

springone 

DALLAS
2014

Creating Modular Test Driven SPAs with Spring And AngularJS

By Gunnar Hillert - [@ghillert](https://twitter.com/@ghillert)

Goals

- AngularJS Introduction
- Build and Deployment
- Integration with Spring
- Modularization
- Testing
- UI Considerations



Me

- (Java) Web developer since 2005
- Struts 1+2, Spring MVC, GWT, Flex
- Spring Integration + XD committer
- AngularJS since Jan 2014
- Co-organize 



spring



StrutsTM



apache
flex

 **ANGULARJS**
by Google

Non-screen activities



AngularJS Introduction

Audience - What do you use*?

- AngularJS **60%**
- Backbone **20%**
- JQuery **80%**
- Are you using any other SPA Framework? **Ember.js 1 user**
- Spring MVC **80%**
- Spring Boot **20%**

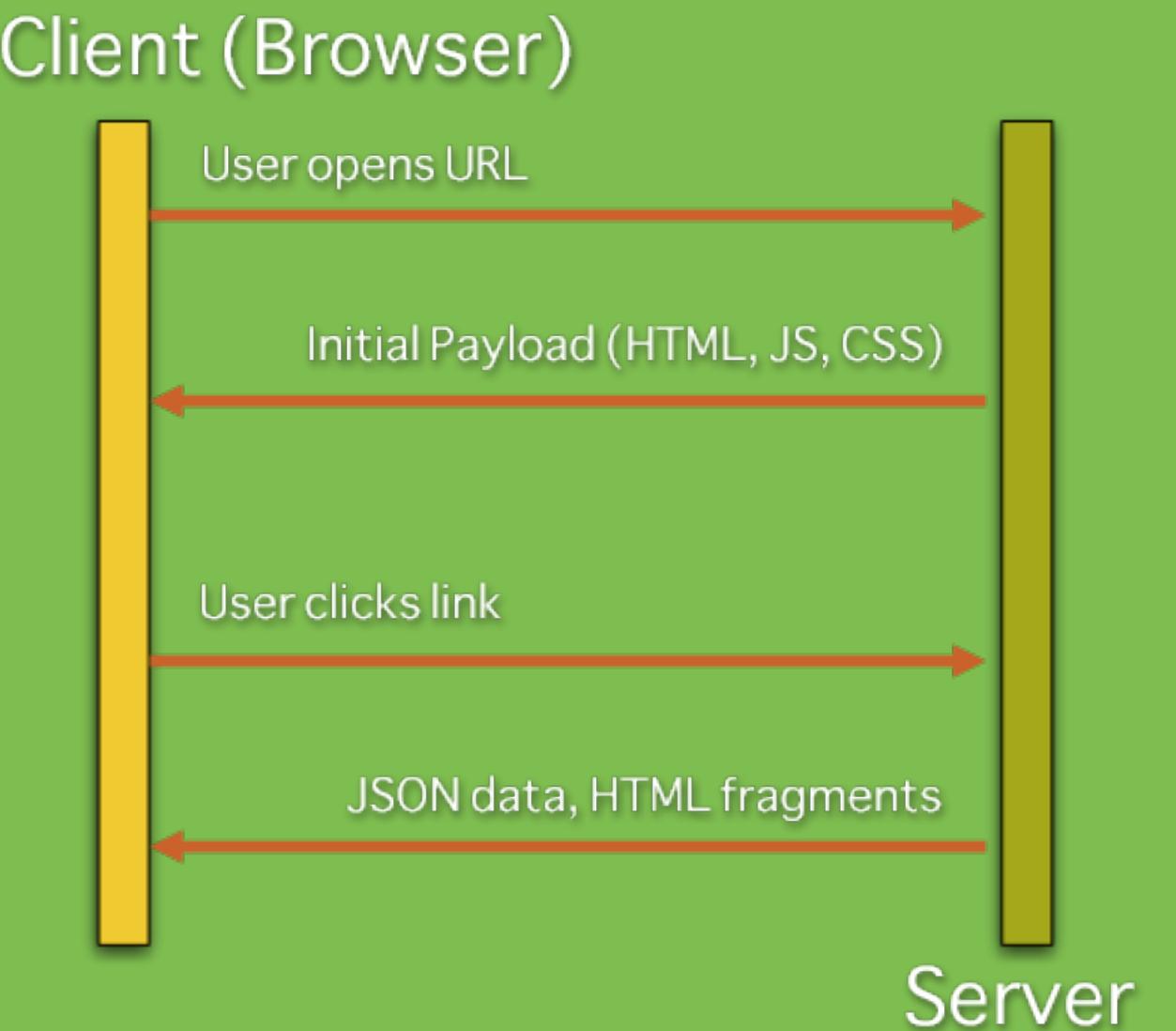
* Recorded from memory

What are SPAs?

A single-page application (SPA), also known as single-page interface (SPI), is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application.

Wikipedia

What are SPAs?



JavaScript WTF

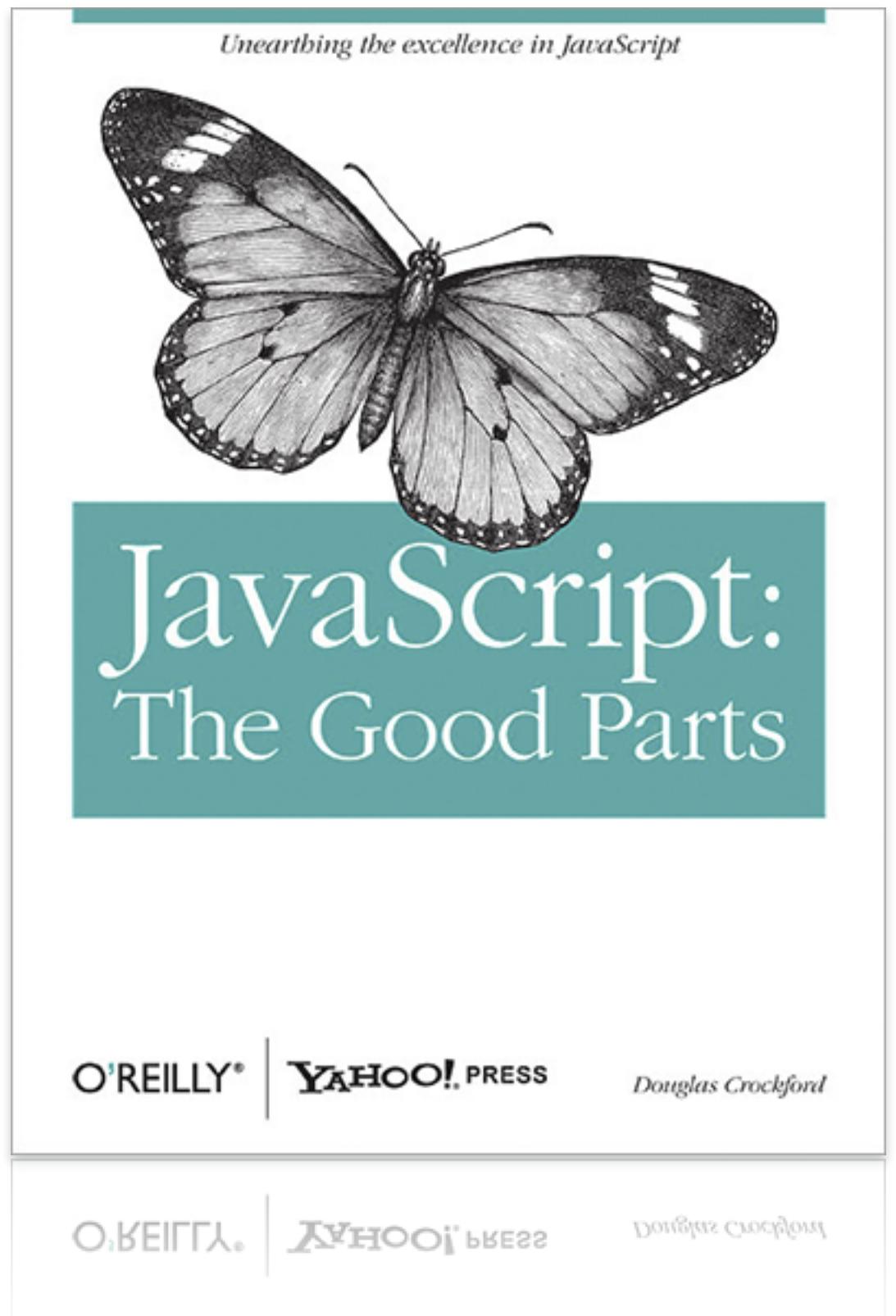
- <http://wtfjs.com/>

```
parseInt('crap');           // NaN
parseInt('crap', 16); // 12

(2 + "3");    // 23
(2 + +"3"); // 5
(+ "");      // 0
```



Read this



From Backbone to Angular

- Too many moving parts, choices
- Boilerplate Code
- Marionette, Backbone.ModelBinder, Backbone.Relational



BACKBONE.JS

Alternatives



Knockout.

AngularJS Basics

- Model
- View (Templates)
- Controller
- Dependency Injection
- Expressions
- Filters
- Directives
- Routing
- Modules
- See also: [AngularJS Concepts](#)

Model

- Angular is very flexible about your model
- Ultimately expressed via the `$scope`
- `$scope` = Glue between Controller and View
- `$scope` mimics DOM (Hierarchical, one `$rootScope`)

Model

- Killer Feature: Data-Binding
- Model === single-source-of-truth
- View reflects model changes automatically
- `$watch`, `$apply`

View

- HTML is your templating Engine
- Minimize logic as much as possible
- Consider Custom Directives

¡Hola!

- Demo

```
<div ng-app ng-init="firstName='Eric';lastName='Cartman'">
  <div>
    First Name: <input type="text" ng-model="firstName">
  </div>
  <div>
    Last Name: <input type="text" ng-model="lastName">
  </div>
  <div>
    <b>Complete Name:</b> {{firstName + ' ' + lastName | uppercase}}
  </div>
</div>
```

Controller

- Used to "setup" your `$scope` (values)
- Add behavior to your `$scope` (functions)
- Don't do UI work using controllers!!
- Use directives and filters instead

¡Hola! v2.0 - View

- [Demo](#)

```
<div ng-app="hola" ng-controller="NameController">
  <div>
    First Name: <input type="text" ng-model="firstName">
  </div>
  <div>
    Last Name: <input type="text" ng-model="lastName">
  </div>
  <div>
    <b>Complete Name:</b> {{firstName + ' ' + lastName | uppercase}}
  </div>
</div>
```

¡Hola! v2.0 - Controller

- Demo

```
<script>
  (function () {
    var app = angular.module('hola', []);
    app.controller('NameController', function ($scope) {
      $scope.firstName='Angular';
      $scope.lastName='rocks';
    });
  })();
</script>
```

Dependency Injection

- Consider using array notation:

```
app.controller('NameCtrl', function($scope){ ... });
```

```
app.controller('NameCtrl', ['$scope', function($scope){ ... }]);
```

- Or use ~~ngmin~~ [ng-annotate](#)
 - ~~grunt-ngmin~~, ~~gulp-ngmin~~, ~~grunt-ng-annotate~~, ~~gulp-ng-annotate~~

Expressions

- `{{ expression }}`
- No Control Flow Statements
- Can use filters inside expressions:
- `{{ 'abcd' | uppercase }}`

Filters



```
...
<tr ng-repeat=
  "item in jobDefinitions | filter:filterQuery
  | orderBy:'name'">
...

```

Directives

- Are markers on a DOM element
- Attach behavior/transform DOM elements
- ng-controller, ng-app ...

Types of Directives

- Attribute (default)
- Element
- Class
- See: <https://gist.github.com/CMCDragonkai/6282750>

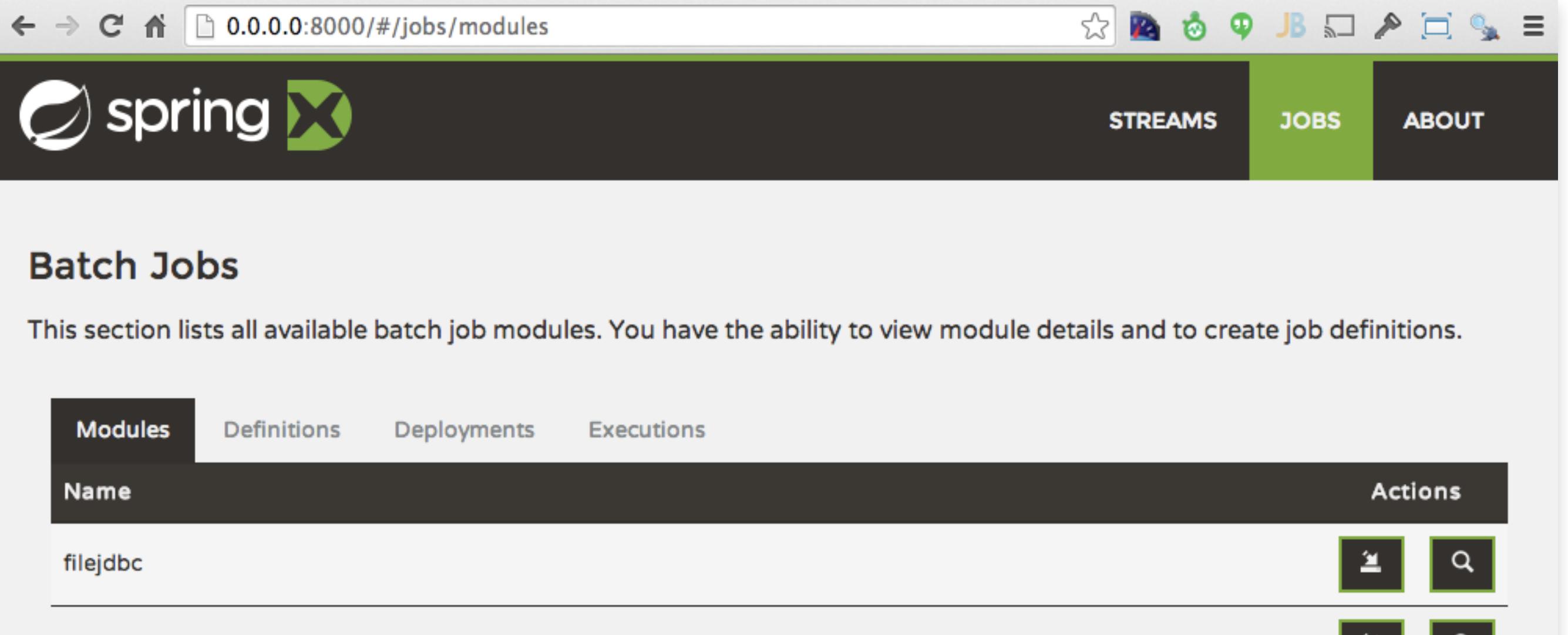
Routing

- ngRoute (built-in)
- Routing on steroids using [ui-router](#)



Routing using UI-Router

- state machine
- nested views
- Spring XD's [routes.js](#)



The screenshot shows the Spring XD web interface at the URL `0.0.0.0:8000/#/jobs/modules`. The top navigation bar includes links for STREAMS, JOBS (which is highlighted in green), and ABOUT. Below the header, the title "Batch Jobs" is displayed, followed by a descriptive text: "This section lists all available batch job modules. You have the ability to view module details and to create job definitions." A table with tabs for Modules, Definitions, Deployments, and Executions is shown. The "Modules" tab is active, displaying a single row for "filejdbc". The table has columns for Name and Actions, with icons for edit and search.

Name	Actions
filejdbc	 

Modules

- Hang on tight...



botanic | NG

Build and Deployment

Build and Deployment

- Do everything via Maven and Gradle?
- What about Non-Java Tooling?
- Consider Web Resource Optimization

Web Resources Optimization

- Minification
- Merging
- Compression
- Caching (and Cache busting)

Web Resources Optimization

Statistics The page has a total of **43** HTTP requests and a total weight of **2580.9K** bytes with empty cache

WEIGHT GRAPHS

HTTP Requests - 43
Total Weight - 2580.9K

1 HTML/Text	6.4K
39 JavaScript File	2222.9K
2 Stylesheet File	336.4K
1 Favicon	15.0K

HTTP Requests - 41
Total Weight - 1283.8K

1 HTML/Text	0.0K
37 JavaScript File	1079.9K
2 Stylesheet File	203.9K
1 Favicon	0.0K

Grade C Overall performance score 71

[ALL \(23\)](#) FILTER BY: [CONTENT \(6\)](#) | [COOKIE](#)

F	Make fewer HTTP requests
F	Use a Content Delivery Network (CDN)
A	Avoid empty src or href
F	Add Expires headers

Statistics The page has a total of **10** HTTP requests and a total weight of **221.6K** bytes with empty cache

WEIGHT GRAPHS

HTTP Requests - 10
Total Weight - 221.6K

1 HTML/Text	1.4K
5 JavaScript File	183.1K
2 Stylesheet File	36.0K
1 Image	0.03K
1 Favicon	0.9K

HTTP Requests - 9
Total Weight - 191.4K

1 HTML/Text	1.4K
4 JavaScript File	152.9K
2 Stylesheet File	36.0K
1 Image	0.0K
1 Favicon	0.9K

Grade B Overall performance score 88

[ALL \(23\)](#) FILTER BY: [CONTENT \(6\)](#) | [COOKIE](#)

A	Make fewer HTTP requests
F	Use a Content Delivery Network (CDN)
A	Avoid empty src or href
F	Add Expires headers
A	Compress components with gzip
A	Put CSS at top
A	Put JavaScript at bottom

Strategies - Java Tooling

- [Wro4j](#)
- [Jawr](#)
- Spring 4.1
 - Flexible resolution and transformation of static web resources
 - See [Blog Post](#)
- [WebJars](#)



Strategies - JavaScript Tooling

- Node (Npm)
- Grunt (Gulp)
- Bower
- Yeoman (angular-seed)



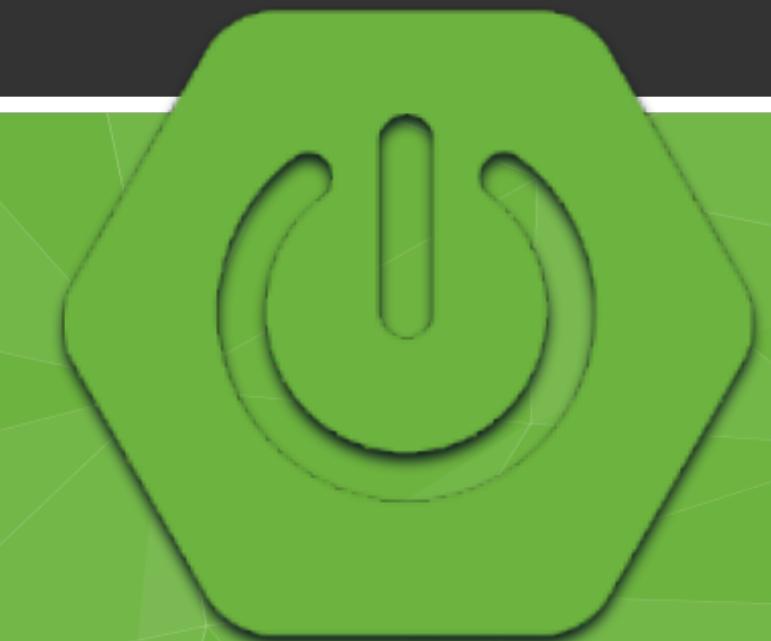
YEOMAN

Make Maven and Gradle Grunt

- Plugins exist for Gradle and Maven
- Spring XD uses Gradle integration
- botanic-ng uses Maven integration
- Spring Boot plus Maven Frontend Plugin



Integration with Spring (Boot)



Hello World fits into Tweet

```
@Controller
class ThisWillActuallyRun
{
    @RequestMapping("/")
    @ResponseBody
    String home() {
        "Hello World!"
    }
}
```

Rapid Prototyping

- Spring Scripts (Samples)
- Starter POMs
- Über-Jars support (can create WARs also)
- Maven + Gradle Plugins
- AutoConfiguration support

Main is BACK

```
@EnableAutoConfiguration @ComponentScan @EnableScheduling
public class MainApp extends RepositoryRestMvcConfiguration {
    @Override
    protected void configureRepositoryRestConfiguration(
        RepositoryRestConfiguration config) {
        config.exposeIdsFor(Image.class, Garden.class, Plant.class);
        config.setBaseUri(URI.create("/api"));
    }
    public static void main(String[] args) {
        final ConfigurableApplicationContext context =
            SpringApplication.run(MainApp.class, args);
        ...
    }
    @Bean
    MultipartConfigElement multipartConfigElement() { ... } ...
}
```

Security

- Best strategy in regards to plugging in Spring Security?
- Authentication and Authorization
- How does it affect Testing
- Consider [Spring Session](#)
- org.springframework.session.web.http.HttpSessionStrategy
 - HeaderHttpSessionStrategy (x-auth-token)
 - CookieHttpSessionStrategy

Serving Static Content

- /META-INF/resources/
- /resources/
- /static/
- /public/
- Also supports WebJars
- Make Boot modules (UI) Pluggable

Demo Backend

Modularization

Modularization Now

- AngularJS Modules
- RequireJS

AngularJS Modules

- Compartmentalize sections of your application
- Does not deal with script loading
- <https://docs.angularjs.org/guide/module>

```
angular.module('myModule', []).
  config(function(injectables) { // provider-injector
    // This is an example of config block.
  }).
  run(function(injectables) { // instance-injector
    // Like a Main method
  });

```

RequireJS

- RequireJS
- JavaScript file and module loader
- RequireJS Optimizer

```
require.config({  
    paths: {  
        angular: '../lib/angular/angular',  
        jquery: '../lib/jquery/jquery',  
        bootstrap: '../lib/bootstrap/bootstrap',  
        ...  
    },  
    shim: {  
        angular: {  
            exports: 'angular'  
        },  
        bootstrap: {  
            deps: ['jquery']  
        }  
    }  
}) ;
```

Modularization Future

- ECMAScript 6 modules
- AngularJS 2 is being written in ES6
- Web Components



Componentization using Directives

- [angular-masonry](#)
- cgBusy
- ngGrowl
- angular-google-maps
- angular-leaflet-directive
- [AngularUI](#)
 - Bootstrap directives



File Upload

- [angular-file-upload](#) (nervgh)
- [angular-file-upload](#) (danialfarid)
- FileReader
- Traditional Post

Select files

Base drop zone

Another drop zone with its own settings

Choose Files No file chosen

Uploads only images (with canvas preview)

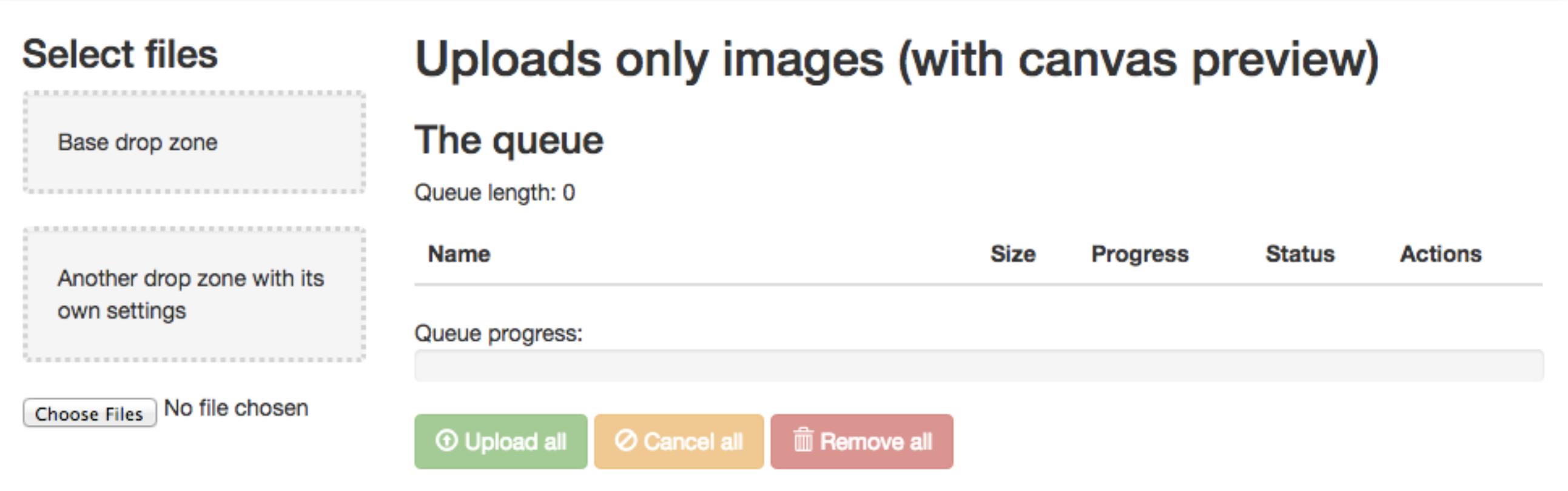
The queue

Queue length: 0

Name	Size	Progress	Status	Actions

Queue progress:

Upload all Cancel all Remove all



Testing

Testing

- E2E testing with [Protractor](#)
- Unit Testing using [Karma](#) and [Jasmine](#)
- Consider using [SauceLabs](#)

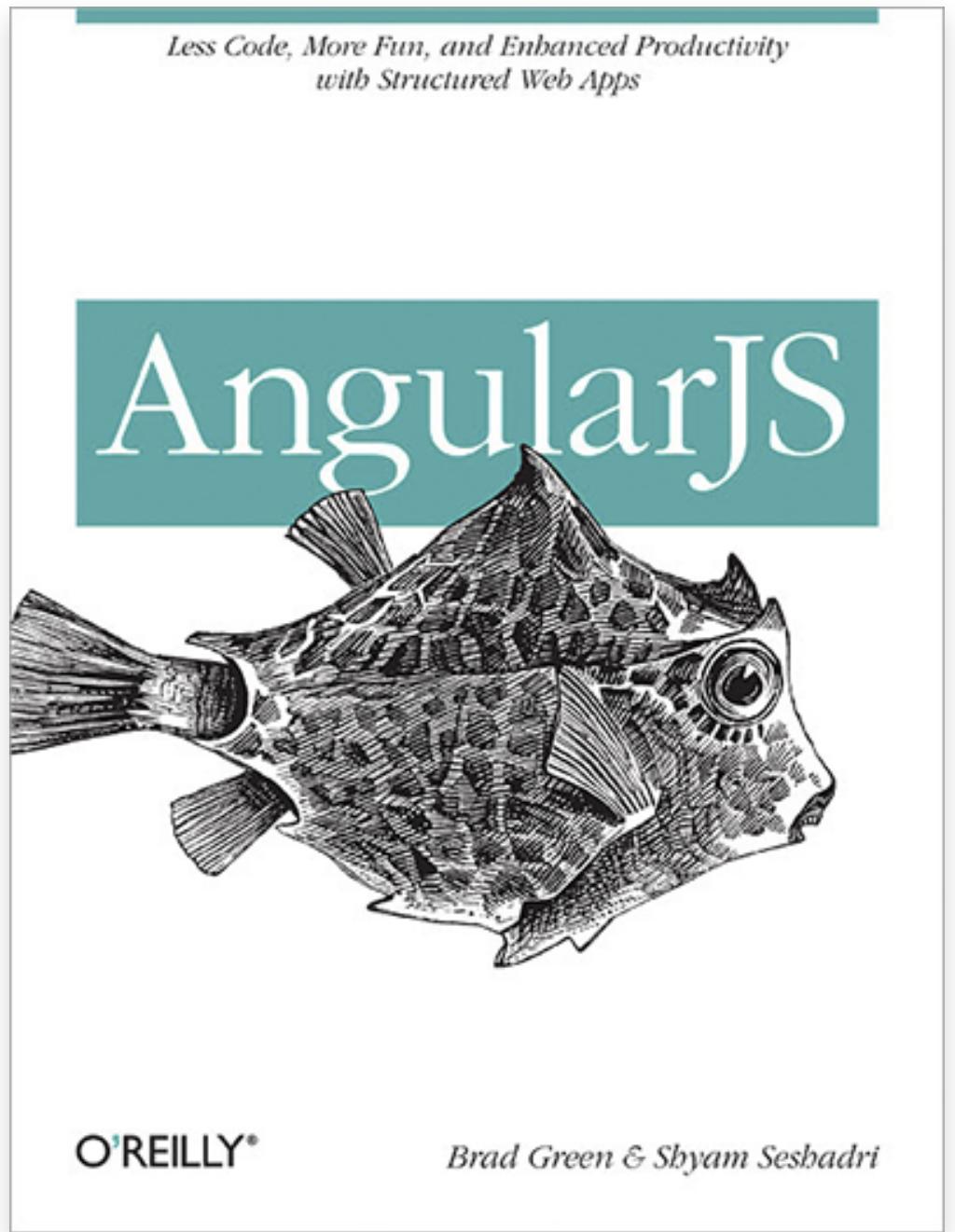
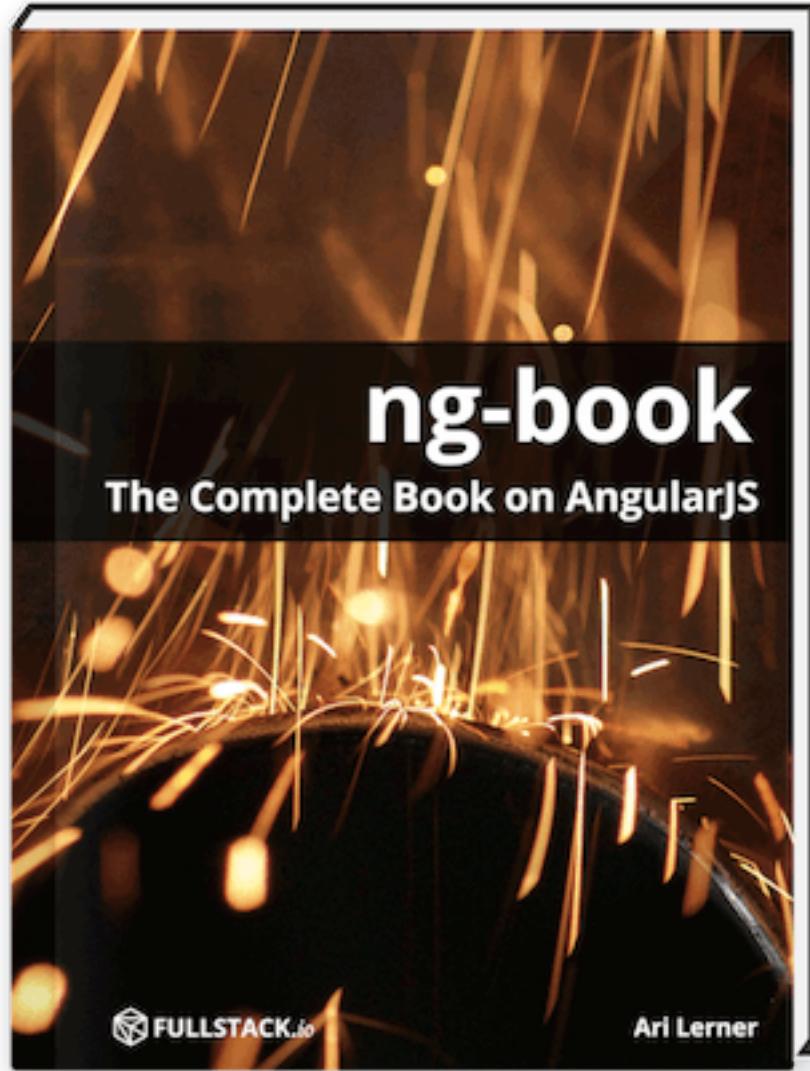
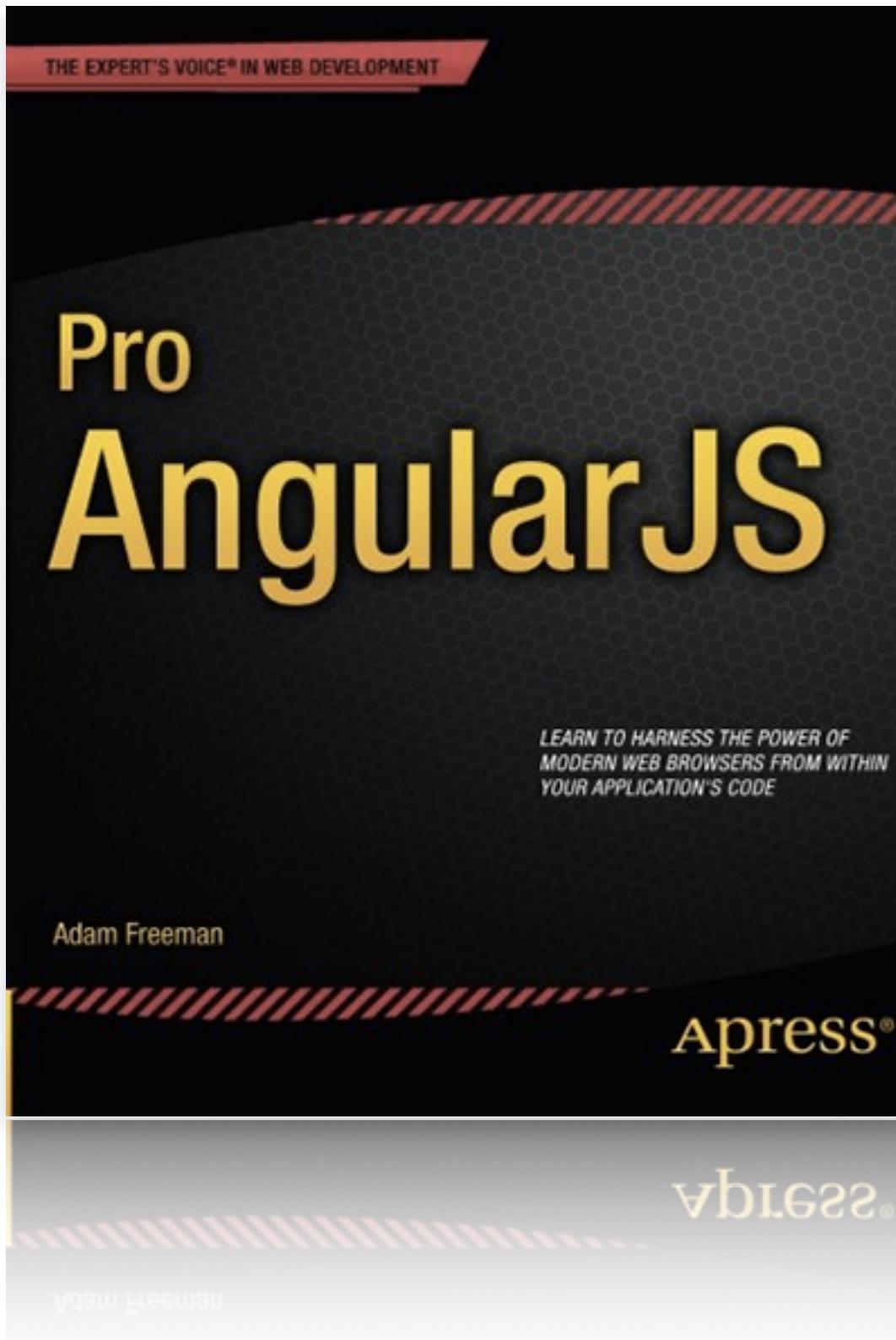
UI Considerations

UI Consideration

- Bootstrap (It is the baseline)
- Keep your CSS maintainable with Less and Sass
- Check your production results with YSlow and PageSpeed
- Load your site from different corners of the planet using
<http://www.webpagetest.org/>

Resources

Books



Videos and More

- Angular JS [Website](#) - Tutorial, Docs
- Angular JS [Youtube Channel](#)
- **ng-conf** has almost [30 session videos](#)
- [Shaping up with Angular JS](#) (Videos + Course)
- Dan Wahlin - [Videos](#) and [Blog](#)
 - [AngularJS Fundamentals In 60-ish Minutes](#)
- <https://egghead.io/>
- Ben Nadel [Blog](#)
- [Year of Moo](#)



ng-conf

Thank You!

Source Code + Preso:

<https://github.com/ghillert/botanic-ng>

Related Talks

Spring 4 Web Applications (R. Stoyanchev)

Deep dive into Spring WebSockets (S. Almar)

Spring Boot for the Web Tier (D. Syer/P. Webb)

Resource Handling in Spring MVC 4.1 (B. Clozel/R. Stoyanchev)

Introducing RaveJS (J. Hann)

Great single page apps need great backends (Adib Saikali)

