

The Art of NGULARJS

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

Photos by  McGinity Photo

 **RAIBLE DESIGNS**
ENTERPRISE OPEN SOURCE CONSULTING

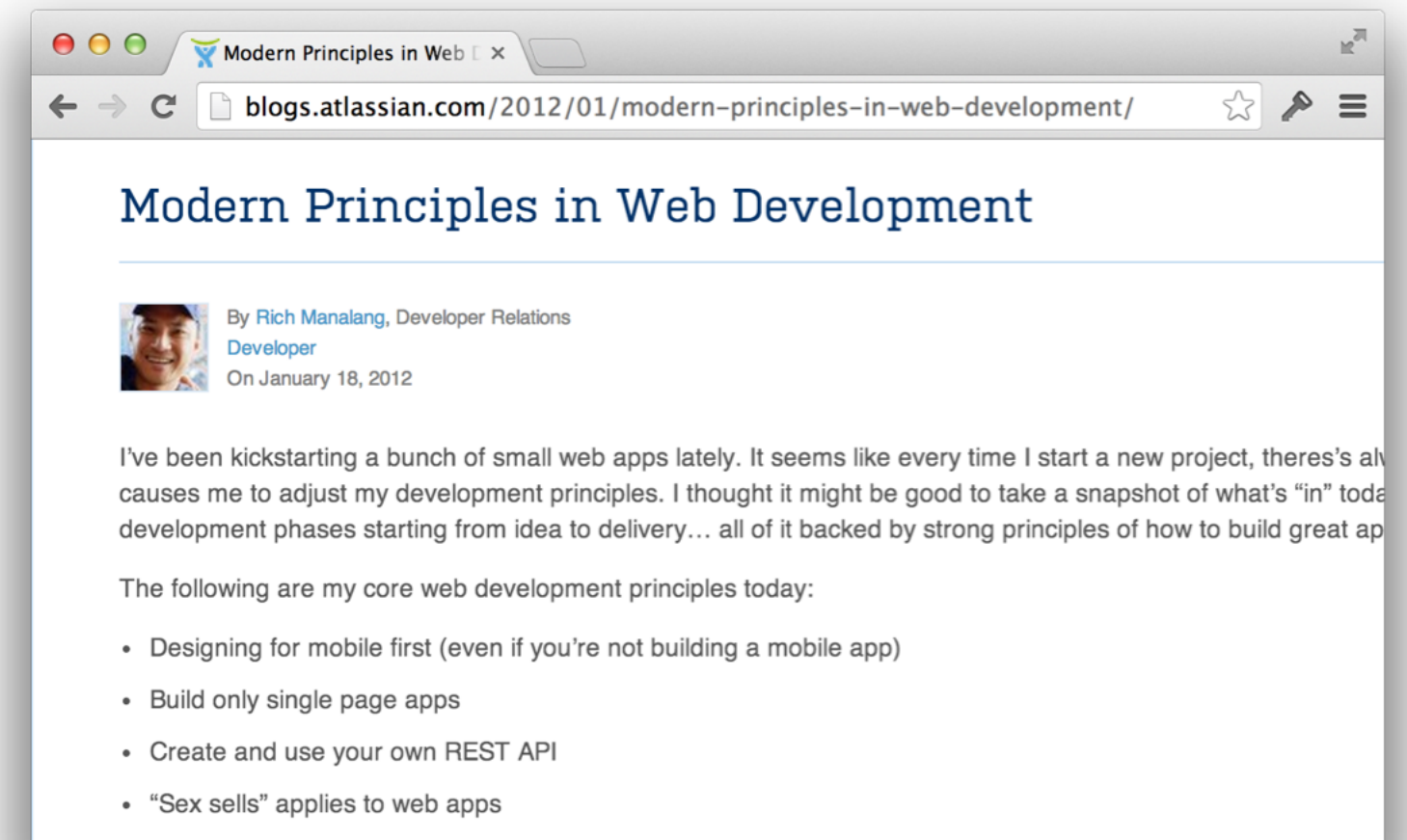
Modern Principles in Web Development

Design for mobile first (even if you're not building a mobile app)

Build only single page apps

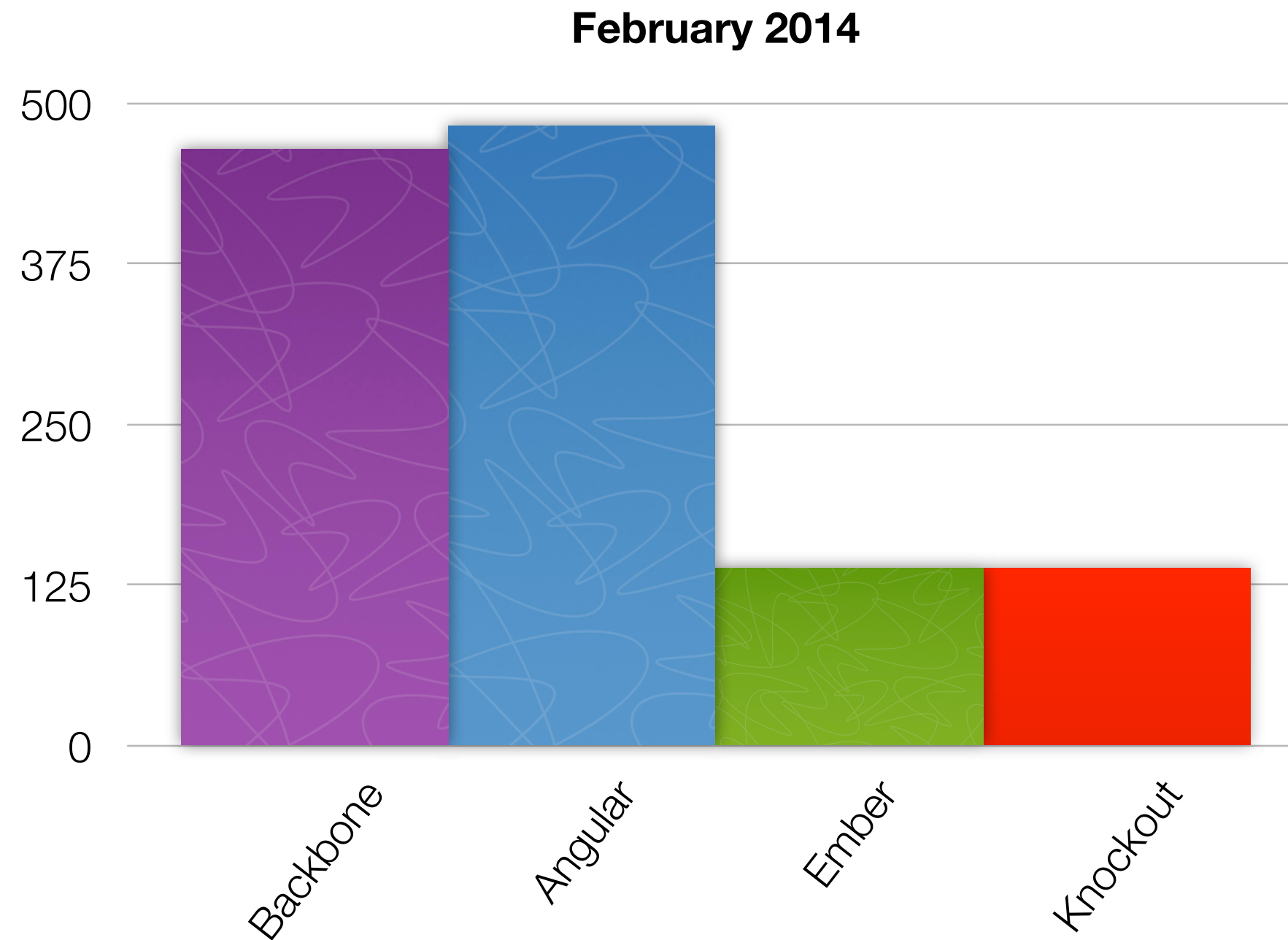
Create and use your own REST API

“Sex sells” applies to web apps

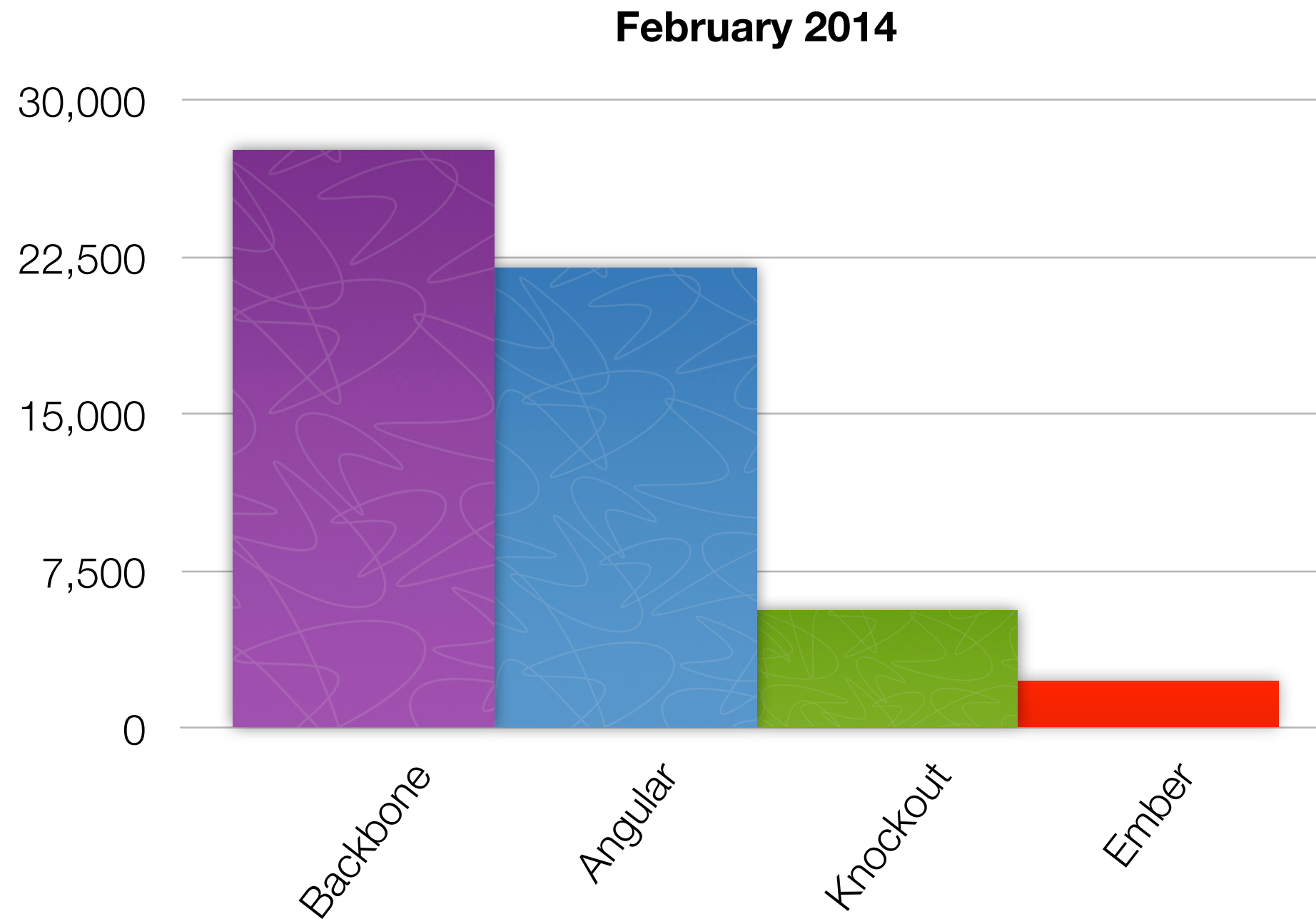




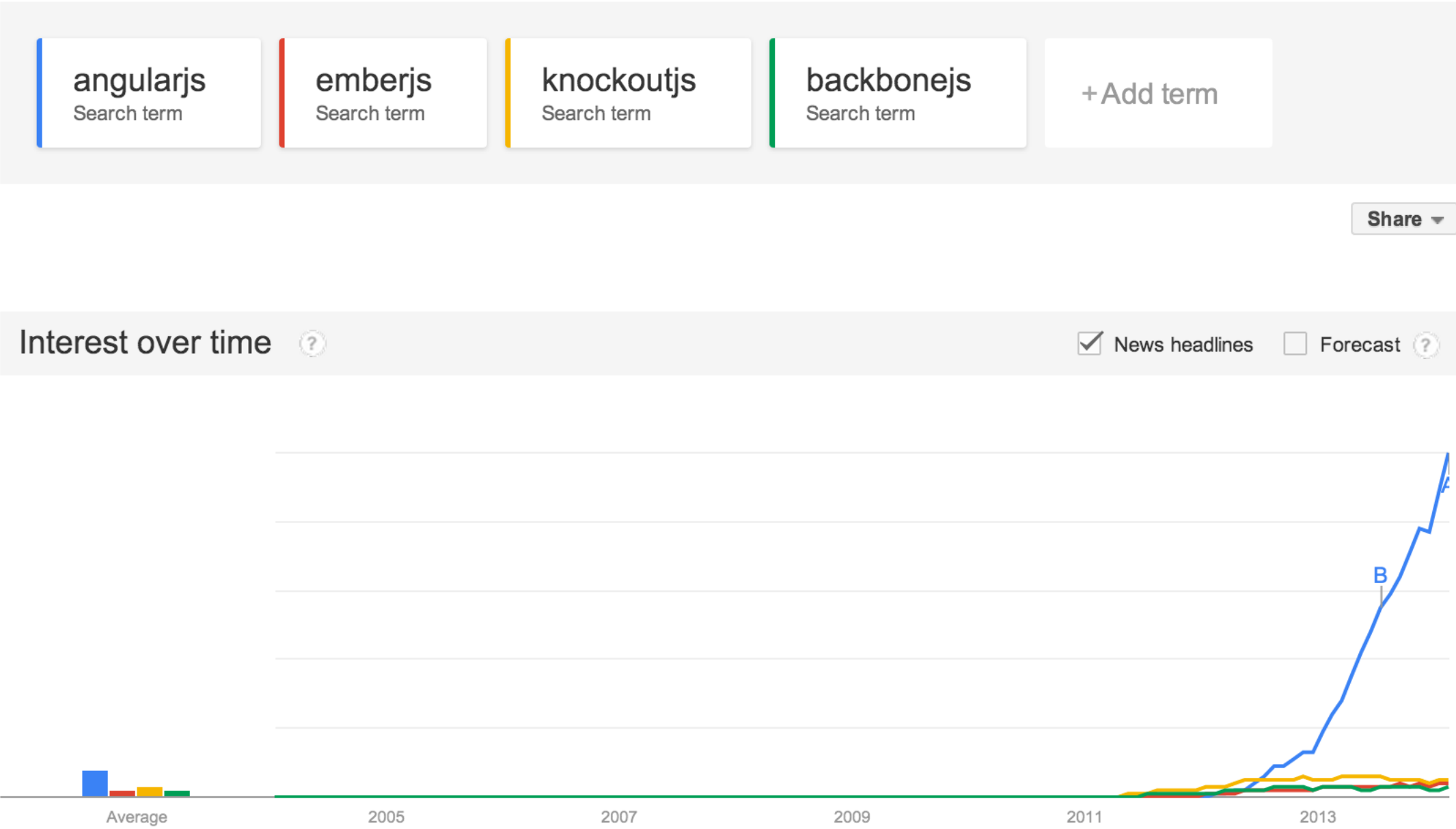
Jobs on Dice.com



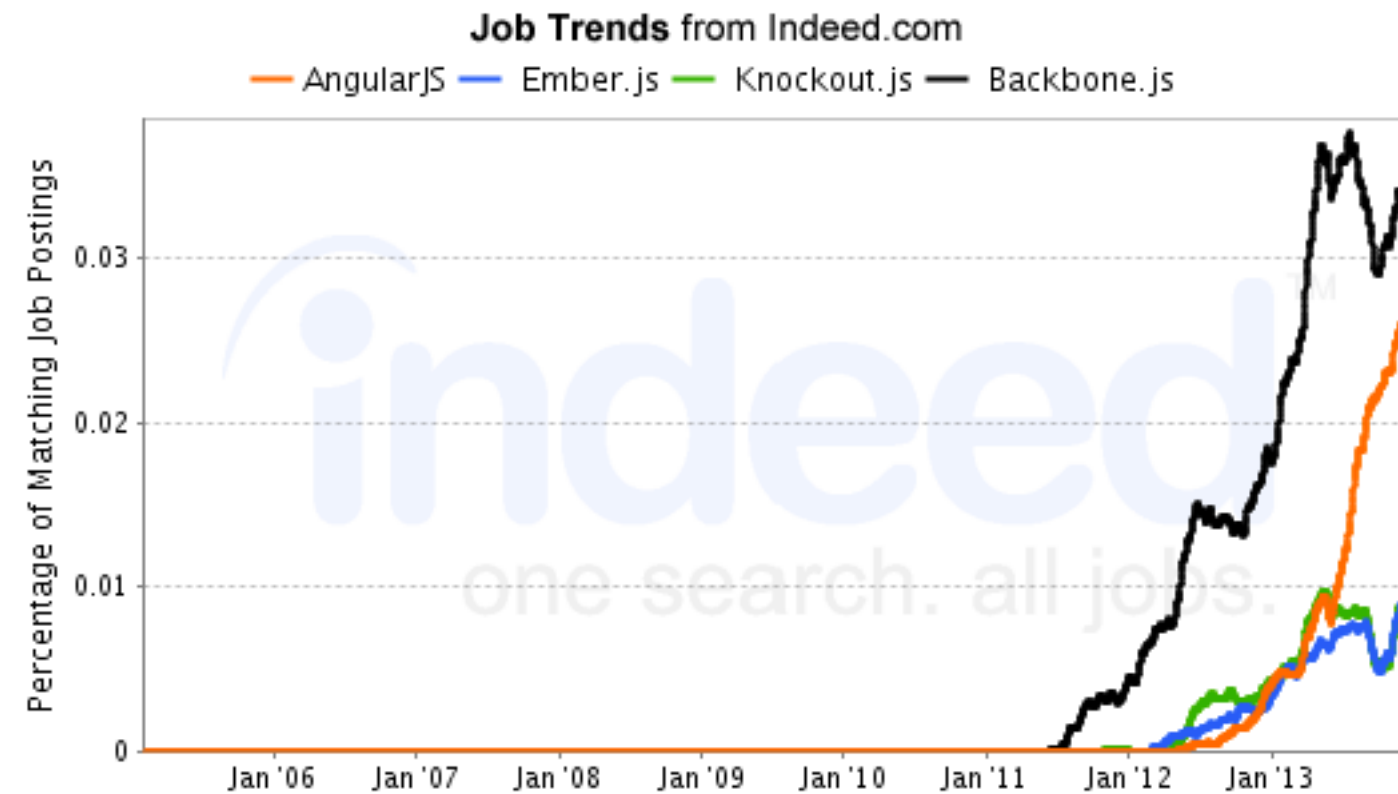
LinkedIn Skills



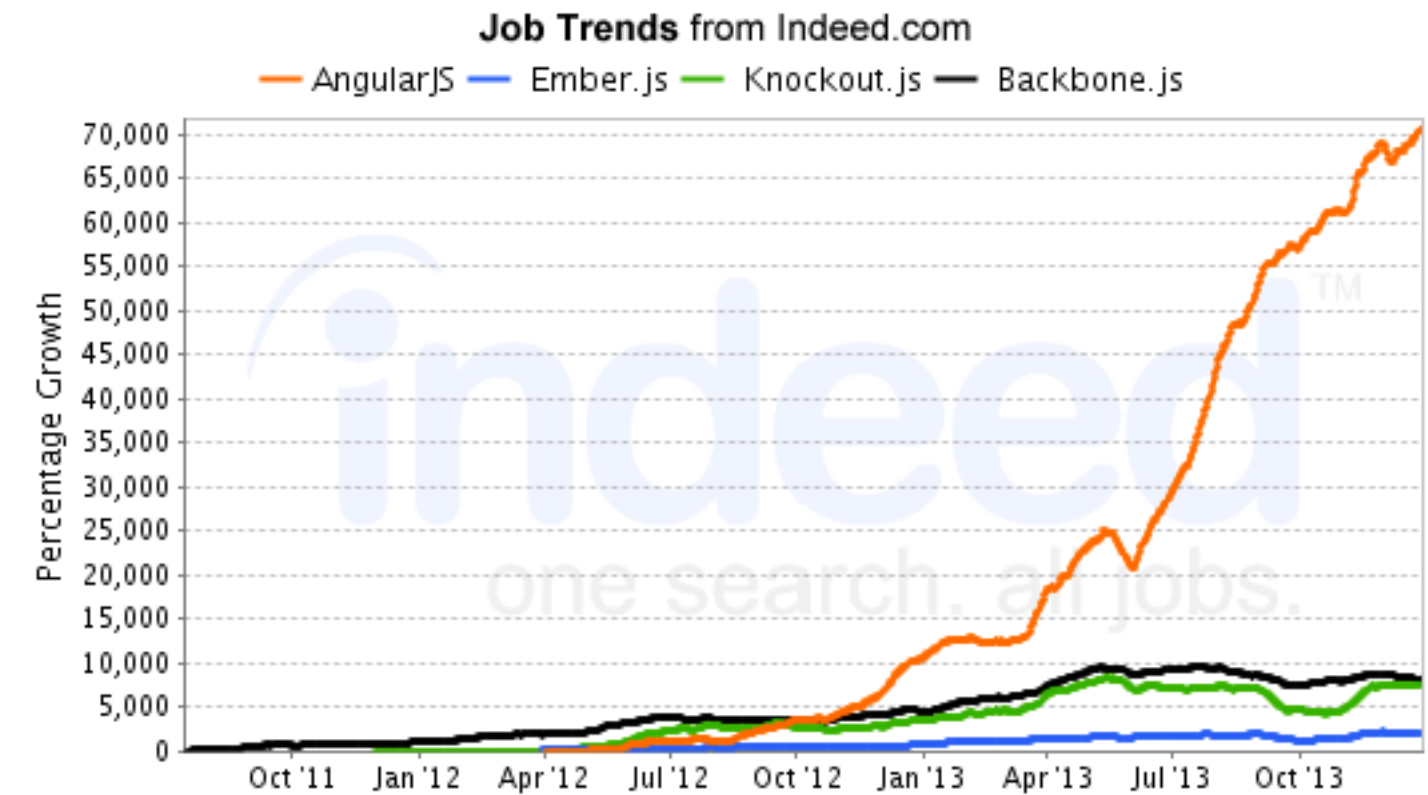
Google Trends



Indeed Job Trends

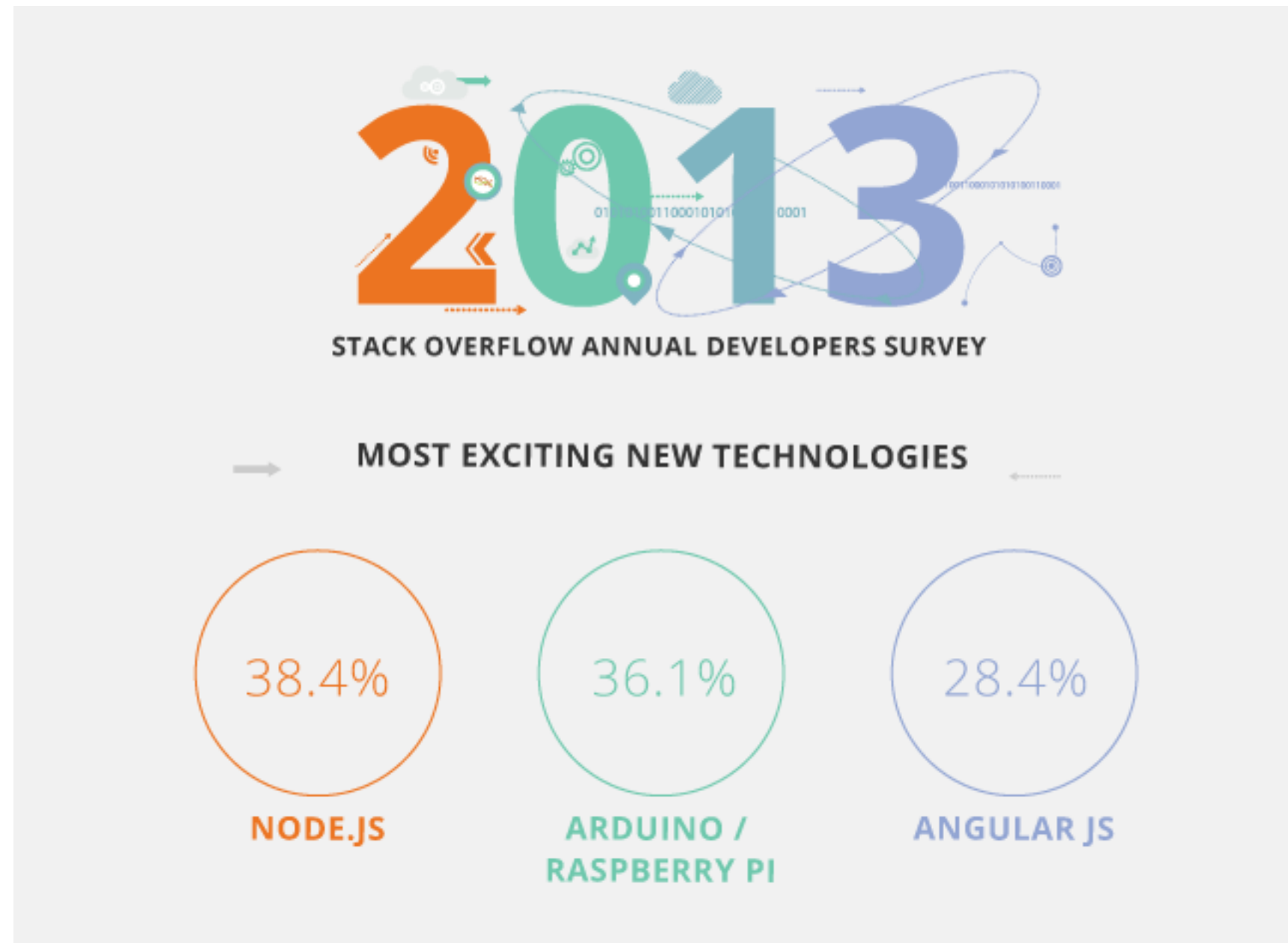


Absolute



Relative

Stack Overflow



Who wants to learn  ANGULARJS?

The History of AngularJS

Started by Miško Hevery in 2009

GWT = 3 developers, 6 months

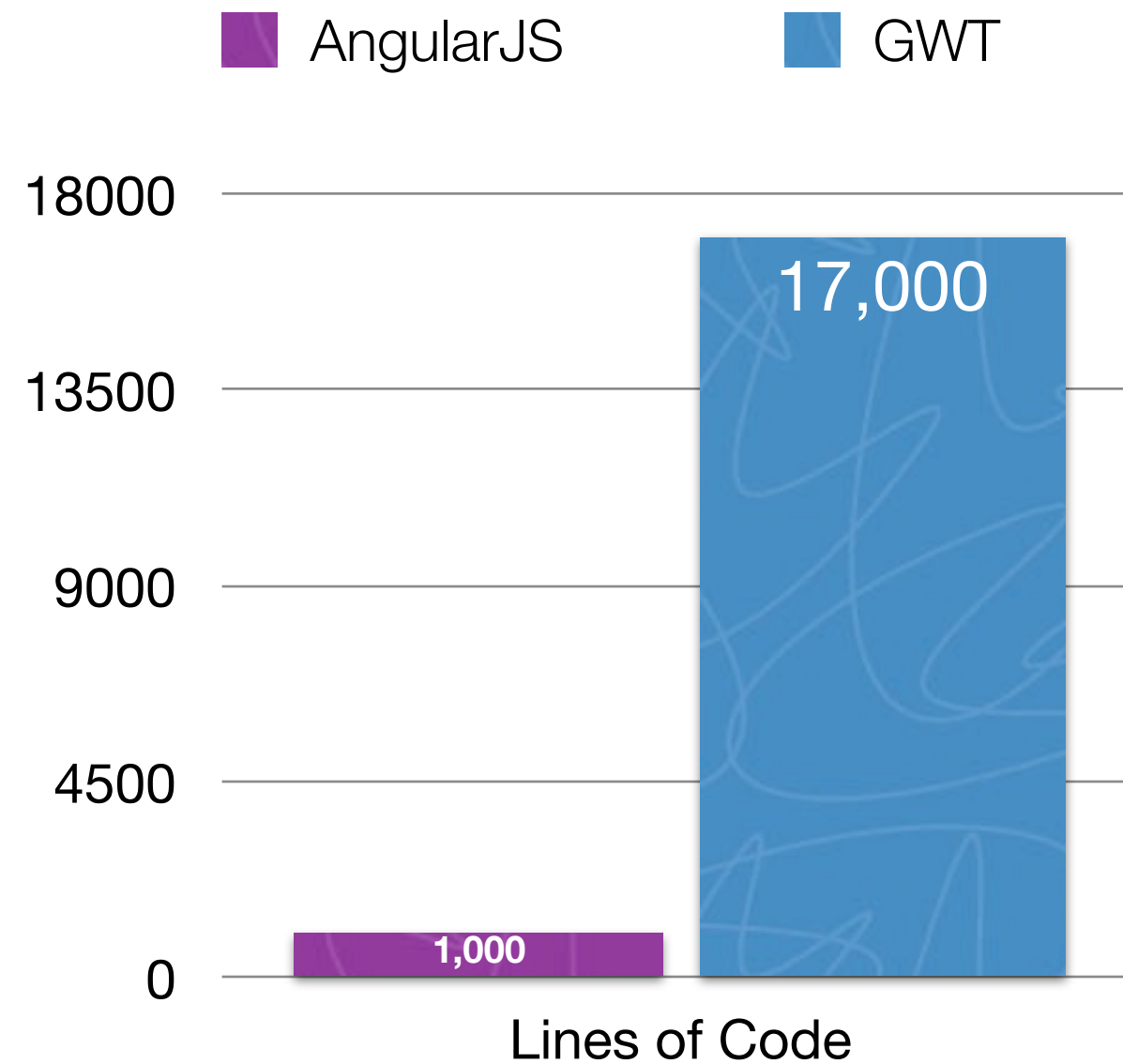
AngularJS = 1 developer, 3 weeks

Learn more:

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



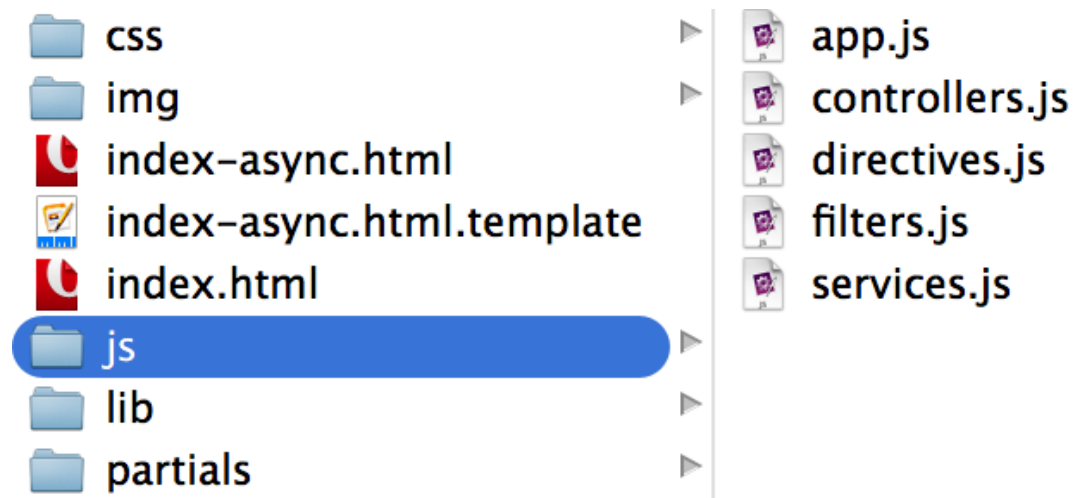
The History of AngularJS



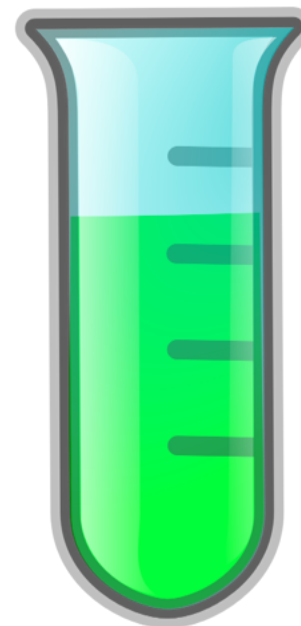
Hello World

```
<!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/1.2.13/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...

App Definition

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

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

App Definition with separate files

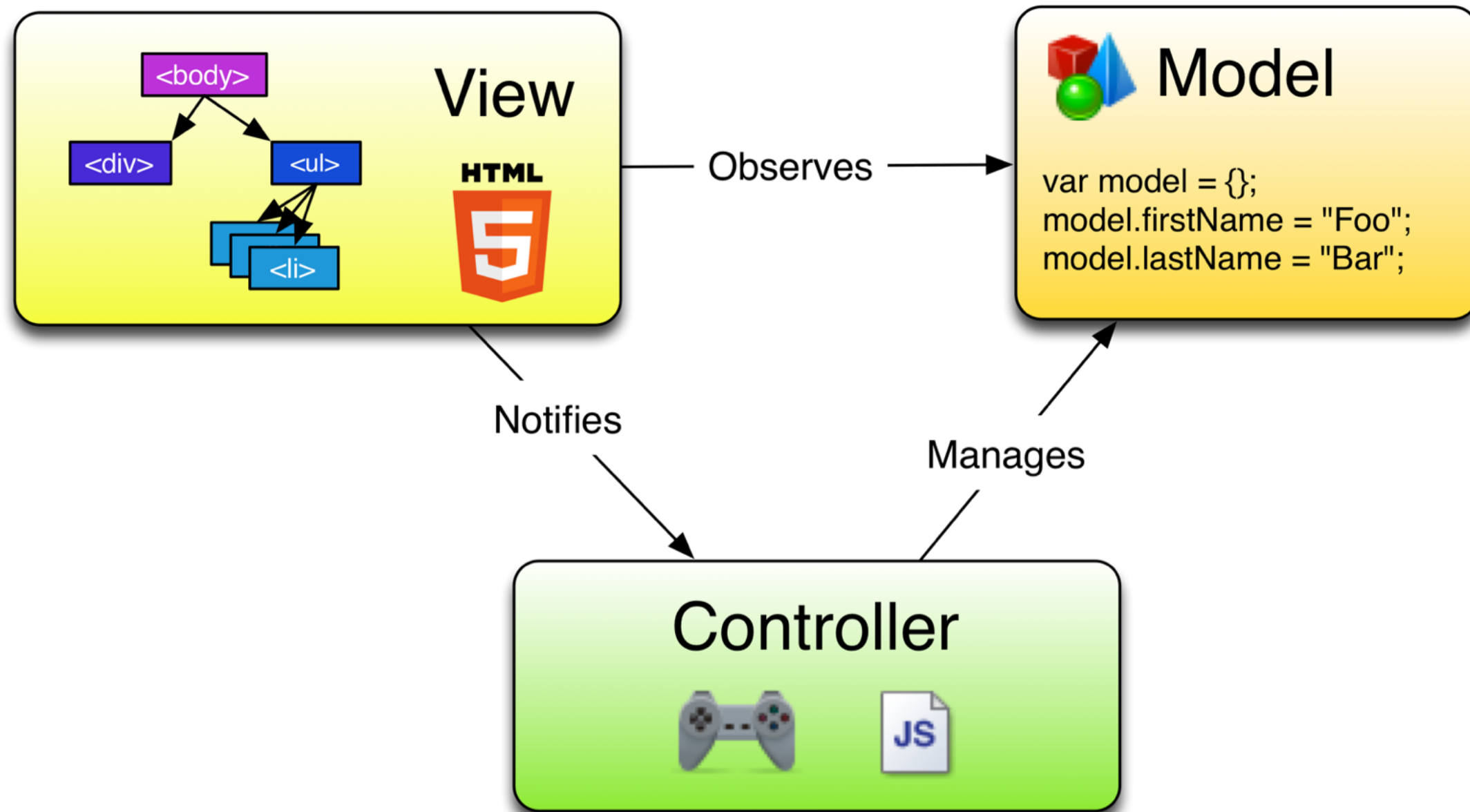
app.js

```
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

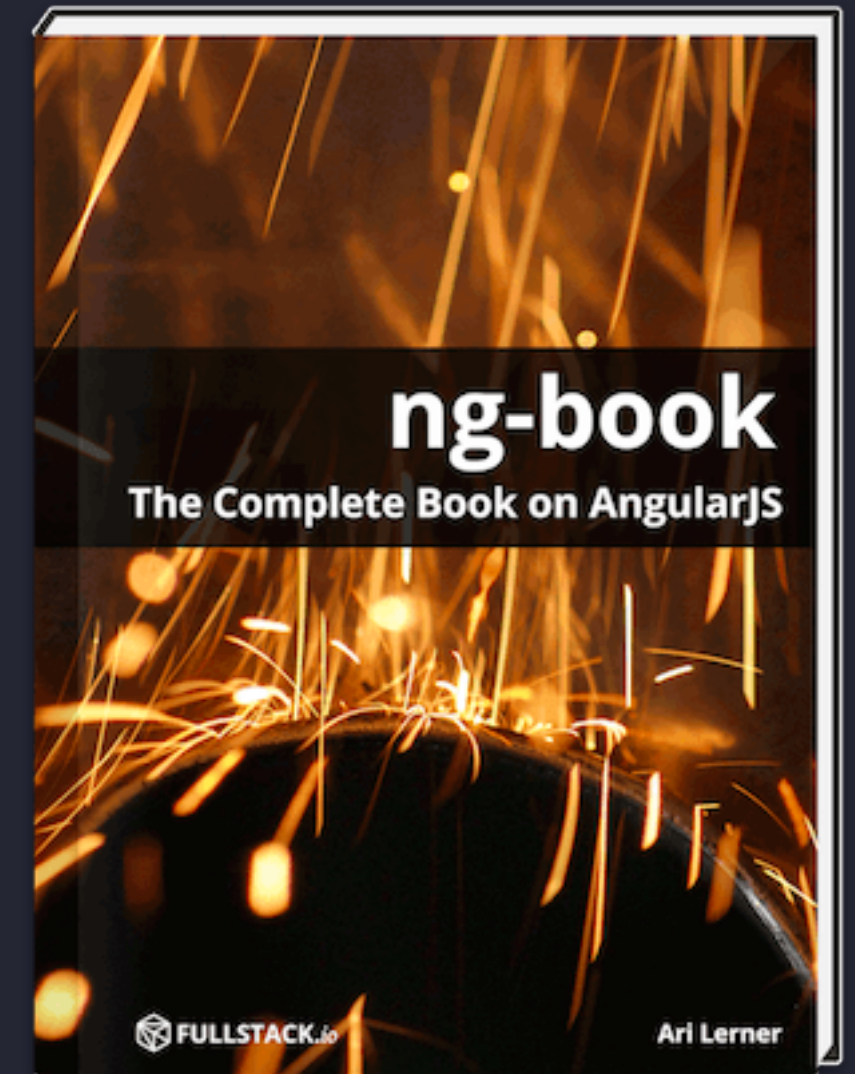
ng-checked

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             # <-- static HTML pages (underscore and handlebars templates are supported)
│       ├── index.us      # <-- a template used to produce the application's index.html
│       └── templates     # <-- client-side 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                # <-- 3rd-party assets will be prepended or merged into the application
    ├── 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
```



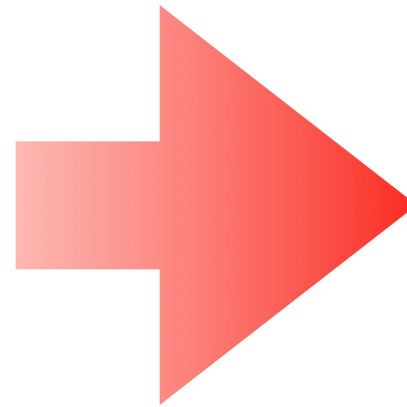
Feature	Lineman	Yeoman
Generators	1 builtin scaffold, clonable templates, no sub-generators	No builtin scaffolds, installable generators, many sub-generators
Command Line Interface	Wrapped utilities that are accessed via CLI ie: <code>lineman grunt</code>	Unwrapped utilities that are accessed via their CLI ie: <code>grunt</code>
HTML5 pushState Simulator	builtin and enabled via <code>config/application</code>	-
API Stubbing Prototyping	builtin and enabled via <code>config/server</code>	-
API Proxy	builtin and enabled via <code>config/application</code>	-
Test Runner	<code>testem</code> , configured for dev and ci mode with <code>lineman spec</code> and <code>lineman spec-ci</code>	-
Grunt Task Lifecycle	preconfigured phases: <code>common</code> , <code>dev</code> , <code>dist</code> , split into <code>config/files</code> and <code>config/application</code>	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 <code>bower</code> , via extensions	<code>bower</code>
SourceMap Generation	builtin, defaults to <code>grunt-concat-sourcemap</code>	-
Dev Server	<code>lineman run</code> , spins up <code>express.js</code>	generator specific
Directory Structure	common across project templates	generator specific
License	MIT	BSD



```

app/
  css/
    app.css
    bootstrap.css
  img/
    phones/
    glyphs-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 AngularJS

https://www.youtube.com/watch?v=UYVcY9EjRs

YouTube

Sublime Text 2

File Edit Selection Find View Goto Tools Project Window Help

login_controller.js — testing-strategies-for-angular

overview.md router.js login_controller_spec.js

```
1 describe("controller: LoginController", function() {
2
3   describe("successfully login in", function() {
4
5   });
6
7 });
8
```

```
1 angular.module("app").controller('LoginController', function($scope, $location, Authentication
2   $scope.credentials = { username: "", password: "" };
3
4   var onLoginSuccess = function() {
5     $location.path('/home');
6   };
7
8   $scope.login = function() {
9     AuthenticationService.login($scope.credentials).success(onLoginSuccess);
10  };
11 });
12
```

git branch: master, index: 27, working: 1* 7* 27, 21 characters selected

15:38 / 1:01:45

CC HD

Testing Strategies for Angular JS

David Mosher · 11 videos

Subscribe

3,782

16,318

142

2

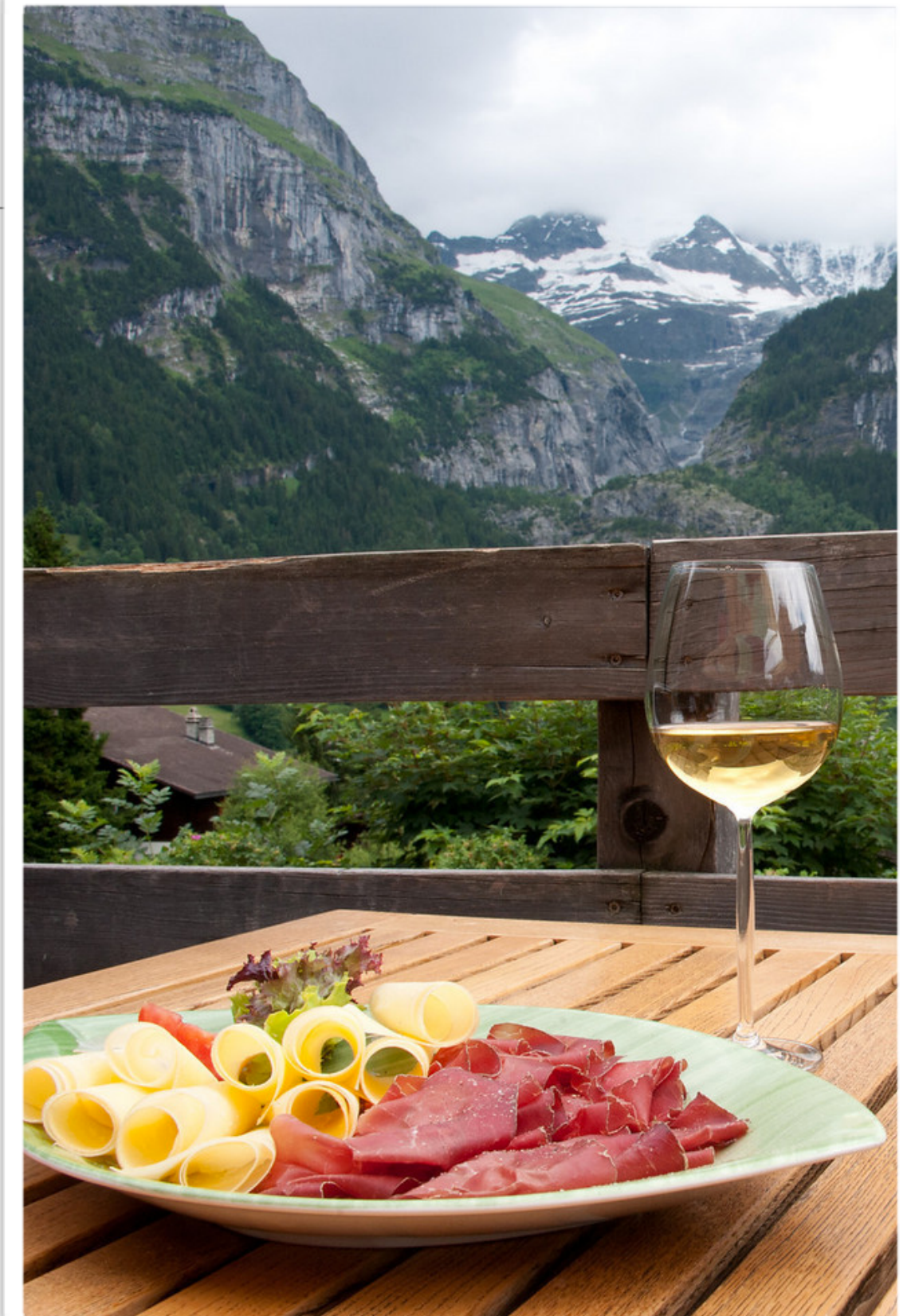
Testing

Testem - 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  
        });  
    });  
});
```



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).toHaveBeenCalled(' /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....](http://raibledesigns.com/rd/entry/using-grunt-with-angularjs-for-front-end-optimization) From 27 requests to 5 on initial page load. #angularjs #grunt

**Mark** @markj9 
[@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>

CloudBees



CloudBees - GrandCentral

https://grandcentral.cloudbees.com/?#CB_clickstart=https://raw.github.com/CloudBees-community/angular-js-clickstart/master/...

ClickStart Apps ClickStart

Click to show core services

Upgrade Jenkins...

Support

Request Support...

View Support Tickets...

Getting Started

- DEV@cloud - Jenkins as a
- RUN@cloud - Deploying J
- apps
- Continuous deployment
- CloudBees for Eclipse

Quick Reference

- DEV@cloud Wiki
- GIT/SVN repositories
- Application monitoring
- Selenium browser testing
- RUN@cloud Wiki
- Maven Integration
- MySQL databases
- Sonar code coverage

AngularJS - HTML enhanced for web apps!

Application Components

- Source repository
- Jenkins build
- Web Application

Name

derailed

Setup a Jenkins running Angular tests (with Chrome and Firefox) and deploy the 'seed' app

Please note: This process will build and deploy a working application into the cloud. The process may take several minutes to complete. The operation is asynchronous so you may leave the page at any time.

Cancel Create App >

Community Links

Try our new UI

Blog Entries

- to do Maven release builds
- DEV@cloud to central
- g Jenkins with Docker
- ker on OS-X
- ing Jenkins Horizontally
- lar questions and answers.
- en: Keeping secrets out of
- ce Control
- ularJS starter app updated
- framework clickstart
- vs from CloudBees...

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-0.10.0.min.js"></script>  
<script src="lib/angular/ui-bootstrap-tpls-0.10.0.min.js"></script>  
  
angular.module('myApp', ['ui.bootstrap']);
```

Ionic Framework

<http://ionicframework.com>



Create amazing apps

The beautiful, open source front-end framework for developing hybrid mobile apps with HTML5.

Download alpha

Get started

MIT Licensed · v0.9.26-alpha "rabbit" · 2014-02-26



[Ionic Weather Demo App](#)

AngularJS Batarang



The screenshot shows a web browser window with the address bar at `localhost:8080/#!/`. The page title is "Angular+Rest+SpringSecurity" and the user is logged in as "admin". The main content area displays a "News" form with a "Create" button and a "Title #0" input field. Below the form are "Remove" and "Edit" buttons. The Batarang extension is open, showing the "Models" tab. The "Scopes" panel on the left shows a tree view of scopes: `< Scope (002)`, `< Scope (003)`, `< Scope (009)`, `< Scope (00A)`, `< Scope (00B)`, `< Scope (00C)`, and `< Scope (00D)`. The "Models for (002)" panel on the right shows a JSON model for a user:

```
{
  hasRole: null
  logout: null
  user: {
    name: admin
    roles: {
      ROLE_USER: true
      ROLE_ADMIN: true
    }
  }
  token: admin:1393391653180:0e9f29207c2faa6c1e4ebee67bf3c49
}
```

My Experience



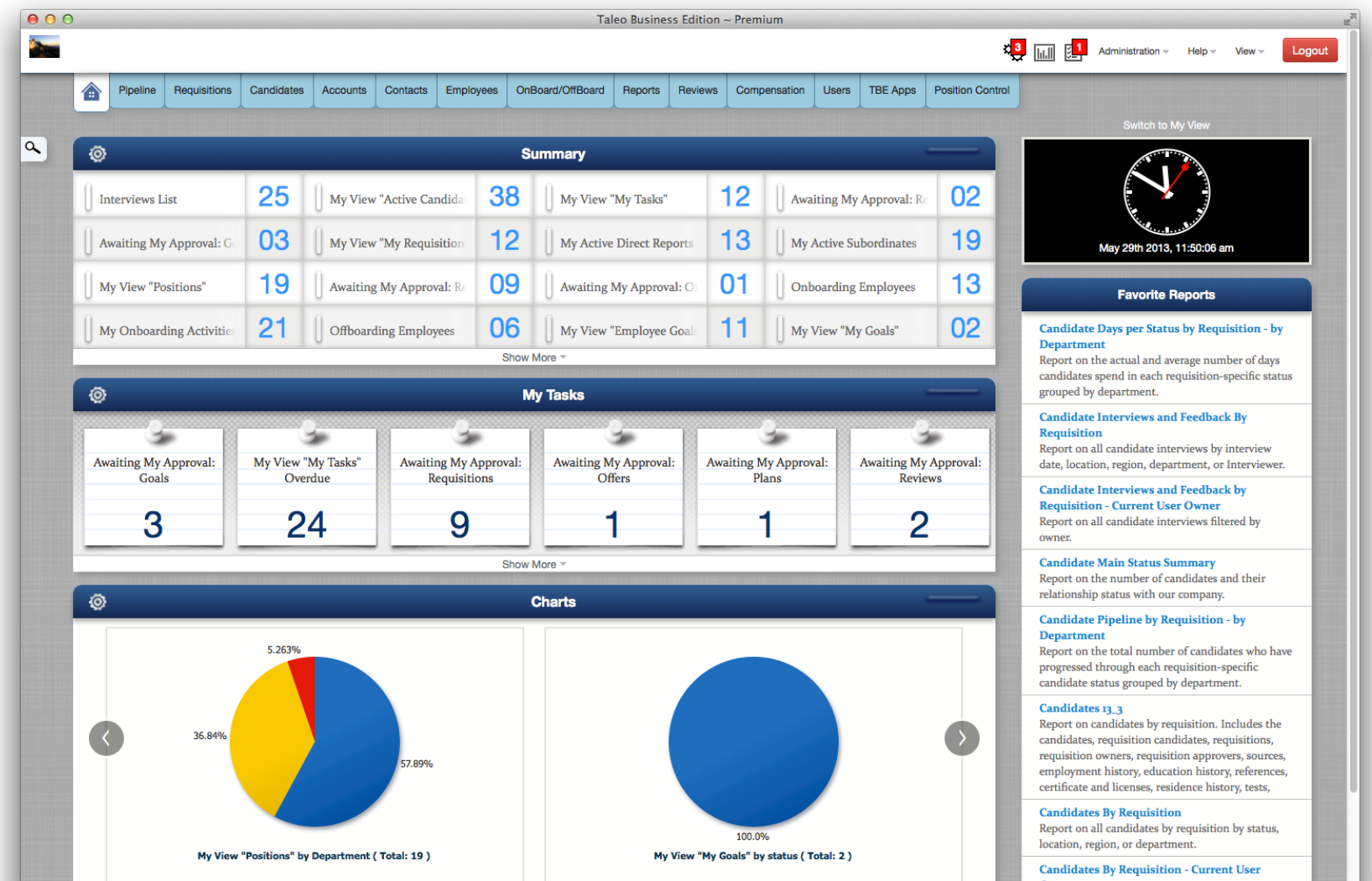
Developing with AngularJS Series

Part I: The Basics

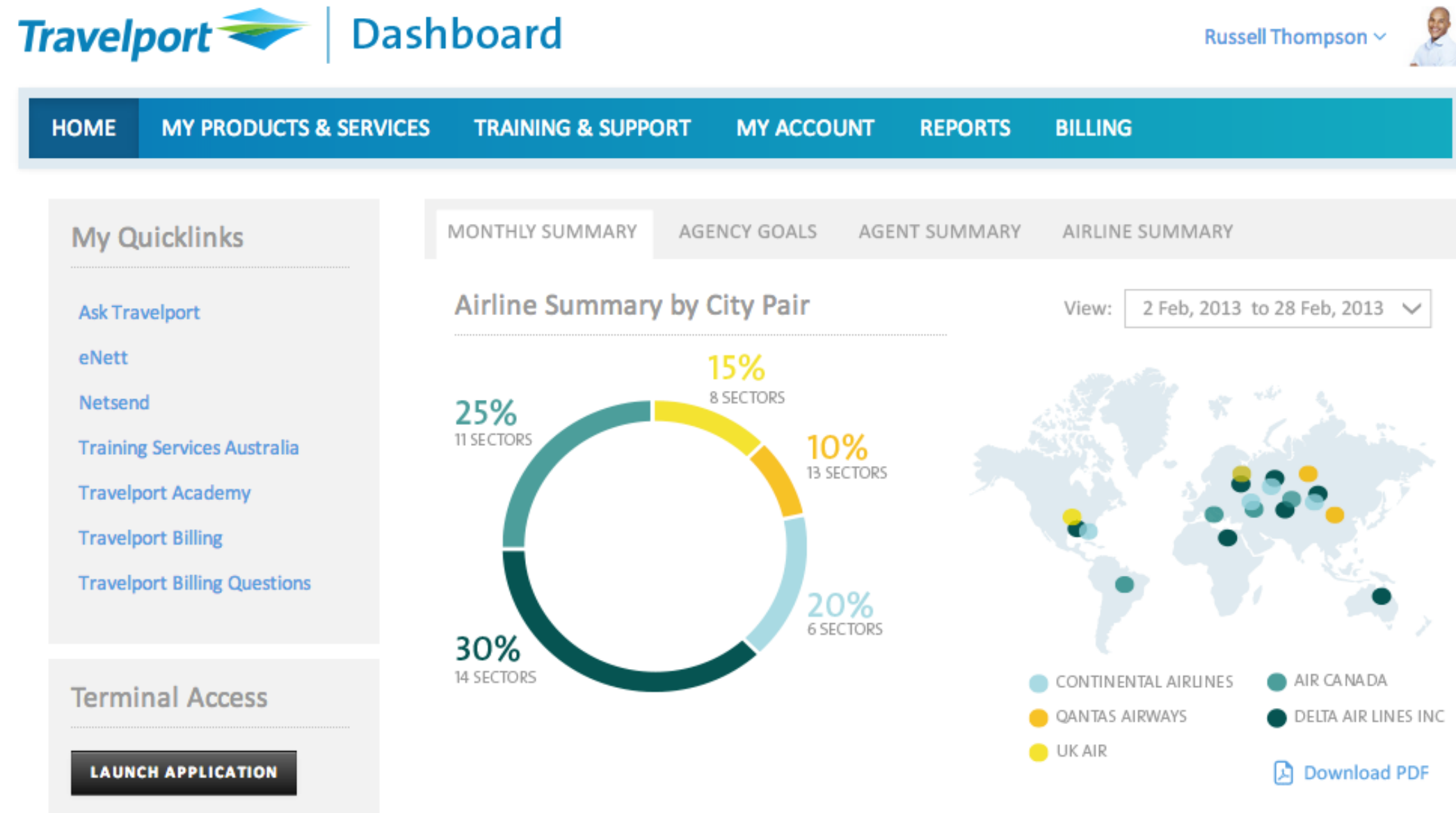
Part II: Dialogs and Data

Part III: Services

Part IV: Making it Pop



My Experience



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

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

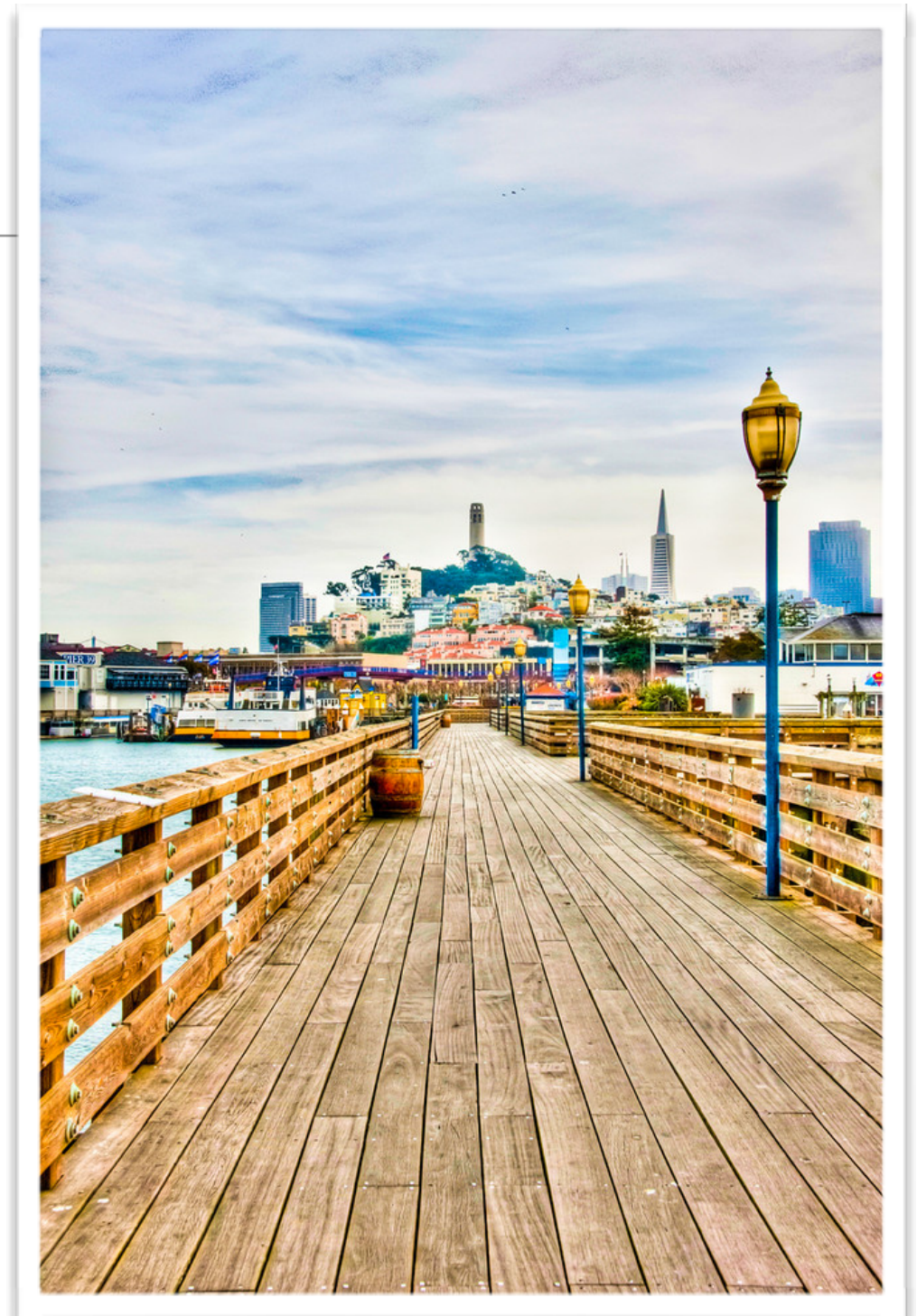
Get the supplies you will need

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

Presentations

 <http://slideshare.net/mraible>

Code

 <http://github.com/mraible>



Who to follow on Twitter



AngularJS Team at Google

Miško Hevery - [@mhevery](https://twitter.com/mhevery)

Igor Minar - [@IgorMinar](https://twitter.com/IgorMinar)

Brian Ford - [@briantford](https://twitter.com/briantford)

Web Performance

Ilya Grigorik - [@igrigorik](https://twitter.com/igrigorik)

Andy Davis - [@andydavies](https://twitter.com/andydavies)

Steve Souders - [@Souders](https://twitter.com/Souders)

Resources



Angular Dart

<https://angulardart.org>

Devovx AngularJS Talks on Parleys.com



<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>

