CREATING MODULAR TEST DRIVEN SPAS WITH SPRING AND ANGULAR JS

Created by Gunnar Hillert / @ghillert

GOALS

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

ME

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

AUDIENCE - WHAT DO YOU USE?

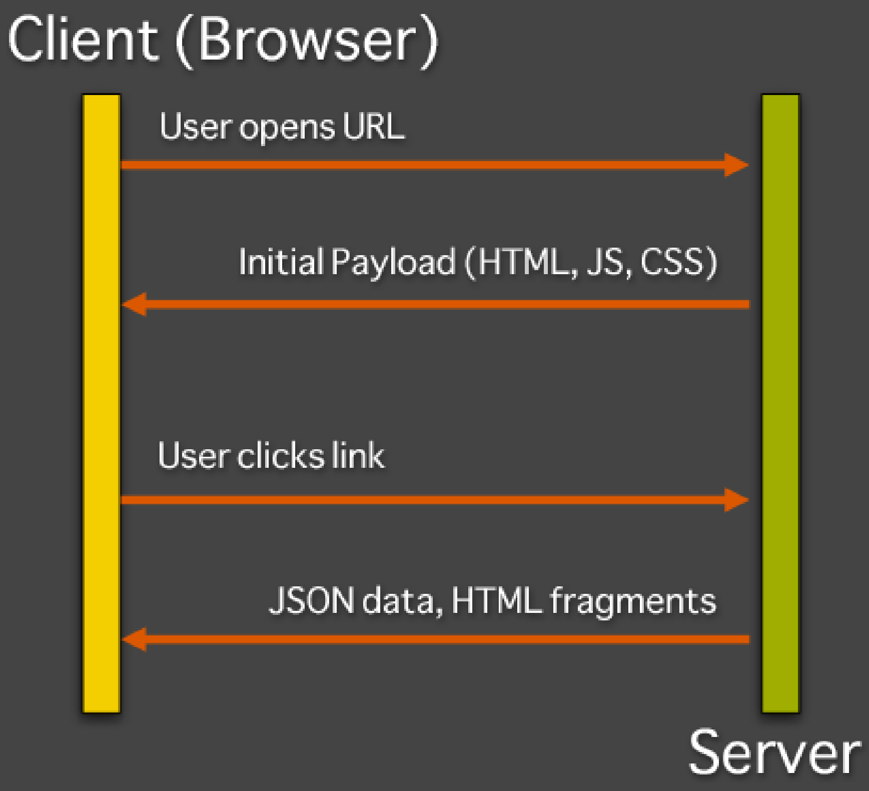
- AngularJS? 50%
- Backbone? 20%
- JQuery? 90%
- Are you using any other SPA Framework? ExtJS
- Spring MVC? 60%
- Spring Boot? 10%

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 1/2

http://wtfjs.com/

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

NaN

12

JAVASCRIPT WTF 2/2

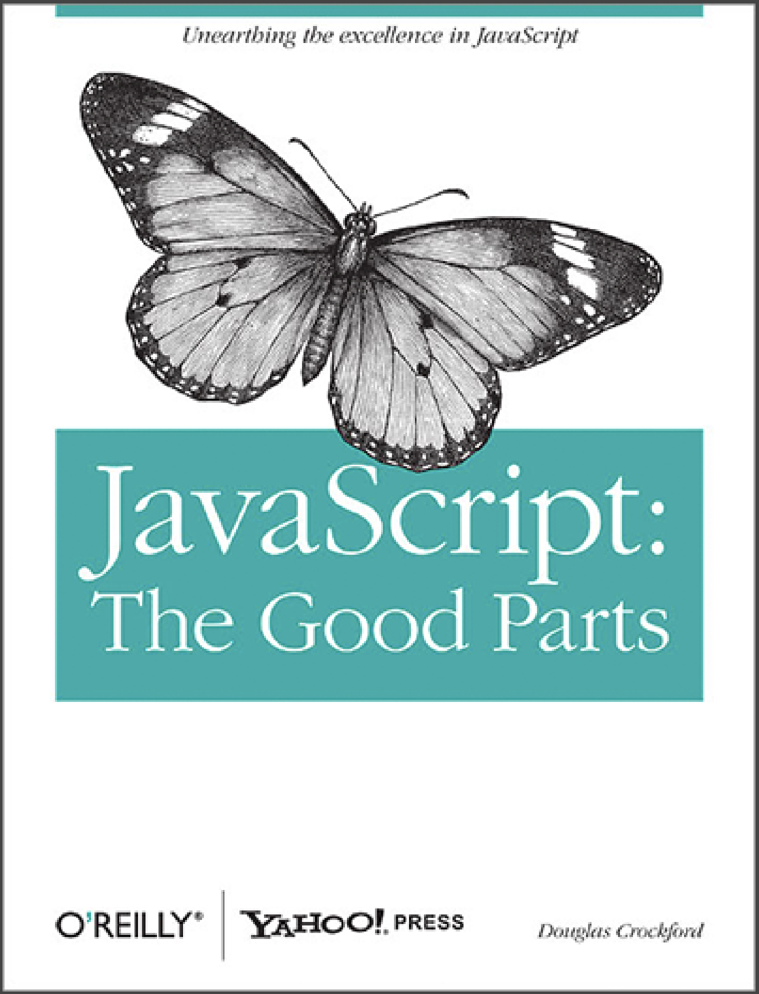
http://wtfjs.com/

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

23

5

0



Unearthing the excellence in JavaScript

JavaScript: The Good Parts

O'REILLY | YAHOO! PRESS

Douglas Crockford

FROM BACKBONE TO ANGULAR

Too many moving parts, choices

Boilerplate Code

Marionette, Backbone.ModelBinder, Backbone.Relational

ALTERNATIVES

ANGULAR JS BASICS

Model

View (Templates)

Controller

Expressions

Directives

Modules

See also: AngularJS Concepts

¡HOLA!

```
<div ng-app ng-init="firstName='Angular';lastName='rocks'">
  <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>
```

Demo

MODEL 1/2

Angular is very flexible about your model

Ultimately expressed via the $scope

$scope = Glue between Controller and View

$scope mimics DOM (Hierarchical, one $rootScope)

$watch, $apply

MODEL 2/2

Killer Feature: Data-Binding

Model === single-source-of-truth

View reflects model changes automatically

VIEW

HTML is your templating Engine

Minimize logic as much as possible

Consider Custom Directives

CONTROLLER

Used to "setup" your $scope

Add behavior to your $scope

Don't do UI work using controllers!!

Use directives and filters instead

¡HOLA! V2.0 - VIEW

```
<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>
```

Demo

¡HOLA! V2.0 - CONTROLLER

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

Demo

DEPENDENCY INJECTION

Consider using array notation

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

Or use ngmin

grunt-ngmin, gulp-ngmin

EXPRESSIONS

{{ expression }}

No Control Flow Statements

Can use filters inside expressions:

{{ 'abcd' | uppercase }}

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

MODULES

Bear with me ...

BUILD AND DEPLOYMENT

STRATEGIES - JAVA TOOLING

- Wro4j
- Jawr
- Spring 4.1 (SPR-10310, SPR-10933)
- See Blog Post
- WebJars

STRATEGIES - JAVASCRIPT TOOLING

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

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() { ... } ...
}
```

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

https://docs.angularjs.org/guide/module

Compartmentalize sections of your application

Does not deal with script loading

```
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

MODULARIZATION FUTURE

ECMAScript 6 modules

AngularJS 2 is being written in ES6

Web Components

MORE COOLNESS

FILTERS

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

FILE UPLOAD

angular-file-upload (nervgh)

angular-file-upload (danialfarid)

File Reader

Traditional Post

ROUTING

ngRoute (built-in)

Routing on steroids using ui-router

ROUTING USING UI-ROUTER

state machine

nested views

Spring XD's **routes.js**

TESTING

E2E testing with Protractor

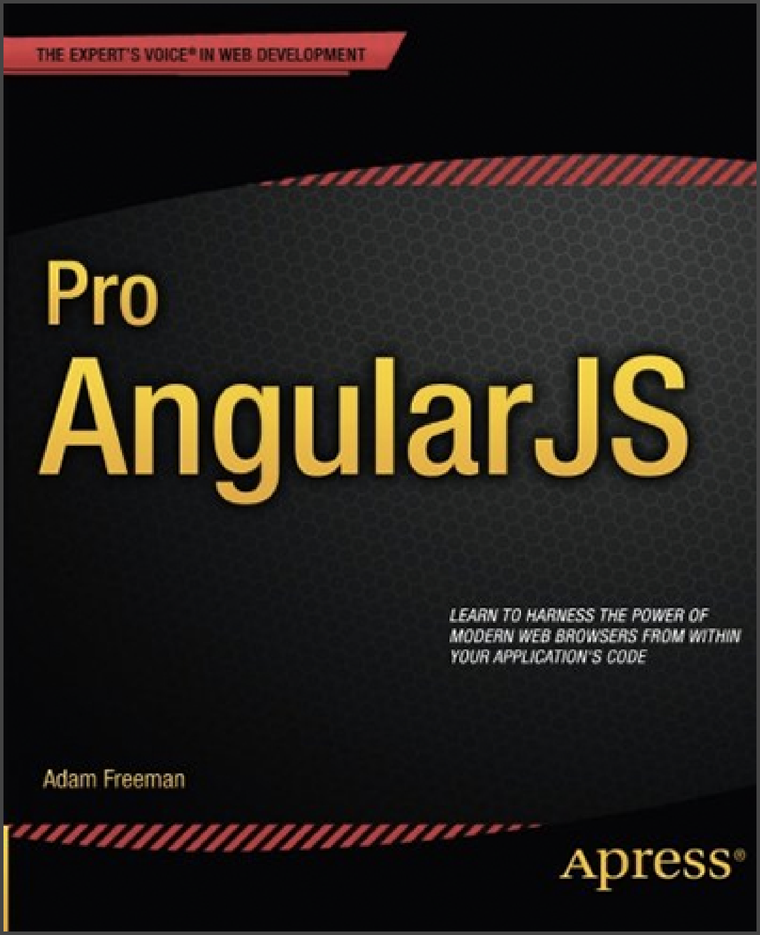
Unit Testing using Karma and Jasmine

UI CONSIDERATIONS

Bootstrap

Keep your CSS maintainable with Less and Sass

RESOURCES



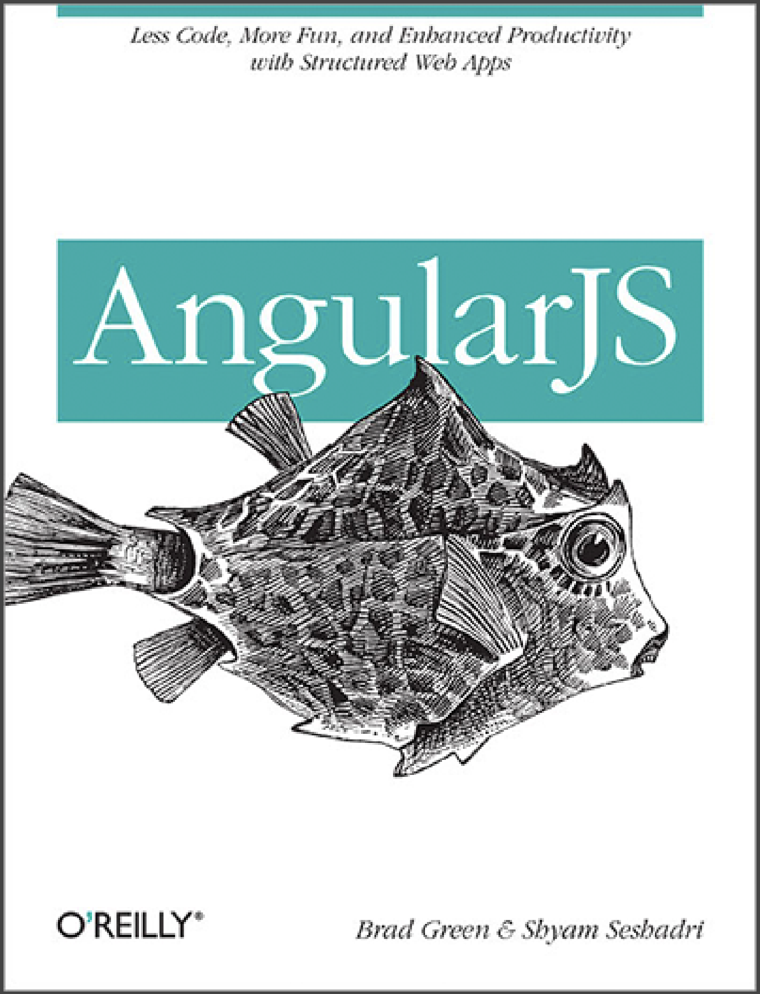
THE EXPERT'S VOICE® IN WEB DEVELOPMENT

Pro AngularJS

LEARN TO HARNESS THE POWER OF MODERN WEB BROWSERS FROM WITHIN YOUR APPLICATION'S CODE

Adam Freeman

Apress®



Less Code, More Fun, and Enhanced Productivity with Structured Web Apps

AngularJS

O'REILLY®

Brad Green & Shyam Seshadri

THANK YOU!!

Source Code + Preso:

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