

# From Java To AngularJS (Without Pain)



# Background



# JavaScript!!!





# Java <3



# JavaScript





# JavaScript is evil





# JavaScript is eval



# JavaScript is evil





# JS: The Home Of Antipatterns



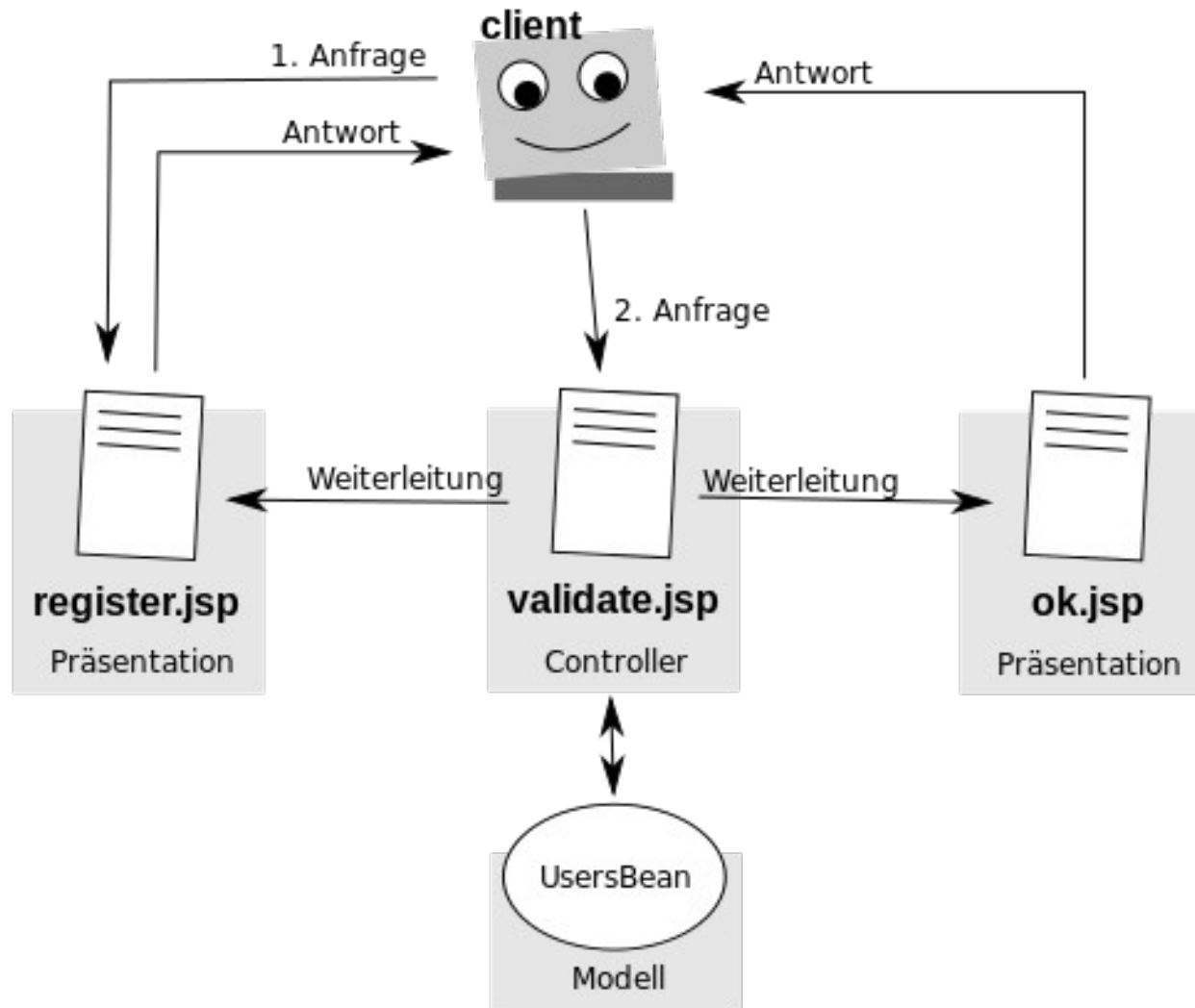


# Some master it...





# MVC in Java

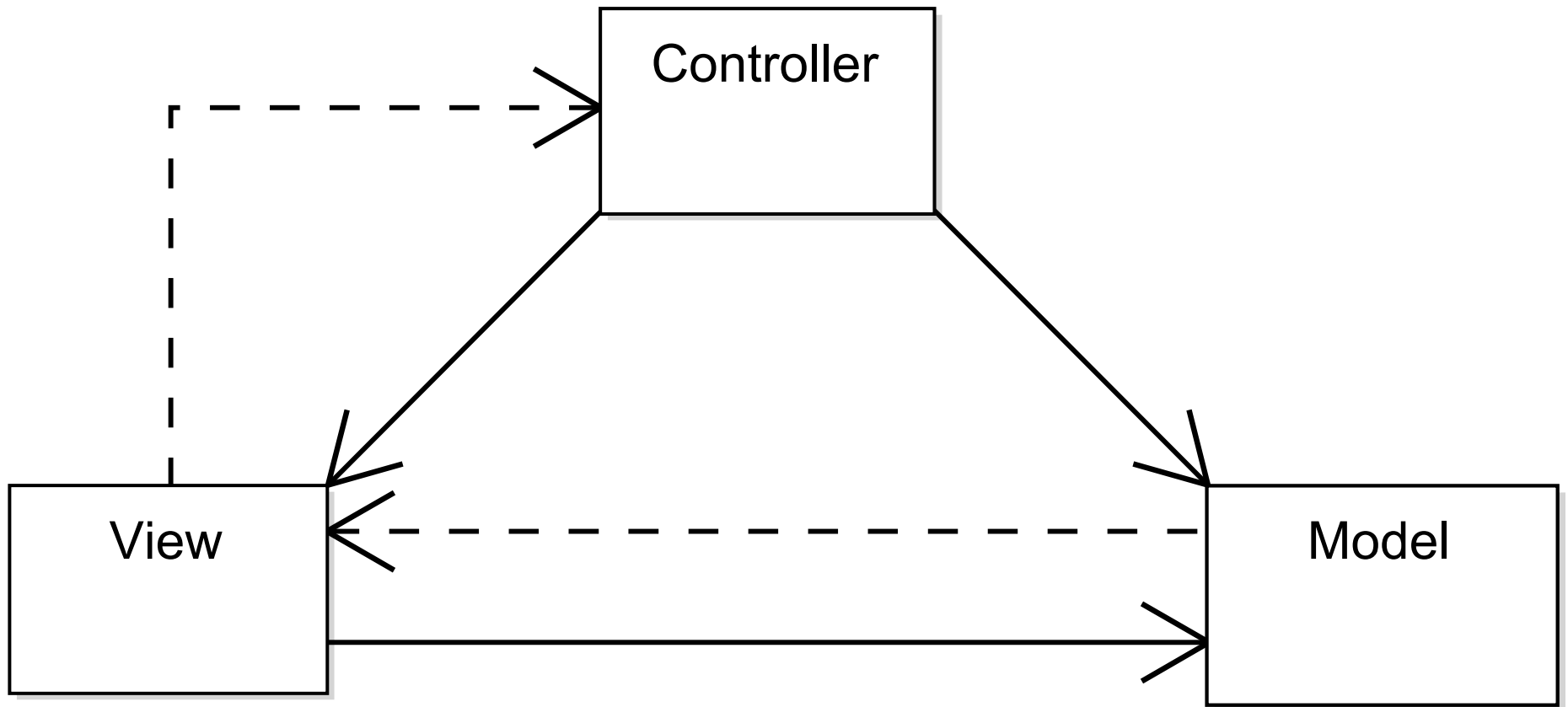


# AngularJS FTW!





# MVW in Angular

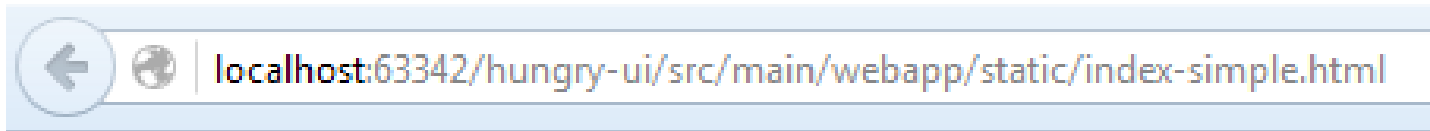


# ng-what?

```
index-simple.html x
1  <!doctype html>
2  <html ng-app>
3  <head>
4      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5  </head>
6  <body>
7  <div>
8      <label>Name:</label>
9      <input type="text" ng-model="yourName" placeholder="Enter a name here">
10     <hr>
11     <h1>Hello {{yourName}}!</h1>
12 </div>
13 </body>
14 </html>
```



# Live Edit in Browser

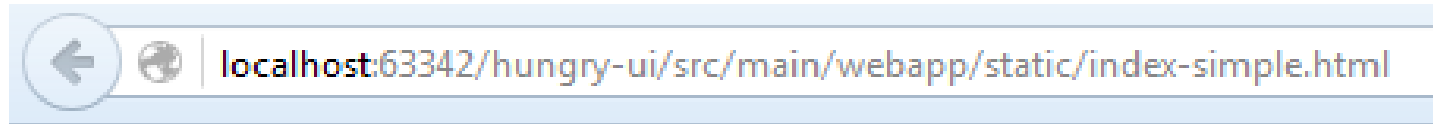


Name:

---

**Hello !**

# Live Edit in Browser



Name:

---

**Hello World!**

# Add a Controller to View

```
index-controller.html x  venues-controller.js x
1  <!doctype html>
2  <html ng-app="hungryUIApp">
3  <head>
4      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5      <script src="js/venues-controller.js"></script>
6  </head>
7  <body>
8      <h1>Hungry? UI</h1>
9
10     <h2>Venue Lists</h2>
11
12     <div ng-controller="VenueListController as ctrl">
13         <span>Venue Lists {{ctrl.venueLists.length}}</span>
14         <ul>
15             <li ng-repeat="venueList in ctrl.venueLists">
16                 <span>{{venueList.name}}</span>
17             </li>
18         </ul>
19
20         <h>New Venue List</h>
21         <form ng-submit="ctrl.addNewVenueList()">
22             <input type="text" ng-model="ctrl.newVenueList.name" size="30" placeholder="add new venue list name here">
23             <input class="btn-primary" type="submit" value="add">
24         </form>
25     </div>
26 </body>
27 </html>
```



# The Controller

```
index-controller.html x  venues-controller.js x
1  angular.module('hungryUIApp', [])
2  .controller('VenueListController', function () {
3      var ctrl = this;
4      ctrl.venueLists = [
5          {id: 1, name: 'Cool Berlin List'},
6          {id: 2, name: 'Vienna Hipster Places List'}];
7
8      ctrl.addNewVenueList = function () {
9          console.log("ctrl.venueLists", ctrl.venueLists);
10         ctrl.venueLists.push(ctrl.newVenueList);
11         ctrl.newVenueList = {};
12     };
13 };
```

# Live Edit in Browser



localhost:63342/hungry-ui/src/main/webapp/static/index-controller.html

## Hungry? UI

### Venue Lists

#### Venue Lists 2

- Cool Berlin List
- Vienna Hipster Places List

#### New Venue List

# Live Edit in Browser

← localhost:63342/hungry-ui/src/main/webapp/static/index-controller.html

## Hungry? UI

### Venue Lists

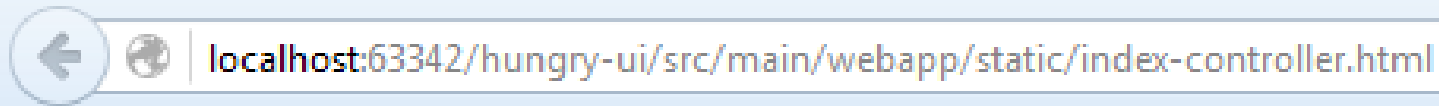
#### Venue Lists 2

- Cool Berlin List
- Vienna Hipster Places List

#### New Venue List



# Live Edit in Browser



## Hungry? UI

### Venue Lists

#### Venue Lists 3

- Cool Berlin List
- Vienna Hipster Places List
- Super Duper London List

#### New Venue List

# Use A Model

```
index-resource.html x  js  venues-resource.js x
1  <!doctype html>
2  <html ng-app="hungryUIApp">
3  <head>
4      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular-resource.min.js"></script>
6      <script src="js/venues-resource.js"></script>
7  </head>
8  <body>
9      <h1>Hungry? UI</h1>
10
11     <h2>Venue Lists</h2>
12
13     <div ng-controller="VenueListController as ctrl">
14         <span>Venue Lists: {{ctrl.venueLists.length}}</span>
15         <ul>
16             <li ng-repeat="venueList in ctrl.venueLists">
17                 <span>{{venueList.name}}</span>
18             </li>
19         </ul>
20
21         <h>New Venue List</h>
22         <form ng-submit="ctrl.addNewVenueList()">
23             <input type="text" ng-model="ctrl.newVenueList.name" size="30" placeholder="add new venue list name here">
24             <input class="btn-primary" type="submit" value="add">
25         </form>
26     </div>
27 </body>
28 </html>
```

# Different Modules in Controller

```
angular.module('hungryUIApp', ['hungry.controllers']);

// controllers
function VenueListController(VenueList) {
  this.venueLists = VenueList.query();

  this.newVenueList = new VenueList();

  this.addNewVenueList = function () {
    this.newVenueList.$save(function (ret) {
      this.venueLists.push(ret);
      this.newVenueList = new VenueList();
    }).bind(this), function (err) {
      console.log("---err ", err);
    });
  };

  this.deleteVenueList = function (venueList) {
    venueList.$delete().then(function () {
      this.venueLists.remove(venueList);
    }).bind(this);
  }
}

function UserController() {
  // TODO
}

angular.module('hungry.controllers', ['hungry.constants', 'hungry.factories']);

angular.module('hungry.controllers').controller('VenueListController', VenueListController);
angular.module('hungry.controllers').controller('UserController', UserController);

// constants
angular.module('hungry.constants', [])
  .constant('API_PATH', 'http://localhost:8080/hungry-rest');

// factories
var factoryModule = angular.module('hungry.factories', ['ngResource', 'hungry.constants']);

factoryModule.factory('VenueList', VenueListFactory);

function VenueListFactory($resource, API_PATH) {
  return $resource(API_PATH + '/api/venueLists/:id',
    {
      id: '@id'
    });
}
```



# Dependency Injection

```
index-resource.html x venues-resource.js
angular.module('hungryUIApp', ['hungry.controllers']);

// controllers
function VenueListController(VenueList) {
  this.venueLists = VenueList.query();

  this.newVenueList = new VenueList();

  this.addNewVenueList = function () {
    this.newVenueList.$save(function (ret) {
      this.venueLists.push(ret);
      this.newVenueList = new VenueList();
    }).bind(this), function (err) {
      console.log("---err ", err);
    });
  };

  this.deleteVenueList = function (venueList) {
    venueList.$delete().then(function () {
      this.venueLists.remove(venueList);
    }).bind(this);
  }
}

function UserController() {
  // TODO
}

angular.module('hungry.controllers', ['hungry.constants', 'hungry.factories']);

angular.module('hungry.controllers').controller('VenueListController', VenueListController);
angular.module('hungry.controllers').controller('UserController', UserController);

// constants
angular.module('hungry.constants', [])
  .constant('API_PATH', 'http://localhost:8080/hungry-rest');

// factories
var factoryModule = angular.module('hungry.factories', ['ngResource', 'hungry.constants']);

factoryModule.factory('VenueList', VenueListFactory);

function VenueListFactory($resource, API_PATH) {
  return $resource(API_PATH + '/api/venueLists/:id',
    {
      id: '@id'
    });
}
```

# Directives

```
JS VenueListEditDirective.js x JS modules.js x
1 // directives
2 angular.module('hungry.directives', [])
3 .directive('showVenueListDetails', function() {
4     return {
5         template: 'Id: {{venueList.id}}, Name: {{venueList.name}}'
6     };
7 });
8
```

# Deep Linking & Routing

```
routes.js x
function Routes($routeProvider) {
  $routeProvider.when('/', {
    templateUrl: 'main.html'
  });

  $routeProvider.when('/venueLists', {
    templateUrl: 'venueList/venueLists.html',
    controller: 'VenueListController as ctrl'
  });

  $routeProvider.when('/users', {
    templateUrl: 'user/userList.html',
    controller: 'UserController as ctrl'
  });
}

angular.module('hungryUIApp').config(['$routeProvider', Routes]);
```

# Nested Views

```
index.jade x  main.jade x
1  .row
2    .col-md-12
3      h1 Hungry? UI
4
5      a(href="#/venueLists")
6        h2 Venue Lists
7
8      a(href="#/users")
9        h2 Users
10
```



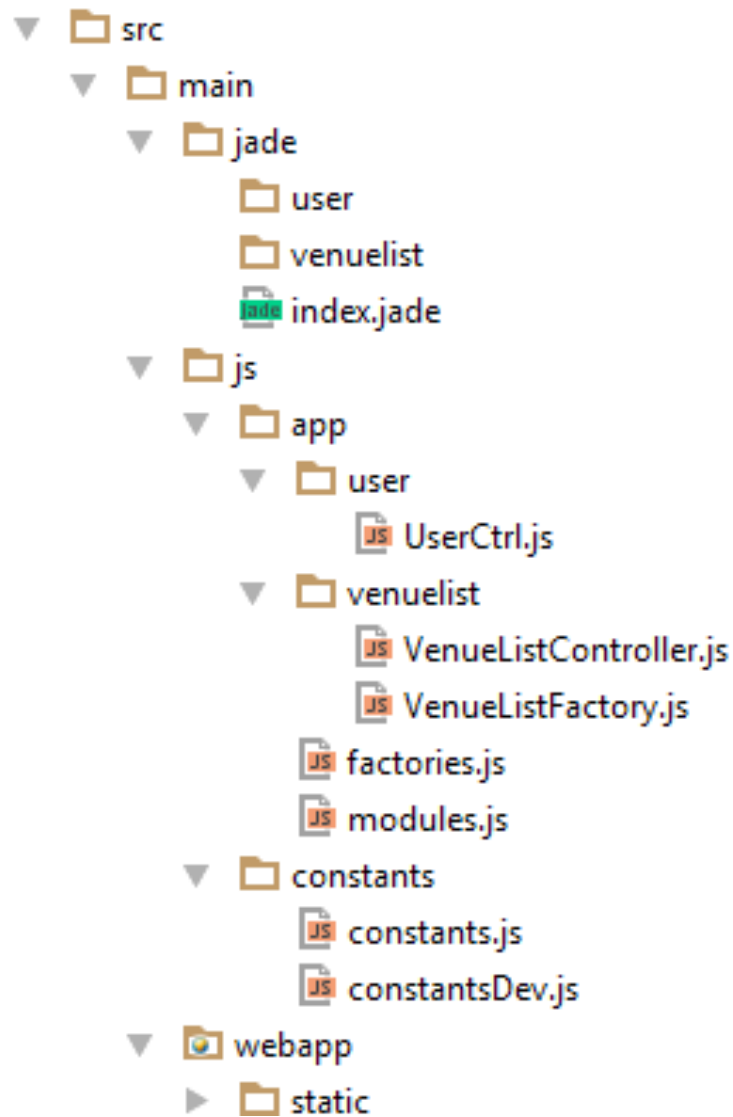
# Nested Views

```
index.jade x  main.jade x  header.jade x
1  doctype html
2  html
3    head
4      base(href='.')
5      meta(charset='utf-8')
6      meta(http-equiv='Cache-Control' content='no-cache, no-store, must-revalidate')
7      meta(http-equiv="Pragma" content="no-cache")
8      meta(http-equiv="Expires" content="0")
9      meta(http-equiv="Vary" content="*")
10     title Hungry? UI
11     link(rel='shortcut icon' href='static/ico/favicon.ico')
12     script(src='static/js/app/libs.js')
13     script(src='static/js/app/main.js')
14     link(href='static/bower_components/jquery-ui-bootstrap/jquery.ui.theme.css' rel='stylesheet')
15
16   body (ng-app="hungryUIApp")
17     div.container#maincontainer
18       block content
```

# Nested Views

```
index.jade x
1 extends header
2
3 -block content
4   -div(ng-view)
5
6
```

# Structure



# Testability

```
Terminal
+
x
E:\Dokumente\java\intellij\george\george-backoffice\backoffice-ui>gulp
[02:36:02] Using gulpfile E:\Dokumente\java\intellij\george\george-backoffice\backoffice-ui\gulpfile.js
[02:36:02] Starting 'webpack'...
[02:36:02] Starting 'jadeStatic'...
[02:36:02] Finished 'jadeStatic' after 11 ms
[02:36:08] [webpack] Hash: 31b3b83c45b61d5c51e6
Version: webpack 1.12.4
Time: 6038ms

      Asset      Size  Chunks             Chunk Names
96f337bc7a3d5d8beebe128ff6630b62.svg  115 kB          0  [emitted]
84af6538466d3d7af69fe5d417254906.png   5.6 kB          0  [emitted]
f4769f9bdb7466be65088239c12046d1.eot  20.1 kB          0  [emitted]
e18bbf611f2a2e43afc071aa2f4e1512.ttf  45.4 kB          0  [emitted]
89889688147bd7575d6327160d64e760.svg   109 kB          0  [emitted]
7b53eaf5cf6b574674895416f5edda9c.eot   24.6 kB          0  [emitted]
ef057bca0253651284caddf19a9a2bd2.png    15 kB          0  [emitted]
57c1ddb3d0e033a133a670525037a224.png  25.7 kB          0  [emitted]
      bundle.js    255 kB          0  [emitted] bundle
      libs.js      3.44 MB          1  [emitted] libs
      styles.js    53.3 kB          2  [emitted] styles
      templates.js  178 kB          3  [emitted] templates

[02:36:08] Finished 'webpack' after 6.1 s
[02:36:08] Starting 'karma'...
09 12 2015 02:36:12.011:INFO [karma]: Karma v0.13.15 server started at http://localhost:9876/
09 12 2015 02:36:12.023:INFO [launcher]: Starting browser PhantomJS
09 12 2015 02:36:12.623:INFO [PhantomJS 1.9.8 (Windows 7 0.0.0)]: Connected on socket 9S0wlcPqGFmZ7Tz4AAAA with id 72342548
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 101 of 504 SUCCESS (0 secs / 0.648 secs)
09 12 2015 02:36:14.411:WARN [web-server]: 404: /export/logos/logo_12_100.png?1449624974409409
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 115 of 504 SUCCESS (0 secs / 0.805 secs)
09 12 2015 02:36:14.492:WARN [web-server]: 404: /export/logos/logo_12_100.png?1449624974436436
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 188 of 504 SUCCESS (0 secs / 1.276 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 205 of 504 SUCCESS (0 secs / 1.423 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 232 of 504 SUCCESS (0 secs / 1.643 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.186 secs: NGO Factories Spec NGO Spec should refine resource NGO
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.145 secs: Plugin Component List Directive Spec Adding Spec should show edit area when adding
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.229 secs: PluginFiles directive Spec should upload image for plugin
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 504 of 504 SUCCESS (5.383 secs / 5.275 secs)
[02:36:19] Finished 'karma' after 11 s
[02:36:19] Starting 'default'...
[02:36:19] Finished 'default' after 5.66 ms

E:\Dokumente\java\intellij\george\george-backoffice\backoffice-ui>
```



# Build

```
gulpfile.js x
27  gulp.src(libs)
28      .pipe(concat('libs.js'))
29      .pipe(gulp.dest(DEST + '/static/js/app/'));
30
31  gulp.src([confFile].concat(sources))
32      .pipe(concat('main.js'))
33      .pipe(gulp.dest(DEST + '/static/js/app/'));
34  }
35
36  gulp.task('concatDev', function () {
37      doConcat("src/main/js/constants/constantsDev.js");
38  });
39
40  gulp.task('jade', function (done) {
41      gulp.src(jadeSrc)
42          .pipe(jade({pretty: true}))
43          .on('error', function (err) {
44              console.log("Jade error: ", err.message);
45          })
46          .pipe(gulp.dest(DEST))
47          .on("end", function () {
48              done();
49          });
50  });
51
52
53  function doWatch(concatTask) {
54      gulp.watch([sources], [concatTask]);
55      gulp.watch(['src/main/jade/**/*.jade'], ['jade']);
56  }
57
58  gulp.task('watch', function () {...});
59
60
61  gulp.task('default', ['concatDev', 'jade']);
62
63
64
65  // dist
66  gulp.task('uglify', function () {
67      gulp.src([DEST + '/static/js/app/libs.js'].
68          concat(["src/main/js/constants/constants.js", DEST + '/static/js/app/main.js']))
69          .pipe(uglify())
70          .pipe(concat('main.min.js'))
71          .pipe(gulp.dest(DEST + '/static/js/app/'));
72  });
73
74  gulp.task('concatDist', function () {
75      doConcat("src/main/js/constants/constants.js");
76  });
77
78  gulp.task('dist', ['concatDist', 'jade']);
79
```

# Dependency Management

```
bower.json x
1 {
2   "name": "HungryUIApp",
3   "version": "1.21.0",
4   "authors": [
5     "Barbara Ondrisek <office@electrobabe.at>"
6   ],
7   "license": "ISC",
8   "repository": {
9     "type": "git",
10    "url": "git://github.com/electrobabe/hungry-ui.git"
11  },
12  "ignore": [
13    "**/*.*",
14    "node_modules",
15    "bower_components"
16  ],
17  "dependencies": {
18    "angular": "^1.4.7",
19    "angular-ui-bootstrap": "^0.14.2",
20    "jquery-ui-bootstrap": "~0.2.5",
21    "angular-resource": "^1.4.7",
22    "bootstrap": "^3.3.5",
23    "lodash": "^3.10.1"
24  }
25 }
26
```

# Supporting Frameworks

```
devDependencies
23   "angular-ui-bootstrap": "0.11.2",
24   "babel-core": "^6.1.2",
25   "babel-loader": "^6.0.1",
26   "babel-preset-es2015": "^6.1.2",
27   "bootstrap": "^3.3.5",
28   "bootstrap-markdown": "^2.9.0",
29   "bower": "~1.3.8",
30   "chai": "^3.3.0",
31   "css-loader": "^0.22.0",
32   "eslint": "^1.7.3",
33   "eslint-loader": "^1.1.0",
34   "event-stream": "^3.3.0",
35   "file-loader": "^0.8.4",
36   "gridster": "^0.5.6",
37   "gulp": "^3.8.11",
38   "gulp-jade": "^1.1.0",
39   "gulp-replace": "^0.5.0",
40   "gulp-util": "^3.0.7",
41   "jade": "^1.11.0",
42   "jade-loader": "^0.8.0",
43   "jasmine-core": "^2.1.2",
44   "jquery": "^2.1.4",
45   "jshint": "^2.8.0",
46   "jshint-stylish": "^1.0.0",
47   "json-loader": "^0.5.4",
48   "karma": "^0.13.15",
49   "karma-beep-reporter": "^0.1.4",
50   "karma-chai": "^0.1.0",
51   "karma-coffee-preprocessor": "^0.3.0",
52   "karma-coverage": "0.4.2",
53   "karma-jasmine": "^0.3.0",
54   "karma-mocha-reporter": "^1.1.1",
55   "karma-phantomjs-launcher": "^0.2.1",
56   "karma-sinon": "^1.0.4",
57   "less": "^2.5.3",
58   "less-loader": "^2.2.1",
59   "lodash": "^3.10.1",
60   "markdown": "^0.5.0",
61   "moment": "^2.8.4",
62   "phantomjs": "^1.9.18",
63   "should": "~3.1.3",
64   "sinon": "^1.17.1",
65   "style-loader": "^0.13.0",
66   "ui-select": "^0.13.2",
67   "url-loader": "^0.5.6",
68   "webpack": "^1.12.2",
69   "webpack-dev-server": "^1.12.1",
70   "webpack-manifest-plugin": "^0.5.0",
71   "webpack-md5-hash": "0.0.4",
72   "yaml-loader": "^0.1.0"
```

# Best practices & Lessons Learned





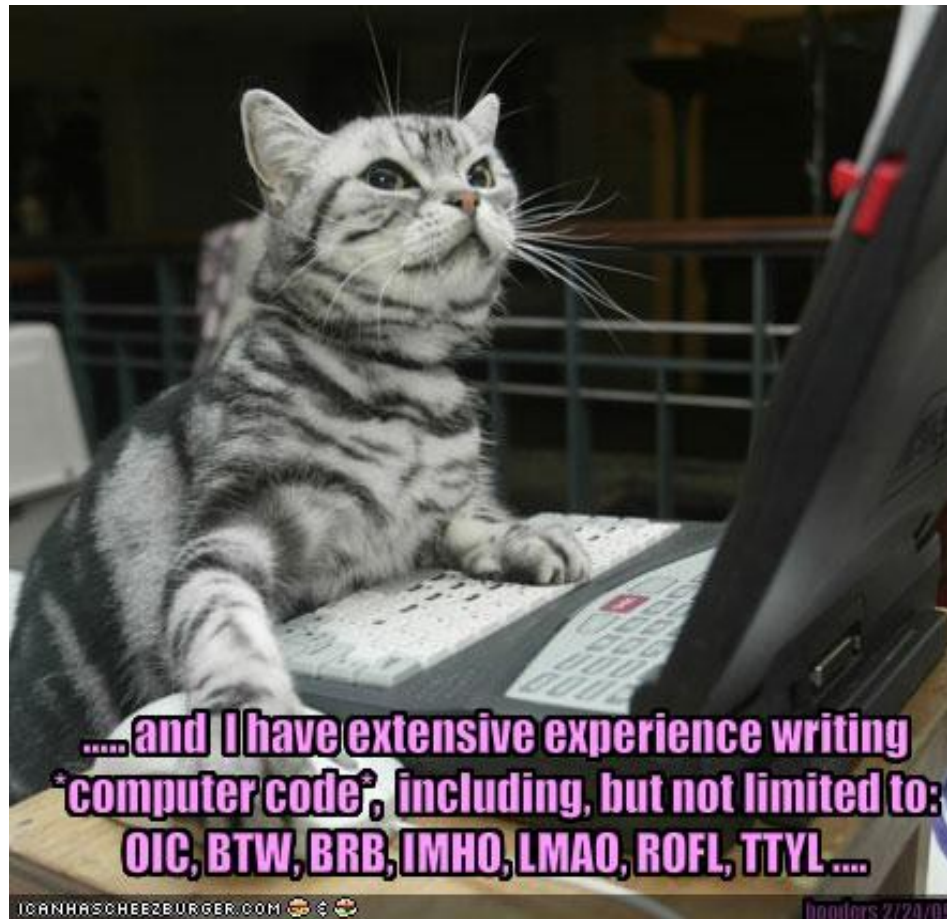
# Downsides

# Downsides

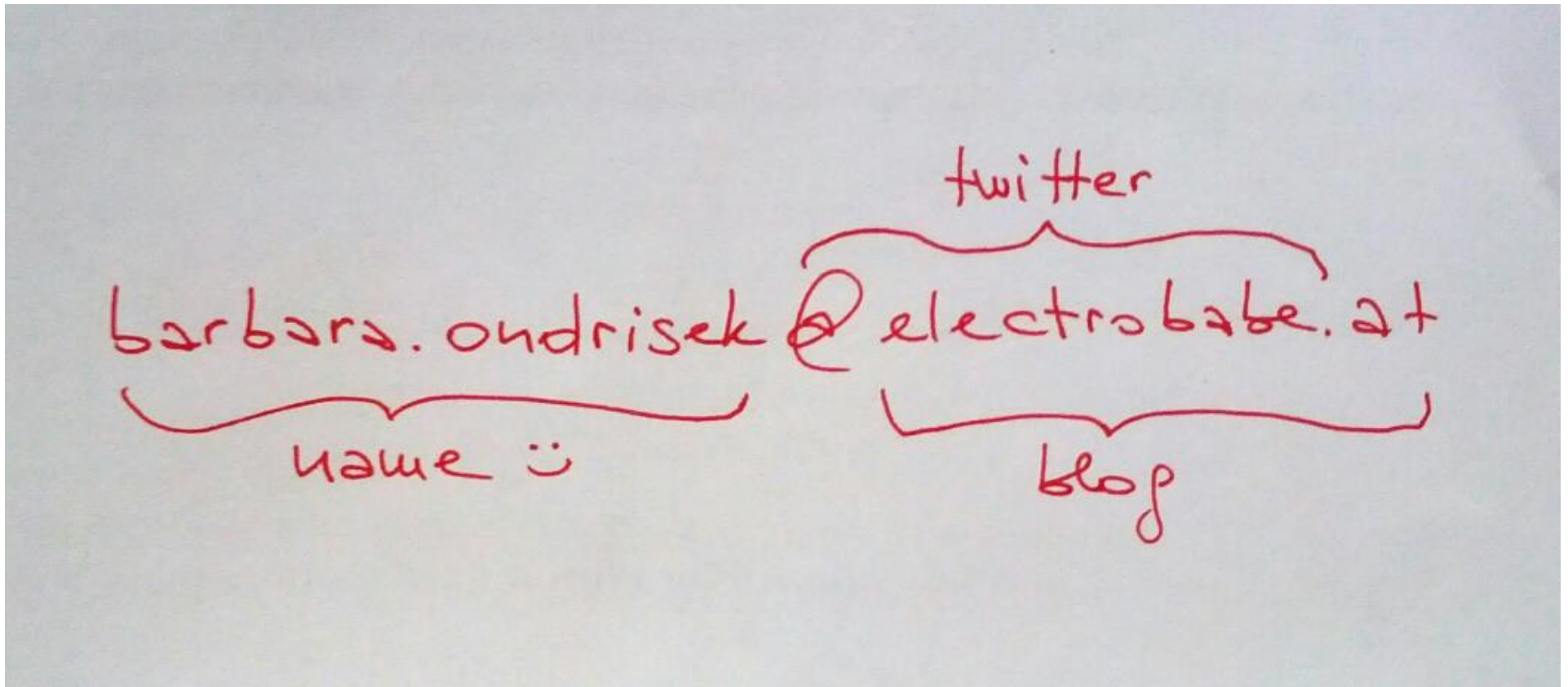


# I can has code? kthxbye

<https://github.com/electrobabe/hungry-ui>



# Any Questions?





# CATS!



# Links

- Slides and Src on Github  
<https://github.com/electrobabe/hungry-ui>
- <http://electrobabe.at>
- <https://play.google.com/store/apps/details?id=at.chefbabe.hungry>
- <https://play.google.com/store/apps/details?id=com.likeahipster.app>

# Credentials

Images from

- [hipsterlogogenerator.com](http://hipsterlogogenerator.com)
- [stocksnap.io](http://stocksnap.io)
- Wikipedia
- Giphy

# From Java To AngularJS (Without Pain)



@electrobabe

Hello!

My name is Barbara Ondrisek and I'm working as a freelancer for almost 15 years for different companies mostly on Java web projects, but I also like mobile development – especially Android.

Currently I'm working for the Erste Bank on the new Version of their netbanking system George.

## Background



9.12.2015 jquery conf berlin

@electrobabe



I have a strong Java background – I love Java - but I always wanted to keep in touch with the Frontend. So last year my team leader asked me wheater I want to join a Full Stack team.

BUT!

They use JavaScript



# JavaScript!!!



9.12.2015 jquery conf berlin

@electrobabe



BUT Java developers fear Java Script.

Why?

ps. first I wanted to make a presentation solely with cat pictures but I decided against it

## Java <3



9.12.2015 jquery conf berlin

@electrobabe



For me object oriented Java is so comforting because it uses a lot of structure – law and order, especially for the "Special Victims Unit" the Java Server Faces guys.

There is a huge barrier when you start programming Java, especially when you come from "script land". But there are many very handy frameworks and standard tools such as Maven, JPA, or Spring that might help you with developing reliable large scale web apps.

And for us Java people JS looks like...

# JavaScript



9.12.2015 jquery conf berlin

@electrobabe



Sodom and Gomorrha, because there are sooo many frameworks and tools and noone standardizes anything. Of cause there are certain streams to use structural frameworks such as Backbone, React, ... and now Ecma Script 6 is coming...

But for me as a Java developer JS was always a place where every thing is allowed and possible and you can do whatever comes to your mind...

# JavaScript is evil



9.12.2015 jquery conf berlin

@electrobabe



I mean seriously

# JavaScript is eval



9.12.2015 jquery conf berlin

@electrobabe



...eval is the same as ...



# JavaScript is evil



9.12.2015 jquery conf berlin

@electrobabe



...evil!

## JS: The Home Of Antipatterns



9.12.2015 jquery conf berlin

@electrobabe



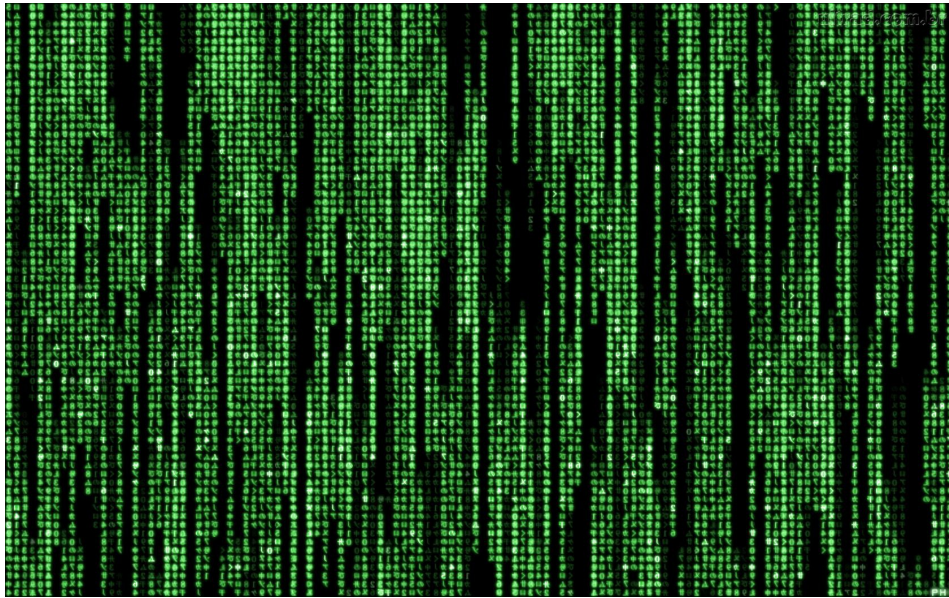
What ever prgramm any antipattern that comes into your mind such as spaghetti code (this is actually a picture of my lunch last week), extension of objects, pollution of the global Namespace, improper Use of Truthy and Falsey Evaluation etc.

JS has no standardisation, millions of different, frameworks and github projects...

All those strange "undefined is not a function"s and "object Object"s (when you print out a variable and HTML code mixed with javascript...

No IDE, no debugging...

# Some master it...



9.12.2015 jquery conf berlin

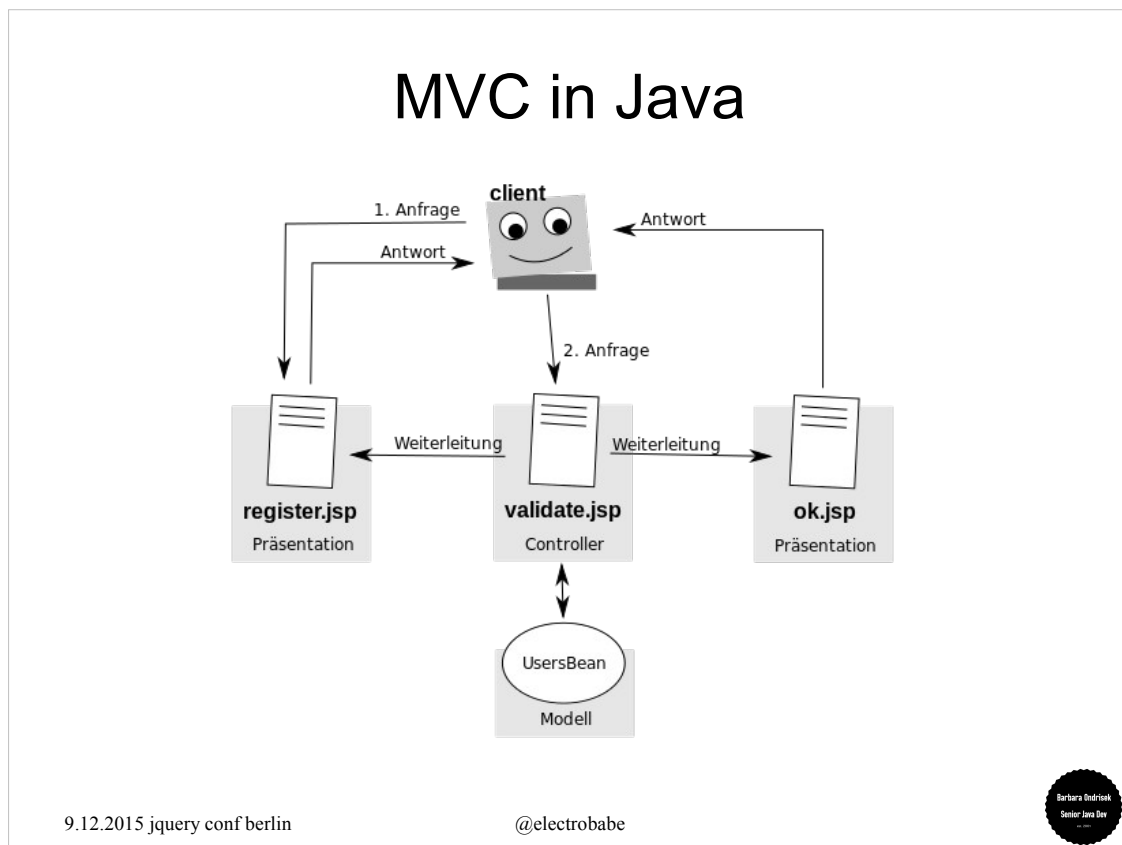
@electrobabe



Some might see through the matrix, but not many can handle JS code in a proper way that it is maintainable or scalable.

BUT web app implementations in Java aren't much better:

ps. now I'm coming to the a little more serious part of my presentation.



Your code base usually grows very fast and you need structure.

Model-View-Controller is a pattern meaning the separation of concerns. You separate the presentation (the view) from the program's logic (the controller) and the data layer (the model).

There are several ways to achieve MVC in Java. A good approach is Spring MVC with Spring Web Flow - but there are also bad and very bad approaches such as JSP or JSF with Beans or Servlets. Old technology I contributed some bad still on my own.

So the solution is:

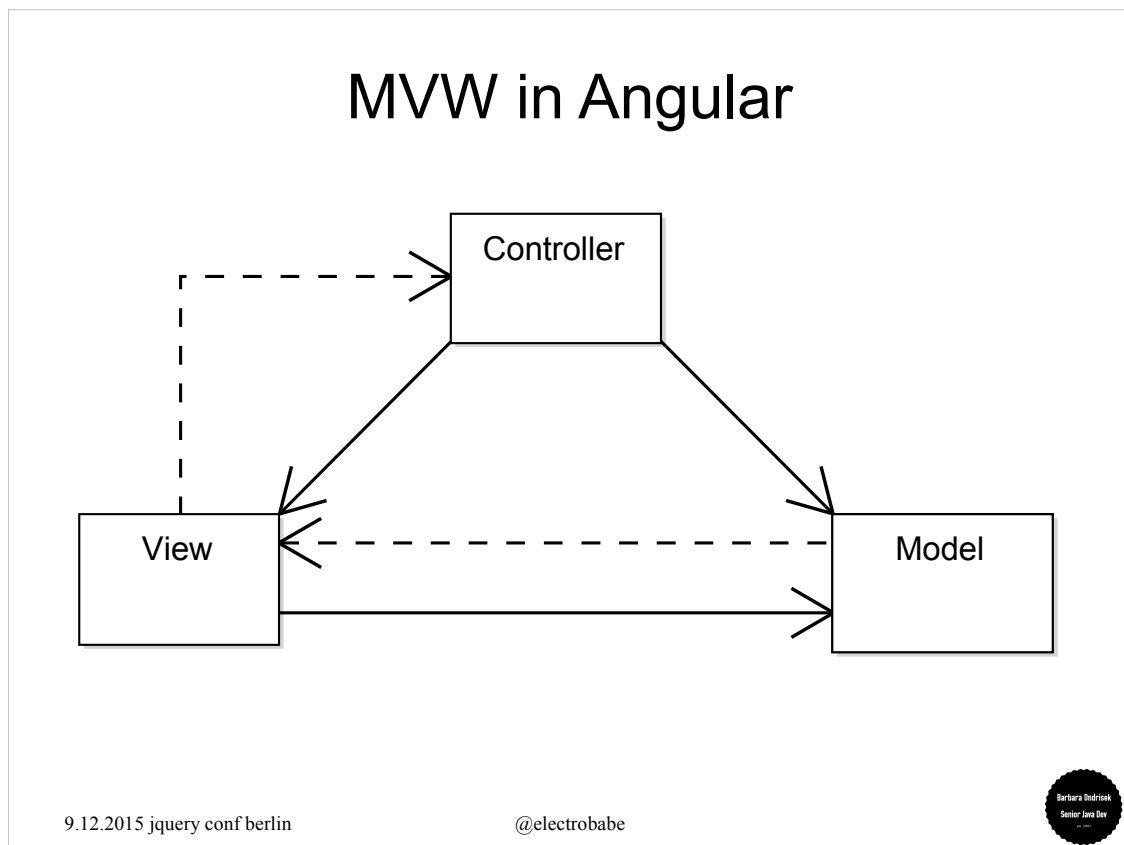


## AngularJS FTW!

The self-called "Superheroic JavaScript MVW Framework" AngularJS is a state of the art JavaScript tool for single page web applications that offers a lot that JS alone does not - namely structure. And it's from Google so it can't be "bad".

In jQuery apps the DOM often represents the model, but with Angular the business logic is decoupled from DOM manipulation.

And just a litte disclaimer: I'm talking about Angular 1, not 2, because AFAIK it's still BETA.



Angular calls their principle "Model-View-Whatever":  
You have a single page app divided into modules  
such as the login, a main screen, config).  
Simply speaking: An Angular app is a collection of  
modules.

A module consists of different components:

- \* Model (Services / Factories <-> REST API)
- \* <-> Controller (Controllers)
- \* <-> View (Templates with build in Directives)



# ng-what?

```
index-simple.html x
1  <!doctype html>
2  <html ng-app>
3  <head>
4    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5  </head>
6  <body>
7    <div>
8      <label>Name:</label>
9      <input type="text" ng-model="yourName" placeholder="Enter a name here">
10     <hr>
11     <h1>Hello {{yourName}}!</h1>
12   </div>
13 </body>
14 </html>
```

9.12.2015 jquery conf berlin

@electrobabe



Here you find a simple html file. I include the angular dependency as js script in the header and define that this references an angular app with "ng-app".

With a very simple reference to "ng-model" I define an object (through data binding) in the current scope respectively DOM that will be initialized on runtime.

Next I just display it by referencing it inside this curly braces, which means that it will be "printed out" in the HTML.



When you now open the html file in a browser the browser loads this mini app and when you now enter something in the input field the value is displayed here in this span immediately.



Easy, isn't it?

## Add a Controller to View



```

1 <!doctype html>
2 <html ng-app="hungryUIApp">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5   <script src="js/venues-controller.js"></script>
6 </head>
7 <body>
8   <h1>Hungry? UI</h1>
9
10  <h2>Venue Lists</h2>
11
12  <div ng-controller="VenueListController as ctrl">
13    <span>Venue Lists {{ctrl.venueLists.length}}</span>
14    <ul>
15      <li ng-repeat="venueList in ctrl.venueLists">
16        <span>{{venueList.name}}</span>
17      </li>
18    </ul>
19
20    <h3>New Venue List</h3>
21    <form ng-submit="ctrl.addNewVenueList()">
22      <input type="text" ng-model="ctrl.newVenueList.name" size="30" placeholder="add new venue list name here">
23      <input class="btn-primary" type="submit" value="add">
24    </form>
25  </div>
26 </body>
27 </html>

```

9.12.2015 jquery conf berlin

@electrobabe



Now you add some swag: A Controller.

This is an example app I wrote for my talk here, but it is also used with a side project of me: The Hungry? App, an app that uses your location to show you venues such as restaurants close by. In this app you can also save your favorite venue in a list. And this is done on a server. For this server, a REST API, I made this Angular frontend.

With "ng-controller" you define a function that encapsulates our logic that is defined in a separate js file.

You also have integrated form validations. Here the input type is text with a max length of 30. You can also define something like floats or regexs.

# The Controller



```
1 angular.module('hungryUIApp', [])
2   .controller('VenueListController', function () {
3     var ctrl = this;
4     ctrl.venueLists = [
5       {id: 1, name: 'Cool Berlin List'},
6       {id: 2, name: 'Vienna Hipster Places List'}];
7
8     ctrl.addNewVenueList = function () {
9       console.log("ctrl.venueLists", ctrl.venueLists);
10      ctrl.venueLists.push(ctrl.newVenueList);
11      ctrl.newVenueList = {};
12    };
13  });
```

9.12.2015 jquery conf berlin

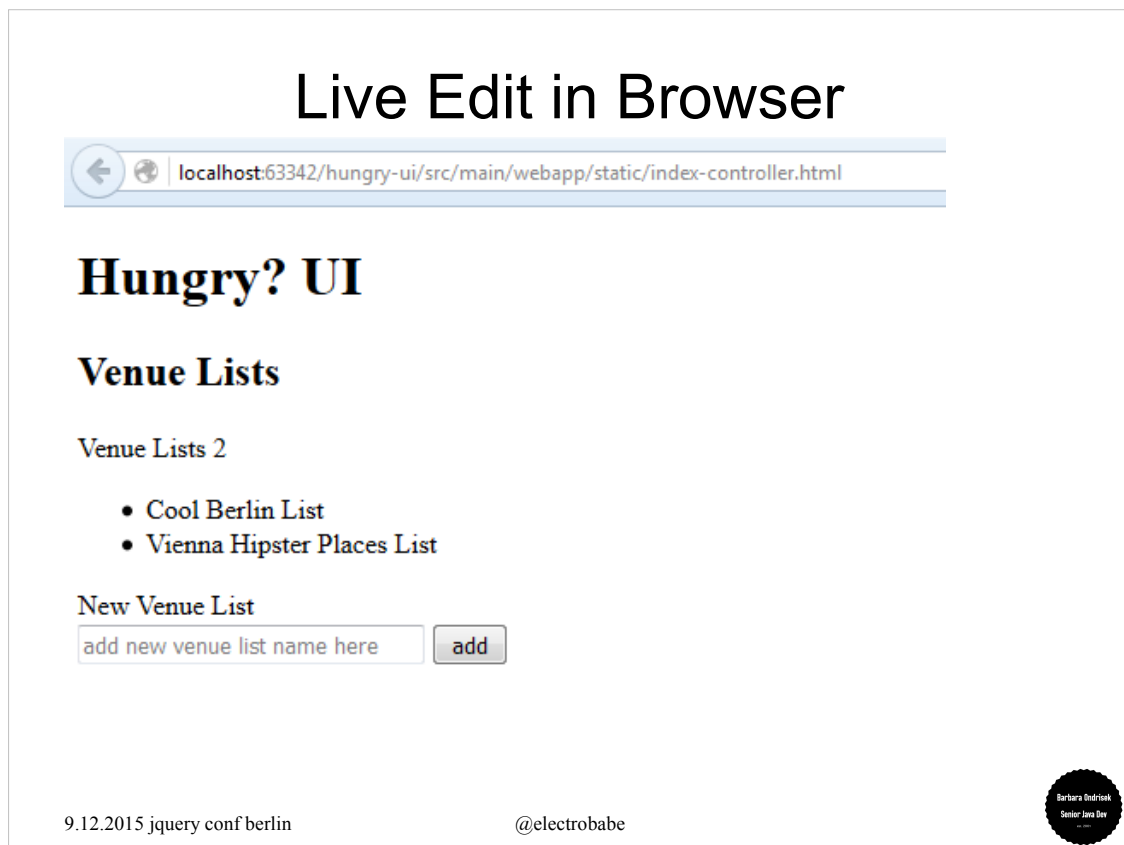
@electrobabe



A typical controller is placed in a separate module – in this case we only have one, that is the app, but in the next example there will be more...

Now you define the controller as a function that contains the app's logic such as scope objects and different functions to handle data.

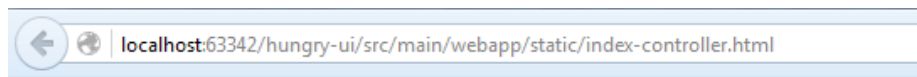
This is a super simple example so let me show you a little bit more.



Here you see...



# Live Edit in Browser



## Hungry? UI

### Venue Lists

Venue Lists 2

- Cool Berlin List
- Vienna Hipster Places List

New Venue List

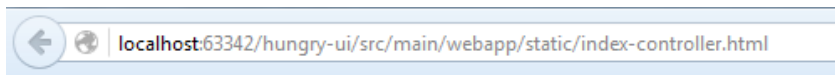
Super Duper London List

9.12.2015 jquery conf berlin

@electrobabe



# Live Edit in Browser



## Hungry? UI

### Venue Lists

Venue Lists 3

- Cool Berlin List
- Vienna Hipster Places List
- Super Duper London List

New Venue List

9.12.2015 jquery conf berlin

@electrobabe



# Use A Model

```
index-resource.html x venues-resource.js x
1 <!doctype html>
2 <html ng-app="hungryUIApp">
3 <head>
4   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
5   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular-resource.min.js"></script>
6   <script src="js/venues-resource.js"></script>
7 </head>
8 <body>
9   <h1>Hungry? UI</h1>
10
11   <h2>Venue Lists</h2>
12
13   <div ng-controller="VenueListController as ctrl">
14     <span>Venue Lists: {{ctrl.venueLists.length}}</span>
15     <ul>
16       <li ng-repeat="venueList in ctrl.venueLists">
17         <span>{{venueList.name}}</span>
18       </li>
19     </ul>
20
21     <h3>New Venue List</h3>
22     <form ng-submit="ctrl.addNewVenueList()">
23       <input type="text" ng-model="ctrl.newVenueList.name" size="30" placeholder="add new venue list name here">
24       <input class="btn-primary" type="submit" value="add">
25     </form>
26   </div>
27 </body>
28 </html>
```

9.12.2015 jquery conf berlin

@electrobabe



## Different Modules in Controller

```

index-resource.html x venues-resource.js x
angular.module('hungryUIApp', ['hungry.controllers']);

// controllers
function VenueListController(VenueList) {
  this.venueLists = VenueList.query();

  this.newVenueList = new VenueList();

  this.addNewVenueList = function () {
    this.newVenueList.$save(function (rst) {
      this.venueLists.push(rst);
      this.newVenueList = new VenueList();
    }).bind(this), function (err) {
      console.log("---err ", err);
    });
  };

  this.deleteVenueList = function (venueList) {
    venueList.$delete().then(function () {
      this.venueLists.remove(venueList);
    }).bind(this);
  }
}

function UserController() {
  // TODO
}

angular.module('hungry.controllers', ['hungry.constants', 'hungry.factories']);

angular.module('hungry.controllers').controller('VenueListController', VenueListController);
angular.module('hungry.controllers').controller('UserController', UserController);

// constants
angular.module('hungry.constants', [])
  .constant('API_PATH', 'http://localhost:8080/hungry-rest');

// factories
var factoryModule = angular.module('hungry.factories', ['ngResource', 'hungry.constants']);
factoryModule.factory('VenueList', VenueListFactory);

function VenueListFactory($resource, API_PATH) {
  return $resource(API_PATH + '/api/venueLists/:id',
    {
      id: 'id'
    });
}

```

9.12.2015 jquery conf berlin

@electrobabe



In this example you see the different modules:  
I defined a module "hungry.controllers", a module "hungry.constants" and "hungry.factories".

The factories module is the model module, in Angular it is called "Factory". With my factory I can fetch data from my REST API. This is accomplished with the resource library that I referenced in the html and injected here into my factory.

I defined a constants module for some fixed values such as the API path and I defined different controllers.

The angular resource library is a jquery tool to fetch data.

# Dependency Injection

```
angular.module('hungryUIApp', ['hungry.controllers']);

// controllers
function VenueListController(VenueList) {
  this.venueLists = VenueList.query();
  this.newVenueList = new VenueList();

  this.addNewVenueList = function () {
    this.newVenueList.$save(function (res) {
      this.venueLists.push(res);
      this.newVenueList = new VenueList();
    }).bind(this), function (err) {
      console.log("---err", err);
    });
  };

  this.deleteVenueList = function (venueList) {
    venueList.$delete().then(function () {
      this.venueLists.remove(venueList);
    }).bind(this);
  }
}

function UserController() {
  // TODO
}

angular.module('hungry.controllers', ['hungry.constants', 'hungry.factories']);
angular.module('hungry.controllers').controller('VenueListController', VenueListController);
angular.module('hungry.controllers').controller('UserController', UserController);

// constants
angular.module('hungry.constants', [])
  .constant('API_PATH', 'http://localhost:8080/hungry-rest');

// factories
var factoryModule = angular.module('hungry.factories', ['ngResource', 'hungry.constants']);
factoryModule.factory('VenueList', VenueListFactory);

function VenueListFactory($resource, API_PATH) {
  return $resource(API_PATH + '/api/venueLists/:id',
    {
      id: '@id'
    });
}
```

9.12.2015 jquery conf berlin

@electrobabe

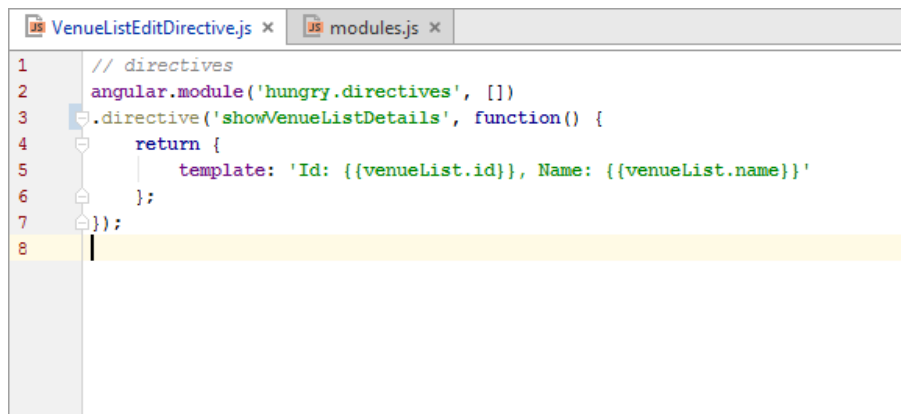


Dependency injection is the most amazing feature of AngularJS.

It allows you to declaratively describe how your application is wired. This means that your application needs no main() method which is usually an unmaintainable mess.

You work modular and can also simply remove or replace components.

# Directives



```
1 // directives
2 angular.module('hungry.directives', [])
3 .directive('showVenueListDetails', function() {
4   return {
5     template: 'Id: {{venueList.id}}, Name: {{venueList.name}}'
6   };
7 });
8
```

9.12.2015 jquery conf berlin

@electrobabe



Directives is a unique and powerful feature available only in Angular. Directives let you invent new HTML syntax, specific to your application.

At a high level, directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler to attach a specified behavior to that DOM element (e.g. via event listeners), or even to transform the DOM element and its children.

Directives are pretty handy, but hard to test because of the HTML code.

This is a simple example. One can also use reusable HTML templates and add lots of complexity. Nice is that the scope is directly separated.





## Localization

An important part of serious apps is localization.  
Angular's locale aware filters and stemming directives give you building blocks to make your application available in all locales.

Angular supports i18n/l10n for date, number and currency filters.

# Deep Linking & Routing

```
routes.js
function Routes($routeProvider) {
  $routeProvider.when('/', {
    templateUrl: 'main.html'
  });

  $routeProvider.when('/venueLists', {
    templateUrl: 'venueList/venueLists.html',
    controller: 'VenueListController as ctrl'
  });

  $routeProvider.when('/users', {
    templateUrl: 'user/userList.html',
    controller: 'UserController as ctrl'
  });
}

angular.module('hungryUIApp').config(['$routeProvider', Routes]);
```

9.12.2015 jquery conf berlin

@electrobabe



## Deep Linking

A deep link reflects where the user is in the app, this is useful so users can bookmark and email links to locations within apps. Round trip apps get this automatically, but AJAX apps by their nature do not. AngularJS combines the benefits of deep link with desktop app-like behavior.

Angular is a single page app, but with these routes you can set ...

## Nested Views

```
1 .row
2   .col-md-12
3     h1 Hungry? UI
4
5     a(href="#/venueLists")
6       | h2 Venue Lists
7
8     a(href="#/users")
9       | h2 Users
10
```

9.12.2015 jquery conf berlin

@electrobabe



For views we use jade templates in our project. And you can nest them inside of each other.

Such as headers, menues, and other reusable components

## Nested Views

```
index.jade x main.jade x header.jade x
1 doctype html
2 html
3   head
4     base(href= '.')
5     meta(charset='utf-8')
6     meta(http-equiv='Cache-Control' content='no-cache, no-store, must-revalidate')
7     meta(http-equiv='Pragma' content='no-cache')
8     meta(http-equiv='Expires' content='0')
9     meta(http-equiv='Vary' content='*')
10    title Hungry? UI
11    link(rel='shortcut icon' href='static/ico/favicon.ico')
12    script(src='static/js/app/libs.js')
13    script(src='static/js/app/main.js')
14    link(href='static/bower_components/jquery-ui-bootstrap/jquery.ui.theme.css' rel='stylesheet')
15
16  body (ng-app='hungryUIApp')
17    div.container#maincontainer
18      block content
```

9.12.2015 jquery conf berlin

@electrobabe



For views we use jade templates in our project. And you can nest them inside of each other.

# Nested Views



The screenshot shows a code editor window titled 'index.jade'. The code is as follows:

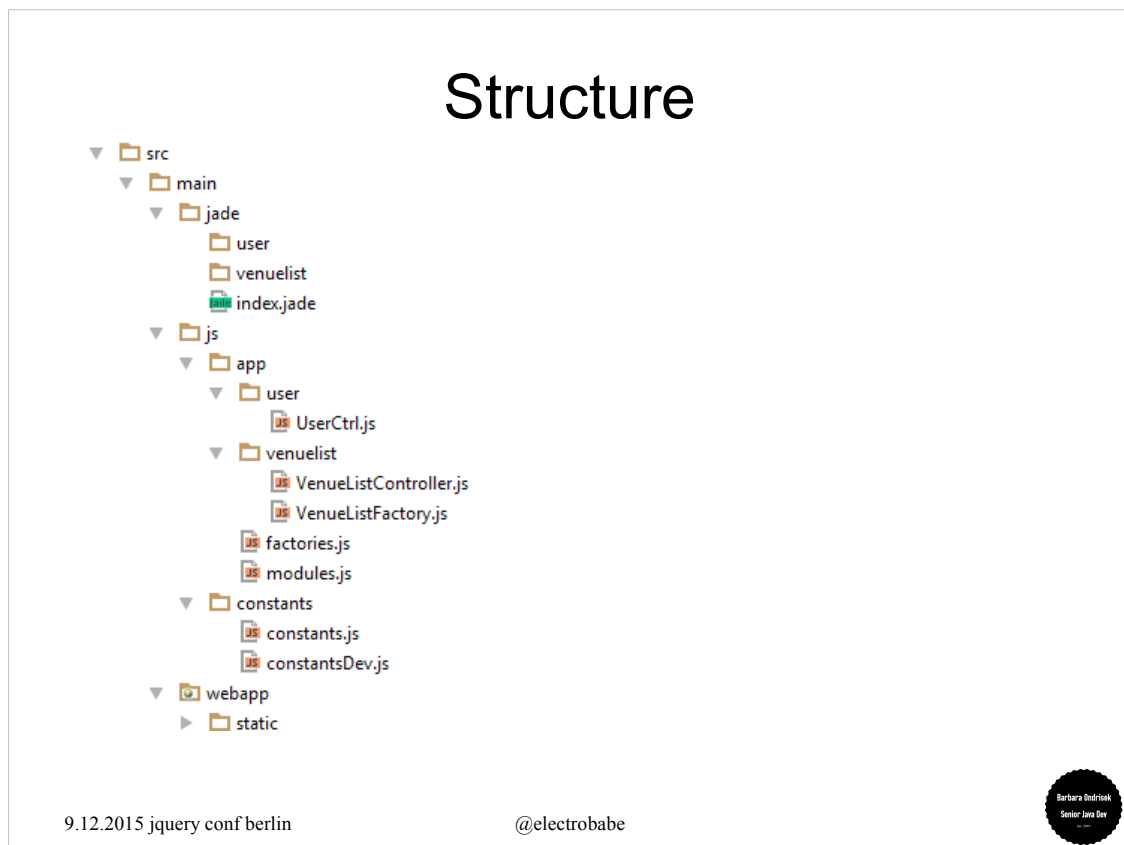
```
1 extends header
2
3 block content
4   div (ng-view)
5
6
```

Line 4, 'div (ng-view)', is highlighted in yellow. Line 6 is also highlighted in yellow. The editor has a light gray background and a dark gray sidebar on the left.

9.12.2015 jquery conf berlin @electrobabe



For views we use jade templates in our project. And you can nest them inside of each other.



Structure by type: controllers, directives, services  
templates

Structure by features



# Testability

```
Terminal
+
x
E:\Dokumente\java\intelliJ\george\george-backoffice\backoffice-ui>gulp
[02:36:02] Using gulpfile E:\Dokumente\java\intelliJ\george\george-backoffice\backoffice-ui\gulpfile.js
[02:36:02] Starting 'webpack'...
[02:36:02] Starting 'jadeStatic'...
[02:36:02] Finished 'jadeStatic' after 11 ms
[02:36:08] [webpack] Hash: 31b3b83c42b61d5c51e6
Version: webpack 1.12.4
Time: 6038ms

   Asset      Size  Chunks             Chunk Names
96f337bc7a3d5d8beeb128ff6630b62.avq  115 kB          0  [emitted]
84af6538466d3d7af69fe5d417254906.png    5.6 kB          0  [emitted]
f4769f9bdc7466be65088239c12046d1.eot   20.1 kB          0  [emitted]
e18bbf611f2a2e43afc071aa2f4e1512.ttf   45.4 kB          0  [emitted]
89889688147b07575d6327160d64e760.avq   109 kB          0  [emitted]
7b53ea75c6b574674895416f5edda9c.eot    24.6 kB          0  [emitted]
ef057bca0253651284caddf19a9a2bd2.png    15 kB          0  [emitted]
57c1d6b3d0e033a133ae670525037a224.png   25.7 kB          0  [emitted]
    bundle.js    255 kB          0  [emitted] bundle
    libs.js     3.44 MB          1  [emitted] libs
    styles.js   53.3 kB          2  [emitted] styles
    templates.js 178 kB          3  [emitted] templates

[02:36:08] Finished 'webpack' after 6.1 s
[02:36:08] Starting 'karma'...
09 12 2015 02:36:12.011:INFO [karma]: Karma v0.13.15 server started at http://localhost:9876/
09 12 2015 02:36:12.029:INFO [launcher]: Starting browser PhantomJS
09 12 2015 02:36:12.623:INFO [PhantomJS 1.9.8 (Windows 7 0.0.0)]: Connected on socket 850wlcPgZm7724hAAA with id 72342548
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 101 of 504 SUCCESS (0 secs / 0.648 secs)
09 12 2015 02:36:14.411:WARN [web-server]: 404: /export/Logos/Logo_12_100.png?1449624974409409
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 115 of 504 SUCCESS (0 secs / 0.805 secs)
09 12 2015 02:36:14.492:WARN [web-server]: 404: /export/Logos/Logo_12_100.png?1449624974436436
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 188 of 504 SUCCESS (0 secs / 1.276 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 205 of 504 SUCCESS (0 secs / 1.423 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 232 of 504 SUCCESS (0 secs / 1.643 secs)
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.186 secs: NGO Factories Spec NGO Spec should refine resource NGO
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.145 secs: Plugin Component List Directive Spec Adding Spec should show edit area when adding
PhantomJS 1.9.8 (Windows 7 0.0.0) SLOW 0.229 secs: PluginFiles directive Spec should upload image for plugin
PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 504 of 504 SUCCESS (5.383 secs / 5.275 secs)
[02:36:19] Finished 'karma' after 11 s
[02:36:19] Starting 'default'...
[02:36:19] Finished 'default' after 5.66 ms

E:\Dokumente\java\intelliJ\george\george-backoffice\backoffice-ui>
```

9.12.2015 jquery conf berlin

@electrobabe



## Testable

AngularJS was designed from ground up to be testable. It encourages behavior-view separation, comes pre-bundled with mocks, and takes full advantage of dependency injection. It also comes with end-to-end scenario runner which eliminates test flakiness by understanding the inner workings of AngularJS.

As tools there are karma and phantom JS. Angular has a built-in mock framework "angular.mock.module".

# Build

```
gulpfile.js x
27  gulp.src(libs)
28    .pipe(concat('libs.js'))
29    .pipe(gulp.dest(DEST + '/static/js/app/'));
30
31  gulp.src([configFile].concat(sources))
32    .pipe(concat('main.js'))
33    .pipe(gulp.dest(DEST + '/static/js/app/'));
34
35
36  gulp.task('concatDev', function () {
37    doConcat('src/main/js/constants/constantsDev.js');
38  });
39
40  gulp.task('jade', function (done) {
41    gulp.src(jadeSrc)
42      .pipe(jade({pretty: true}))
43      .on('error', function (err) {
44        console.log("Jade error: ", err.message);
45      })
46      .pipe(gulp.dest(DEST))
47      .on("end", function () {
48        done();
49      });
50  });
51
52
53  function doWatch(concatTask) {
54    gulp.watch([sources], [concatTask]);
55    gulp.watch(['src/main/jade/**/*.jade'], ['jade']);
56  }
57
58  gulp.task('watch', function () {
59    // ...
60  });
61
62  gulp.task('default', ['concatDev', 'jade']);
63
64
65  // dist
66  gulp.task('uglify', function () {
67    gulp.src([DEST + '/static/js/app/libs.js',
68      concat(['src/main/js/constants/constants.js', DEST + '/static/js/app/main.js'])
69    ])
70    .pipe(uglify())
71    .pipe(concat('main.min.js'))
72    .pipe(gulp.dest(DEST + '/static/js/app/'));
73  });
74
75  gulp.task('concatDist', function () {
76    doConcat('src/main/js/constants/constants.js');
77  });
78
79  gulp.task('dist', ['concatDist', 'jade']);
```

9.12.2015 jquery conf berlin @electrobabe



## Build with gulp!

Gulp: Task runner built on Node.js. Build process (less, coffee script / jasmine, jade)

Gulp + extensions, such as

- babel for ES6 support
- lodash

# Dependency Management

```
bower.json x
1 {
2   "name": "HungryUIApp",
3   "version": "1.21.0",
4   "authors": [
5     "Barbara Ondrisek <office@electrobabe.at>"
6   ],
7   "license": "ISC",
8   "repository": {
9     "type": "git",
10    "url": "git://github.com/electrobabe/hungry-ui.git"
11  },
12  "ignore": [
13    "**/*.*",
14    "node_modules",
15    "bower_components"
16  ],
17  "dependencies": {
18    "angular": "^1.4.7",
19    "angular-ui-bootstrap": "^0.14.2",
20    "jquery-ui-bootstrap": "~0.2.5",
21    "angular-resource": "^1.4.7",
22    "bootstrap": "^3.3.5",
23    "lodash": "^3.10.1"
24  }
25 }
26
```

9.12.2015 jquery conf berlin

@electrobabe



Dependencies are managed by bower resp.  
Webpack.

overview of supporting state of the art frameworks:  
npm FTW!

Bower: Package manager (lodash )

Gulp: Task runner built on Node.js. Build process  
(less, coffee script / jasmine, jade)

Bootstrap UI

## Supporting Frameworks

```
devDependencies
24  "babel-core": "^6.1.2",
25  "babel-loader": "^6.0.1",
26  "babel-preset-es2015": "^6.1.2",
27  "bootstrap": "^3.3.5",
28  "bootstrap-markdown": "^2.9.0",
29  "bower": "^1.3.8",
30  "chai": "^3.3.0",
31  "css-loader": "^0.22.0",
32  "eslint": "^1.7.3",
33  "eslint-loader": "^1.1.0",
34  "event-stream": "^3.3.0",
35  "file-loader": "^0.8.4",
36  "gridster": "^0.5.6",
37  "gulp": "^3.8.11",
38  "gulp-jade": "^1.1.0",
39  "gulp-replace": "^0.5.0",
40  "gulp-util": "^3.0.7",
41  "jade": "^1.11.0",
42  "jade-loader": "^0.8.0",
43  "jasmine-core": "^2.1.2",
44  "jquery": "^2.1.4",
45  "jshint": "^2.8.0",
46  "jshint-stylish": "^1.0.0",
47  "json-loader": "^0.5.4",
48  "karma": "^0.13.15",
49  "karma-beep-reporter": "^0.1.4",
50  "karma-chai": "^0.1.0",
51  "karma-coffee-preprocessor": "^0.3.0",
52  "karma-coverage": "^0.4.2",
53  "karma-jasmine": "^0.3.0",
54  "karma-mocha-reporter": "^1.1.1",
55  "karma-phantomjs-launcher": "^0.2.1",
56  "karma-sinon": "^1.0.4",
57  "less": "^2.5.3",
58  "less-loader": "^2.2.1",
59  "lodash": "^3.10.1",
60  "markdown": "^0.5.0",
61  "moment": "^2.8.4",
62  "phantomjs": "^1.9.18",
63  "should": "^3.1.3",
64  "sinon": "^1.17.1",
65  "style-loader": "^0.13.0",
66  "ui-select": "^0.13.2",
67  "url-loader": "^0.5.6",
68  "webpack": "^1.12.2",
69  "webpack-dev-server": "^1.12.1",
70  "webpack-manifest-plugin": "^0.5.0",
71  "webpack-md5-hash": "^0.0.4",
72  "yaml-loader": "^0.1.0"
```

9.12.2015 jquery conf berlin

@electrobabe



overview of supporting state of the art frameworks:  
npm FTW!

lodash  
Bootstrap UI

Etc.etc.

## Best practices & Lessons Learned



9.12.2015 jquery conf berlin

@electrobabe



No pain, no gain?

Best practices are:

- Define controllers, factories etc. in separate files
- Cache dependencies
- Don't use \$rootScope. At all. Never
- live reload as tool
- Debug in IntelliJ

Lessons Learned

- Integration in mvn, gulp in build (win vs. OSX)
- Test coverage (no integration in sonar?)

# Downsides

9.12.2015 jquery conf berlin

@electrobabe




No Downsides at all!

...just kidding



Downsides



9.12.2015 jquery conf berlin @electrobabe

Barbara Ondrisek  
Senior Java Dev

## Downsides

- Structure is provided but not enforced
- The code is only as good as its programmers

## Why I love Angular

- MVC (MVW)
- Modular design
- Declarative approach
- Active development - community

# I can has code? kthxbye

<https://github.com/electrobabe/hungry-ui>

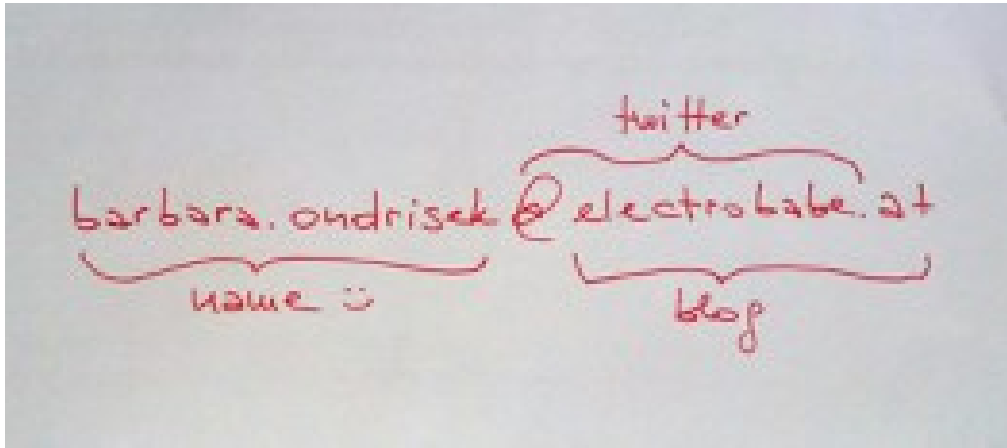


9.12.2015 jquery conf berlin

@electrobabe



## Any Questions?



9.12.2015 jquery conf berlin

@electrolabe



# CATS!



9.12.2015 jquery conf berlin

@electrobabe



## Links

- Slides and Src on Github  
<https://github.com/electrobabe/hungry-ui>
- <http://electrobabe.at>
- <https://play.google.com/store/apps/details?id=at.chefbabe.hungry>
- <https://play.google.com/store/apps/details?id=com.likeahipster.app>