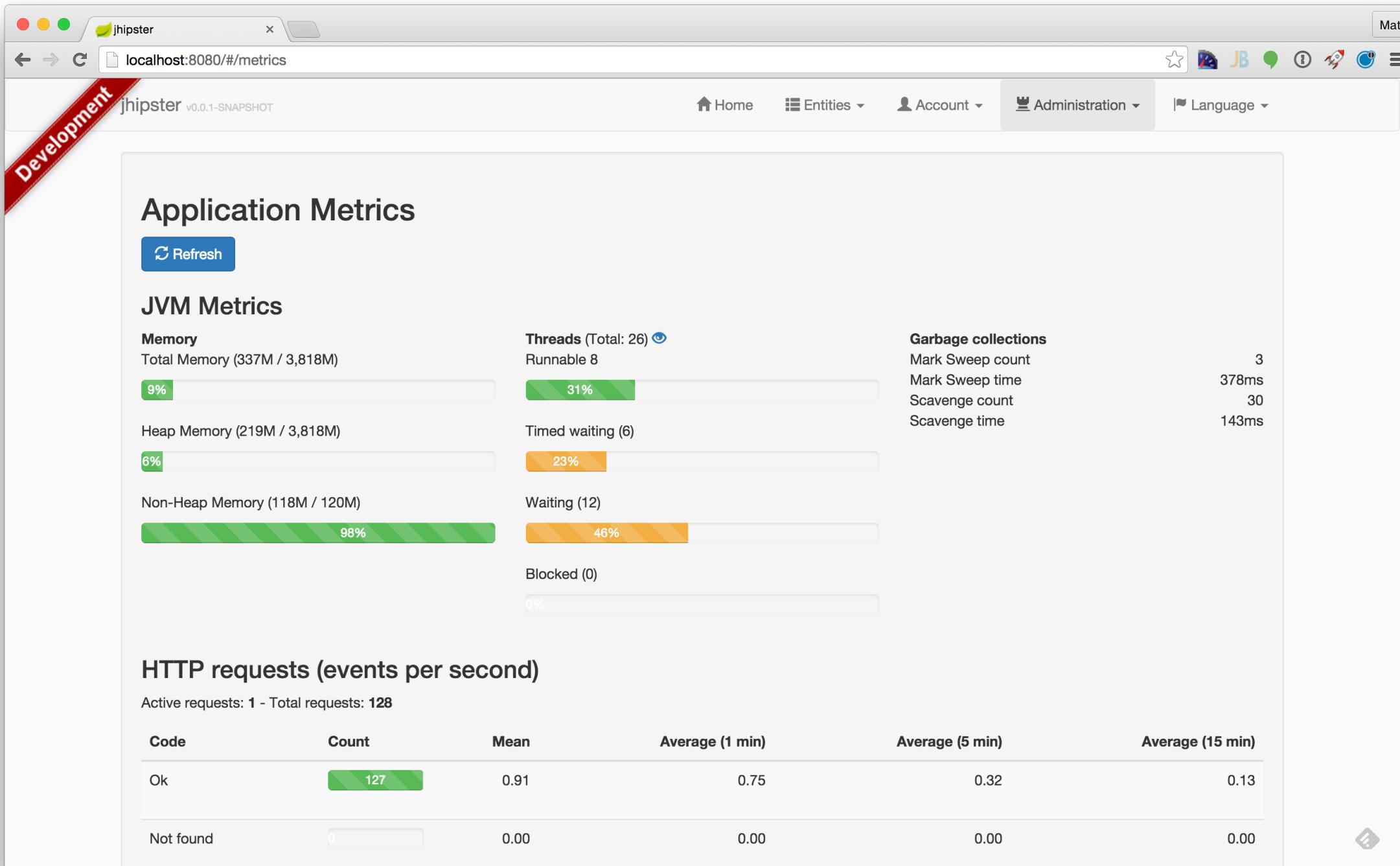
Caching: EhCache or Hazelcast

Grunt or Gulp.js

JHipster



JHipster: Metrics



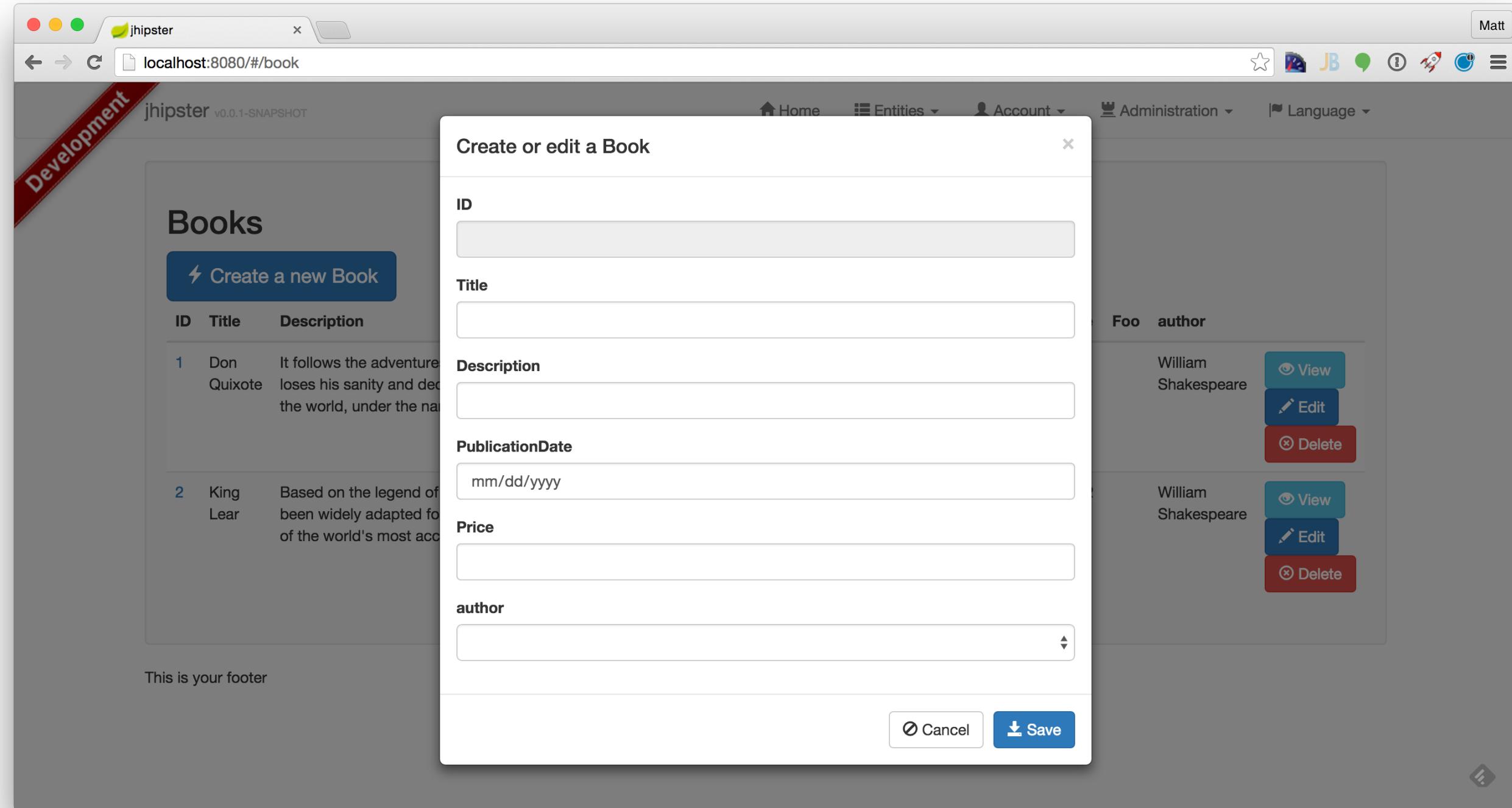
JHipster: Code Generation

A screenshot of a web browser window titled "jhipster" showing a JHipster application at localhost:8080/#/book. The page has a red "Development" ribbon on the left. The header includes links for Home, Entities, Account, Administration, and Language, and a user "Matt". The main content area is titled "Books" and features a blue button "⚡ Create a new Book". Below is a table listing two books:

ID	Title	Description	PublicationDate	Price	Foo	author	
1	Don Quixote	It follows the adventures of a nameless hidalgo who reads so many chivalric novels that he loses his sanity and decides to set out to revive chivalry, undo wrongs, and bring justice to the world, under the name Don Quixote.	1615-01-01	20.5		William Shakespeare	View Edit Delete
2	King Lear	Based on the legend of Leir of Britain, a mythological pre-Roman Celtic king, the play has been widely adapted for the stage and motion pictures, with the title role coveted by many of the world's most accomplished actors.	1606-01-01	15.32		William Shakespeare	View Edit Delete

The footer contains the text "This is your footer".

JHipster: Code Generation



AngularJS Batarang

A screenshot of a Mac OS X desktop showing a browser window and the Chrome DevTools interface. The browser window title is "Angular Rest SpringSecurity" and the URL is "localhost:8080/#!/". The page content shows a "News" section with a single item titled "Title #0" and buttons for "Create", "Remove", and "Edit". The top right of the page shows a dropdown menu for "admin". Below the browser window is the Chrome DevTools "AngularJS" panel. The "Performance" tab is selected. The "Scopes" section shows a tree structure starting from "Scope (002)" down to "Scope (00D)". The "Models for (002)" section displays the following JSON object:

```
{  
  hasRole: null  
  logout: null  
  user: {  
    name: admin  
    roles: {  
      ROLE_USER: true  
      ROLE_ADMIN: true  
    }  
    token: admin:1393391653180:0e9f29207c2faa6c1e4ebeee67bf3c49  
  }  
}
```

My Experience in 2013



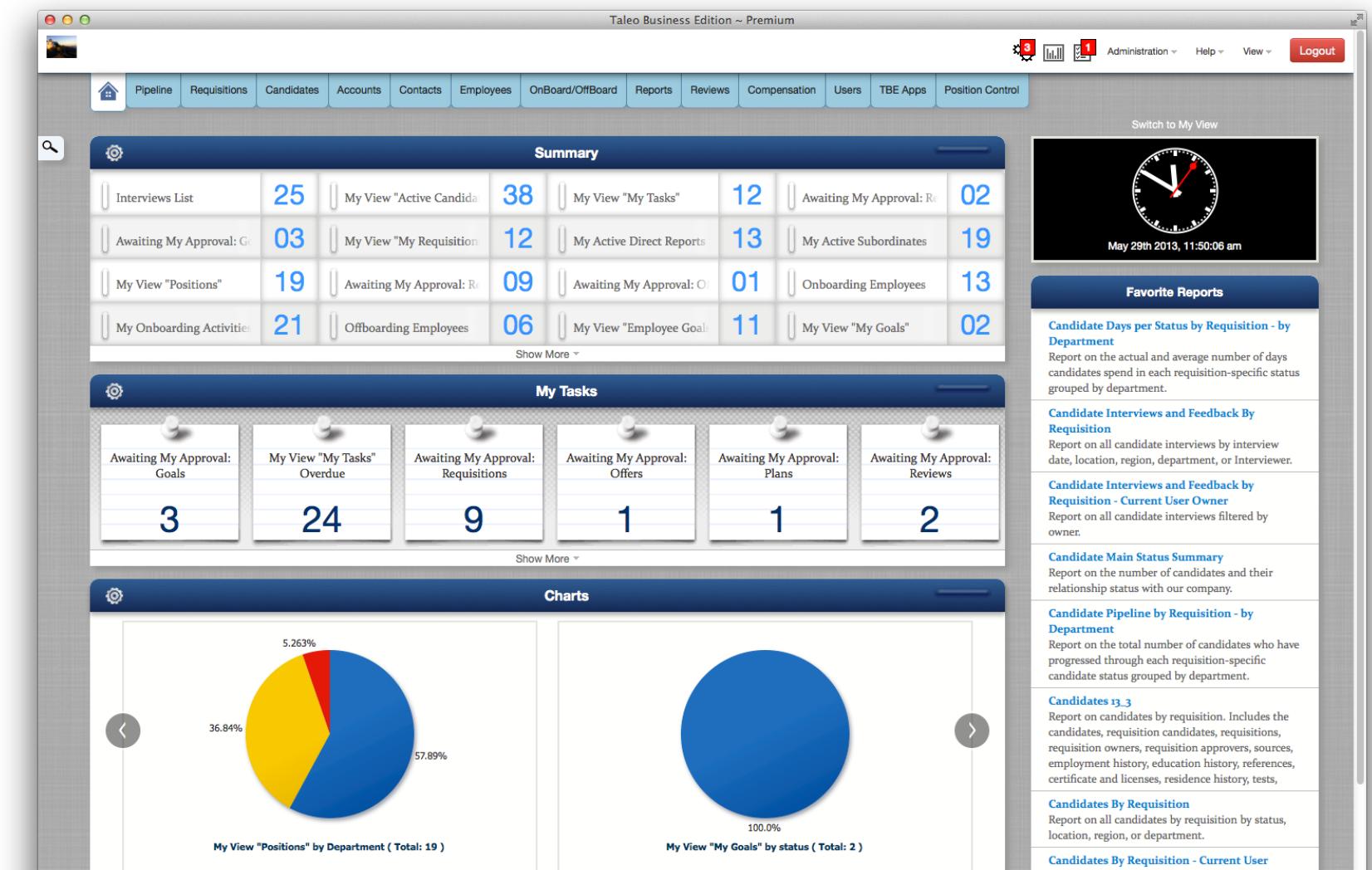
Developing with AngularJS Series

[Part I: The Basics](#)

[Part II: Dialogs and Data](#)

[Part III: Services](#)

[Part IV: Making it Pop](#)



My Experience in 2013

Travelpo^rt | Dashboard

Russell Thompson 

HOME MY PRODUCTS & SERVICES TRAINING & SUPPORT MY ACCOUNT REPORTS BILLING

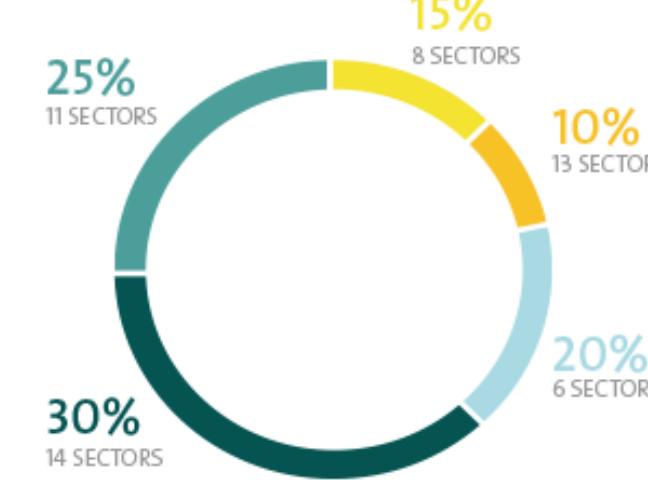
My Quicklinks

- Ask Travelpo^rt
- eNett
- Netsend
- Training Services Australia
- Travelpo^rt Academy
- Travelpo^rt Billing
- Travelpo^rt Billing Questions

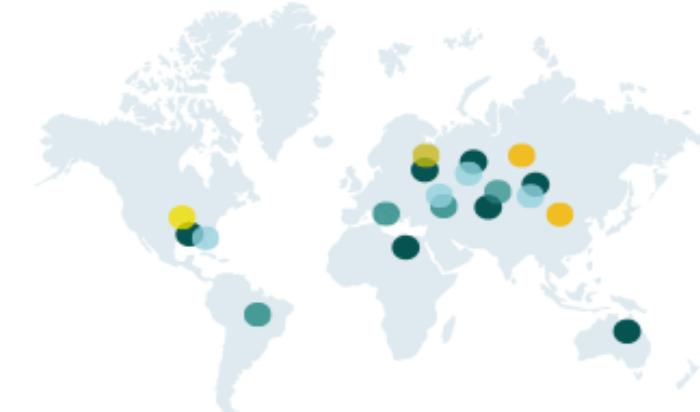
MONTHLY SUMMARY AGENCY GOALS AGENT SUMMARY AIRLINE SUMMARY

Airline Summary by City Pair

View: 2 Feb, 2013 to 28 Feb, 2013



Sector	Percentage	Sectors
1	30%	14 Sectors
2	25%	11 Sectors
3	20%	6 Sectors
4	15%	8 Sectors



- CONTINENTAL AIRLINES
- AIR CANADA
- QANTAS AIRWAYS
- DELTA AIR LINES INC
- UK AIR

[Download PDF](#)

Terminal Access

LAUNCH APPLICATION

<http://vimeo.com/mraible/angularjs-deep-dive>

2015 AngularJS Tutorials



Getting Started with AngularJS

[http://raibledesigns.com/rd/entry/getting started with angularjs](http://raibledesigns.com/rd/entry/getting_started_with_angularjs)

Testing AngularJS Applications

[http://raibledesigns.com/rd/entry/testing angularjs applications](http://raibledesigns.com/rd/entry/testing_angularjs_applications)



Spring and AngularJS

<http://spring.io/blog>



spring

ENGINEERING

Spring and AngularJS: A Secure Single Page Application

 ENGINEERING  DAVE SYER  JANUARY 12, 2015  6 COMMENTS

In this article we show some nice features of Spring Security, Spring Boot and Angular JS working together to provide a pleasant and secure user experience. It should be accessible to beginners with Spring and Angular JS, but there also is plenty of detail that will be of use to experts in either. This is actually the first in a series of articles on Spring Security and Angular JS, with new features exposed in each one successively. We'll improve on the application in the **second** and subsequent installments, but the main changes after this are architectural rather than functional.

<http://spring.io/blog/2015/01/12/spring-and-angular-js-a-secure-single-page-application>

Angular 2.0

```
<input type="text" [value]="firstName">  
  
<button (click)="addPerson()">Add</button>  
  
<input type="checkbox" [checked]="someProperty">
```

Concepts Eliminated in 2.0

Controllers

Directive Definition Object

\$scope

angular.module

jqLite



The Bad News

No migration path from Angular 1.x to 2.0

Angular 1.3 will be supported for 1.5 - 2 years

Will only support Evergreen Browsers (e.g. IE10+)

Learn more on **InfoQ**

<http://www.infoq.com/news/2014/10/angular-2-atscript>

THE TWELVE-FACTOR APP

INTRODUCTION

<http://12factor.net/>

In the modern era, software is commonly delivered as a service: called *web apps*, or *software-as-a-service*. The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use **declarative** formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a **clean contract** with the underlying operating system, offering **maximum portability** between execution environments;
- Are suitable for **deployment** on modern **cloud platforms**, obviating the need for servers and systems administration;
- **Minimize divergence** between development and production, enabling **continuous deployment** for maximum agility;
- And can **scale up** without significant changes to tooling, architecture, or development practices.

The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc).

How to Become an Artist

Part 1 of 3: Learn the Basics on Your Own

Take some time and try various mediums of art

Recognize your strengths

Do your research and learn the basics

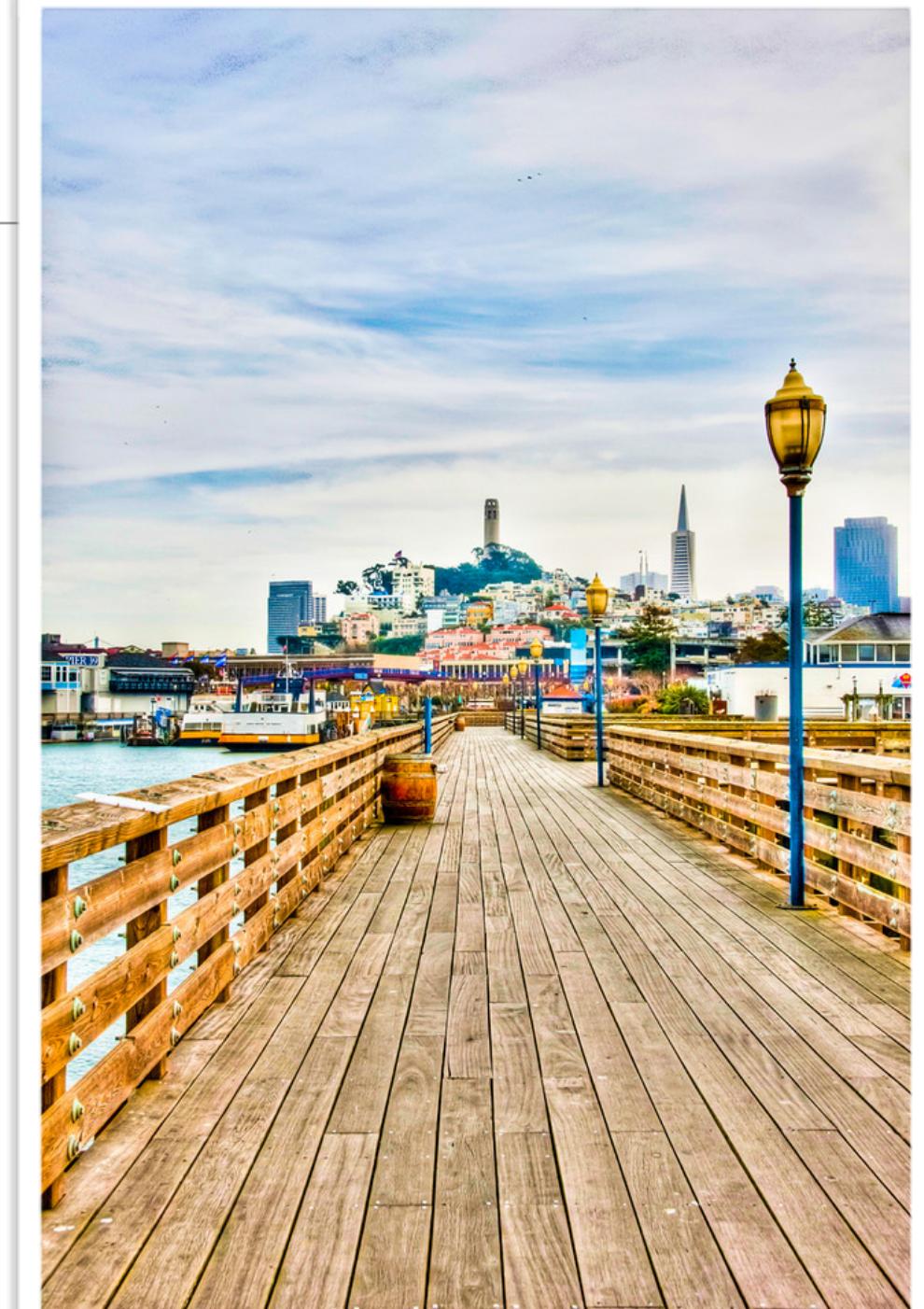
Get the supplies you will need

Observe the world around you

Make time for your art every day

Seek out the opinions of others

Develop your own style



<http://www.wikihow.com/Become-an-Artist>

Shortcut to becoming an Angular Artist

JUST DO IT.

Questions?

Contact Me!

 <http://raibledesigns.com>

 [@mraible](https://twitter.com/mraible)

Presentations

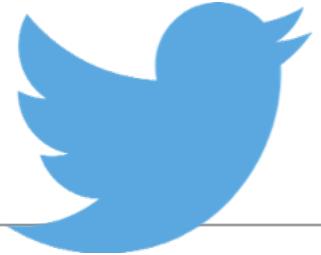
 <http://slideshare.net/mraible>

Code

 <http://github.com/mraible>



Who to follow on Twitter



AngularJS Team at Google

Miško Hevery - [@mhevery](#)

Igor Minar - [@igorMinar](#)

Brian Ford - [@briantford](#)

Web Performance

Ilya Grigorik - [@igrigorik](#)

Andy Davis - [@andydavies](#)

Steve Souders - [@Souders](#)

Resources

Angular Dart



<https://angulardart.org>

Devoxx AngularJS Talks on Parleys.com

DEVOXX™

<http://parleys.com/play/5148922b0364bc17fc56c91b> (2012)

<http://parleys.com/play/529321a5e4b054cd7d2ef4e1> (2013)

Egghead.io - <https://egghead.io/>

Code

GitHub

Angular Seed

<https://github.com/angular/angular-seed>



Lineman Application Template using AngularJS

<https://github.com/linemanjs/lineman-angular-template>

AngularJS + Rest + Spring Security

<https://github.com/joshlong/boot-examples/tree/master/x-auth-security>