Lecture 5: Introduction to Angular

Olivier Liechti
TWEB

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Today's agenda

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

14h00 - 15h00	60'	**JavaScript 101 reminders (10')** Immediately invoked functions **Lecture (20')** Introduction to AngularJS: modules, scopes, controllers, directives and services. **Activity 1: JSON-P (15'+15')**
15h00 - 15h10	*10'*	*Break*
15h10 - 16h25	75'	**Activity 2: CORS (15' + 15')** **Lecture (10')** Making AJAX requests with AngularJS **Exercise: Angularize your project (25'+10')** Angularize your project home page

What is an immediately invoked anonymous function and why is it useful?

#1 Imagine that we write an HTML page and that we load a first JS script with the following code:

```
<html>
  <body>
    <h1>Do not pollute the global scope</h1>

    <script src="js/tweb-library.js"></script>
  </body>
<html>
```

```
var config = { homeUrl : "http://tweb.heig-vd.ch" };
function fetchData() { ... };
```

#2 The config variable is defined on the global scope. Why is this **dangerous**?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#3 What happens if a second script does the same thing? We have a **name collision** and lose one of the config objects!

```
<html>
  <body>
    <h1>Do not pollute the global scope</h1>

    <script src="js/tweb-library.js"></script>
    <script src="js/graph-library.js"></script>
  </body>
<html>
```

```
var config = { homeUrl : "http://tweb.heig-vd.ch" };
function fetchData() { ... };
```

```
var config = { defaultColor : red };
function drawLogo() { ... };
```

#4 We could ask all developers to adopt naming **conventions**, to support a form of **namespace** (and **hope** for the best...):

```
<html>
  <body>
    <h1>Do not pollute the global scope</h1>

    <script src="js/tweb-library.js"></script>
    <script src="js/graph-library.js"></script>
  </body>
<html>
```

```
var twebConfig = { homeUrl : "http://tweb.heig-vd.ch" };
function fetchData() { ... };
```

```
var graphConfig = { defaultColor : red };
function drawLogo() { ... };
```

#5 A better and safer approach is to hide the variables in an (anonymous) function:

```
function () {
  /*
   * config is now in the scope of a function and we do not pollute
   * the global scope!
   */
  var config = { homeUrl : "http://tweb.heig-vd.ch" };
  function fetchData() { ... };
}
```

#6 But how do we make sure that this code is invoked? Right now, nothing will happen when the HTML page is loaded...?

#7 You can **immediately invoke** your anonymous function!

```
(function () {
  /*
   * config is now in the scope of a function and we do not pollute
   * the global scope!
   */
  var config = { homeUrl : "http://tweb.heig-vd.ch" };
  function fetchData() { ... };
})();
```

You invoke your anonymous function here, so the code above is executed as soon as the browser loads this script

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

ANGULARJS by Google

Introduction

What is AngularJS?

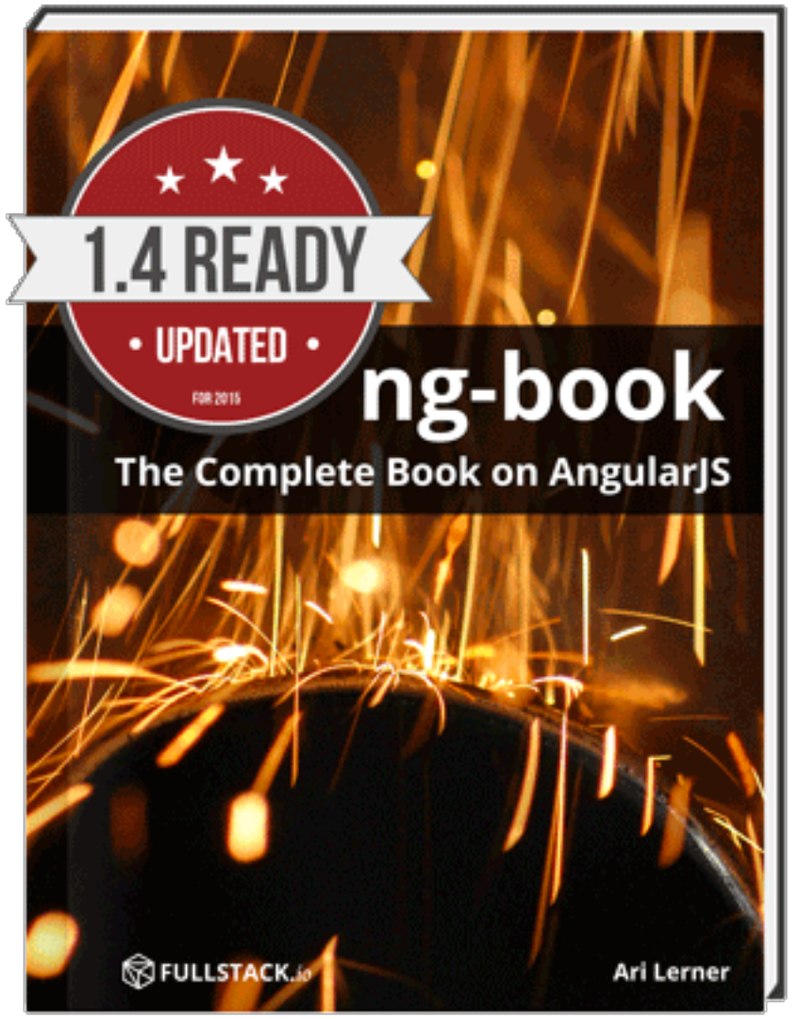
heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- **Client-side** JavaScript framework
- Implements the **MVC design pattern** on the client side
- Designed to create **Single-Page Applications** (SPAs), as opposed to “server-side MVC applications”.
- Initially released in **2009**. Has become one of the most popular frameworks (see job offers!).
- **Large community** (many third-party modules, learning resources, etc.). Open source project with major contributions from **Google**.
- Current version: 1.4.7.
- Angular 2.0 (currently in alpha) will be a major upgrade.

What is the best way to learn AngularJS?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- There are **many books**. I have found this one to be particularly helpful:
  - https://www.ng-book.com/
- There are many sites with **tutorials and webcasts**. Here is a good example:
  - https://egghead.io/technologies/angularjs
- One of the most efficient ways is to study and play with **existing code**:
  - Browse through GitHub repos.
  - Use a yeoman generator
- There are often different ways to do one thing. It is important to adopt coding conventions. See:
  - https://github.com/johnpapa/angular-styleguide



https://www.ng-book.com/

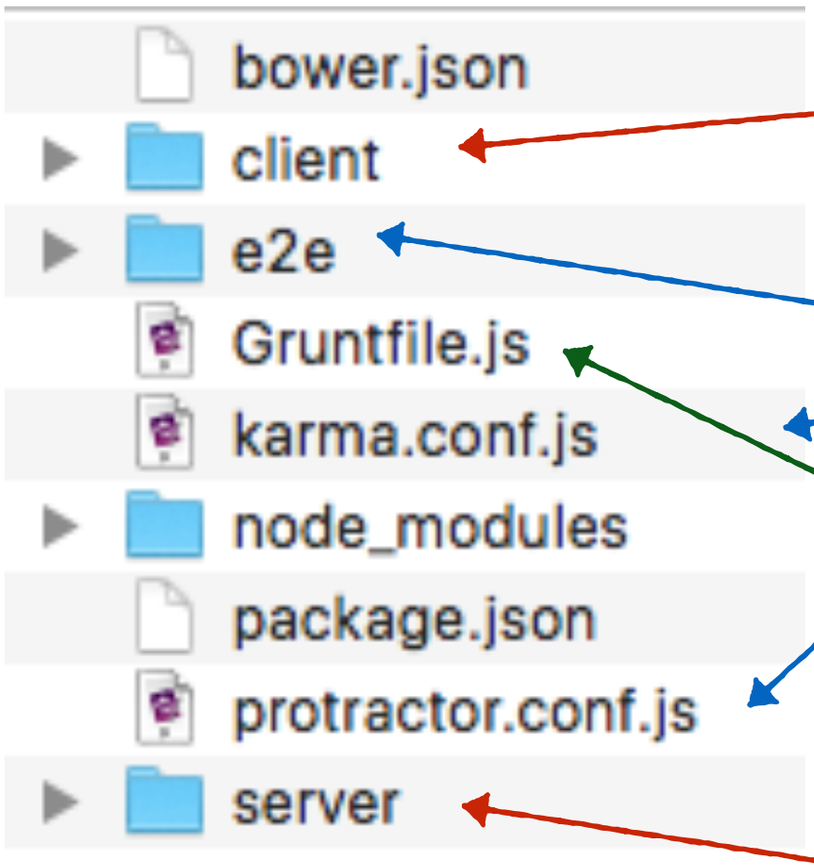
What is the best way to learn AngularJS?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

```
$ mkdir sandbox
$ cd sandbox/
$ yo angular-fullstack
```

Advice: create a "**sandbox**" project. Use it **while you learn** about the AngularJS concepts (modules, controllers, etc.).

Read the code (step-by-step). **Modify the code** to confirm your understanding. Use the **debugger** to really understand what is happening.

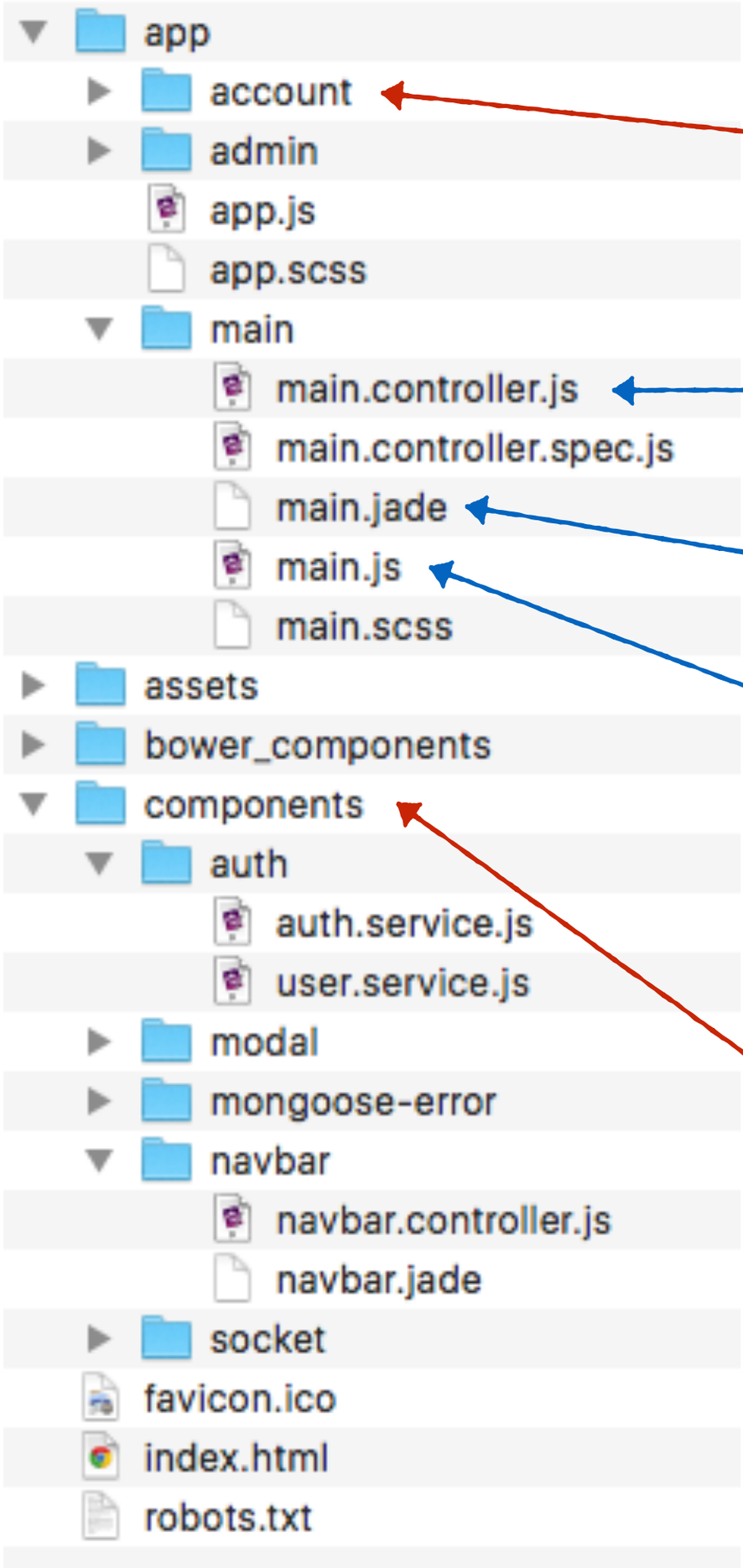


This is the **AngularJS code**. These static files are served by the express.js server and executed in the browser.

These files are used for end-to-end testing (don't worry about this right now)

This is the **build file**. It is powerful but rather complex and a bit overwhelming. You don't need to understand everything at first.

This is the **express.js code**. These scripts are executed on the server side. They serve the static content and implement the REST API (and talk to MongoDB).



The AngularJS components are organized **by feature** (in a hierarchy). This is now the best practice.

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

This file attaches a **controller** to the main module. The controller is used from the main.jade page fragment.

This file defines a **page fragment**. It contains AngularJS **directives**.

This is file is used to manage the **layout** (navigation).

The components folder contains AngularJS scripts that are **reused across features** (mostly services).

How do I bootstrap AngularJS?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- To get started with AngularJS, you first need to **load the core framework** script. You can either use a **CDN**, download the file yourself, or use **bower**.
- You write **your code** in several scripts, which must also be loaded from index.html. In this example, all the code is in one script.

```
<html ng-app="tweb-demo-app">
   <body>
      <h1>Bootstrapping AngularJS</h1>

      <!-- load core AngularJS before any other AngularJS module -->
      <script src="js/angular.js"></script>

      <!-- load my AngularJS module -->
      <script src="js/tweb-demo-app.js"></script>
   </body>
<html>
```

Your script defines a **module** named "tweb-demo-app"

We **declare a new module** and give it a **name** ('twebApp'). Later, we will be able to lookup this module with `angular.module('twebApp')`, in other words by calling the module function without the second argument.

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

```
angular.module('twebApp', [
  'ngCookies',
  'ngResource',
  'ngSanitize',
  'btford.socket-io',
  'ui.router',
  'ui.bootstrap'
])
```

This will **lookup** the twebApp module.

```
<body ng-app="twebApp">

  <!-- build:js({client,node_modules}) app/vendor.js -->
    <!-- bower:js -->
    <script src="bower_components/jquery/dist/jquery.js"></script>
    <script src="bower_components/angular/angular.js"></script>
    <script src="bower_components/angular-resource/angular-resource.js"></script>
    <script src="bower_components/angular-cookies/angular-cookies.js"></script>
    <script src="bower_components/angular-sanitize/angular-sanitize.js"></script>
    <script src="bower_components/angular-bootstrap/ui-bootstrap-tpls.js"></script>
    <script src="bower_components/lodash/dist/lodash.compat.js"></script>
    <script src="bower_components/angular-socket-io/socket.js"></script>
    <script src="bower_components/angular-ui-router/release/angular-ui-router.js"></script>
    <!-- endbower -->
    <script src="socket.io-client/socket.io.js"></script>
  <!-- endbuild -->
```

We **declare** that our module depends on 6 other modules (in this case, they are AngularJS and third-party modules). The corresponding *.js files must be **loaded in index.html**.

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

Double Binding

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

What is a Module?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- When you develop an AngularJS application, you create **controllers**, **services**, **directives**, etc.
- At the minimum, you need to put your components in an application "**module**", which is loaded during the application **bootstrap**.
- If you have a large application, or if you want to share/reuse some of your components, it is a good idea to create **several modules**.
- You can think of modules as "**containers of components**".
- Modules can have **dependencies** on other modules.

This creates a new module, named 'tweb.users'. AngularJS will add it to its registry. The empty brackets mean that the module has no dependency on other modules.

```
angular.module('tweb.users', []);
```

This looks up the module named 'tweb.users' in the AngularJS registry.

```
angular.module('tweb.users');
```

Modules in angular-fullstack

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

1 **Create a new 'twebApp' module**, which depends on other modules. Those starting with "ng" are also provided by AngularJS (but in their own .js files). The others are provided by the community.

app/app.js

```
angular.module('twebApp', [
  'ngCookies',
  'ngResource',
  'ngSanitize',
  'btford.socket-io',
  'ui.router',
  'ui.bootstrap'
])
```

2 **Lookup the 'twebApp' module**, so that we can attach a controller to it.

```
angular.module('twebApp')
  .controller('MainCtrl', function ($scope, $http, socket) {
    $scope.awesomeThings = [];

    $http.get('/api/things').success(function(awesomeThings) {
      $scope.awesomeThings = awesomeThings;
      socket.syncUpdates('thing', $scope.awesomeThings);
    });

    ...
  })
```

app/main/main.controller.js

Naming conventions for modules

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- You could give any **name** to your Angular modules.
- However, it is important that you **define and follow some conventions** if you don't want to loose control of your codebase (and if you want new developers to easily understand it!)

```
angular.module('tweb.security', []);

angular.module('tweb-security', []);

angular.module(‘twebSecurity’, []);
```

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

What is a Directive?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- An AngularJS directive is an **HTML extension** (e.g. a custom element, a custom attribute, which you include in your markup to **trigger some behavior**.
- AngularJS comes with a collection of **built-in directives**.
- **Third-party developers** have created additional directives.
- **You** can write your own directives.

Directive components in `ng`

Name	Description
ngJq	Use this directive to force the angular.element library. This should be used to force either jqLite by leaving ng-jq blank or setting the name of the jquery variable under window (eg. jQuery).
ngApp	Use this directive to **auto-bootstrap** an AngularJS application. The `ngApp` directive designates the **root element** of the application and is typically placed near the root element of the page - e.g. on the `<body>` or `<html>` tags.
a	Modifies the default behavior of the html A tag so that the default action is prevented when the href attribute is empty.
ngHref	Using Angular markup like `{{hash}}` in an href attribute will make the link go to the wrong URL if the user clicks it before Angular has a chance to replace the `{{hash}}` markup with its value. Until Angular replaces the markup the link will be broken and will most likely return a 404 error. The `ngHref` directive solves this problem.
ngSrc	Using Angular markup like `{{hash}}` in a `src` attribute doesn't work right: The browser will fetch from the URL with the literal text `{{hash}}` until Angular replaces the expression inside `{{hash}}`. The `ngSrc` directive solves this problem.
ngSrcset	Using Angular markup like `{{hash}}` in a `srcset` attribute doesn't work right: The browser will fetch from the URL with the literal text `{{hash}}` until Angular replaces the expression inside `{{hash}}`. The `ngSrcset` directive solves this problem.
ngDisabled	This directive sets the `disabled` attribute on the element if the expression inside `ngDisabled` evaluates to truthy.

https://docs.angularjs.org/api/ng/directive

Which directives will use quickly?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

ngApp	Use this directive to **auto-bootstrap** an AngularJS application. The ngApp directive designates the root element of the application and is typically placed near the root element of the page - e.g. on the **<body>** or **<html>** tags.
ngController	The ngController directive **attaches a controller class to the view**. This is a key aspect of how angular supports the principles behind the **Model-View-Controller** design pattern.
ngModel	The ngModel directive **binds an input,select, textarea** (or custom form control) to a **property on the scope** using NgModelController, which is created and exposed by this directive.
ngRepeat	The ngRepeat directive **instantiates a template once per item from a collection**. Each template instance **gets its own scope**, where the given loop variable is set to the current collection item, and $index is set to the item index or key.
ngClick	The ngClick directive allows you to specify **custom behavior when an element is clicked**.
ngInclude	Fetches, compiles and includes an **external HTML fragment**.
ngClass	The ngClass directive allows you to **dynamically set CSS classes** on an HTML element by databinding an expression that represents all classes to be added.

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

What is a Scope?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- An Angular **scope** is a JavaScript **object**, created by the framework.
- It has **properties**, some of which are **functions**. The properties can be displayed in the view. The functions can be called from the view.
- Scopes are created at different levels of the **DOM** (e.g. at the level of a <DIV> node).
- Scopes are organized in a **prototypal inheritance chain**:
  - A scope often extends another scope.
  - The common ancestor of most scopes (i.e. non isolated scopes) is called **$rootScope**.

```
{
   'title' : 'TWEB',
   'getMessage' : function() {
     return this.title;
   }
}
```

prototypal inheritance

```
{
   'subTitle' : ‘Web Technologies',
   'getMessage' : function() {
     return this.title + ", " +
        this.subtitle;
   }
}
```

These 3 directives create a new scope

What is a Scope?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

```
<div class="show-scope-demo">
  <div ng-controller="GreetController">
    Hello {{name}}!
  </div>
  <div ng-controller="ListController">
    <ol>
      <li ng-repeat="name in names">{{name}} from {{department}}</li>
    </ol>
  </div>
</div>
```

The scope created by this directive inherits from the scope created by ListController, which inherits from $rootScope

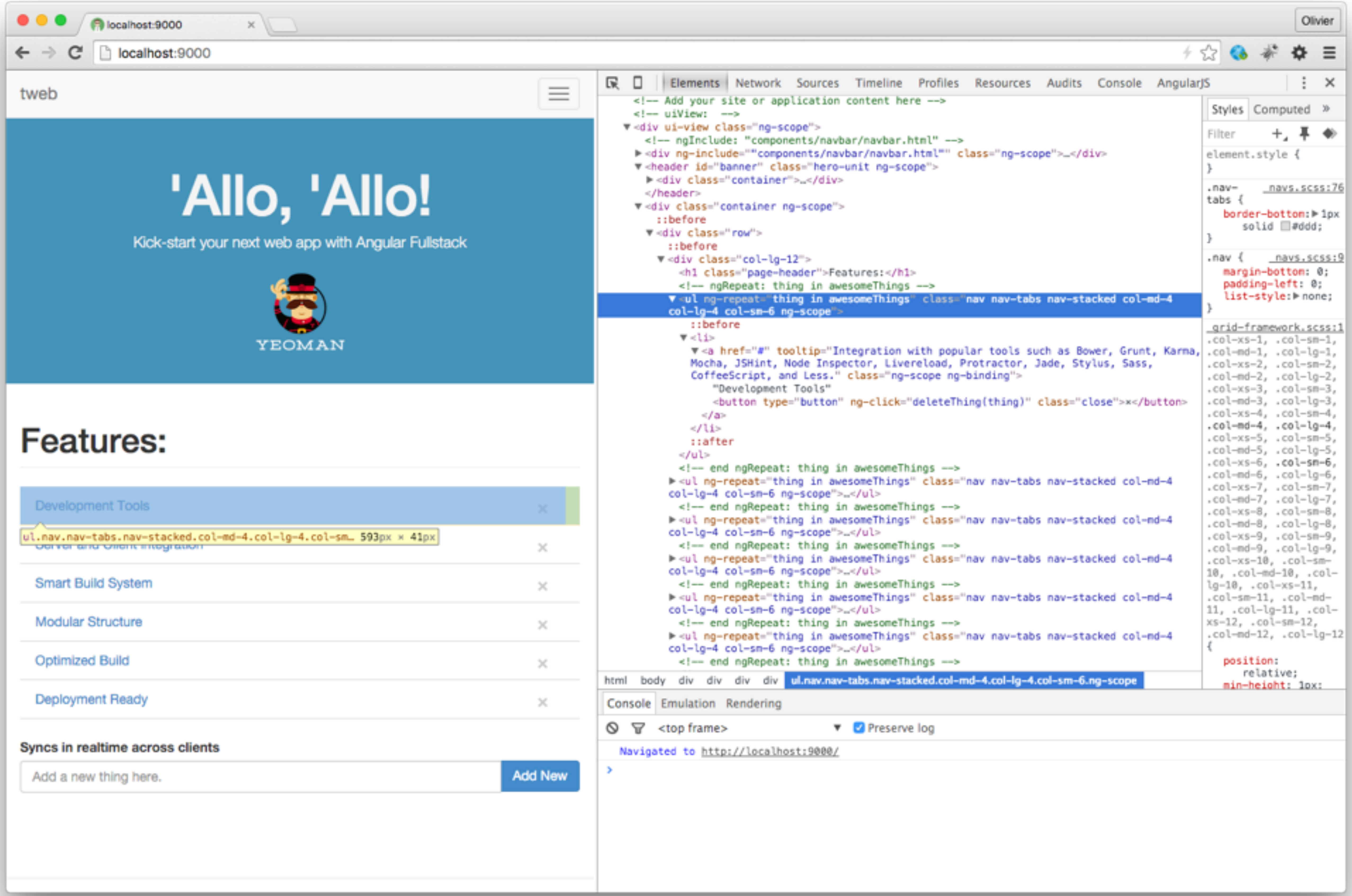
Hello World!

1. Igor from Angular
2. Misko from Angular
3. Vojta from Angular

```
angular.module('scopeExample', [])

.controller('GreetController', ['$scope', '$rootScope',
function($scope, $rootScope) {
  $scope.name = 'World';
  $rootScope.department = 'Angular';
}])

.controller('ListController', ['$scope', function($scope) {
  $scope.names = ['Igor', 'Misko', 'Vojta'];
}]);
```



tweb

'Allo, 'Allo!

Kick-start your next web app with Angular Fullstack

Features:

- Development Tools ×
- Server and Client Integration ×
- Smart Build System ×
- Modular Structure ×
- Optimized Build ×
- Deployment Ready ×

```
ul.nav.nav-tabs.nav-stacked.col-md-4.col-lg-4.col-sm… 593px × 41px
```

Syncs in realtime across clients

Add a new thing here. | Add New

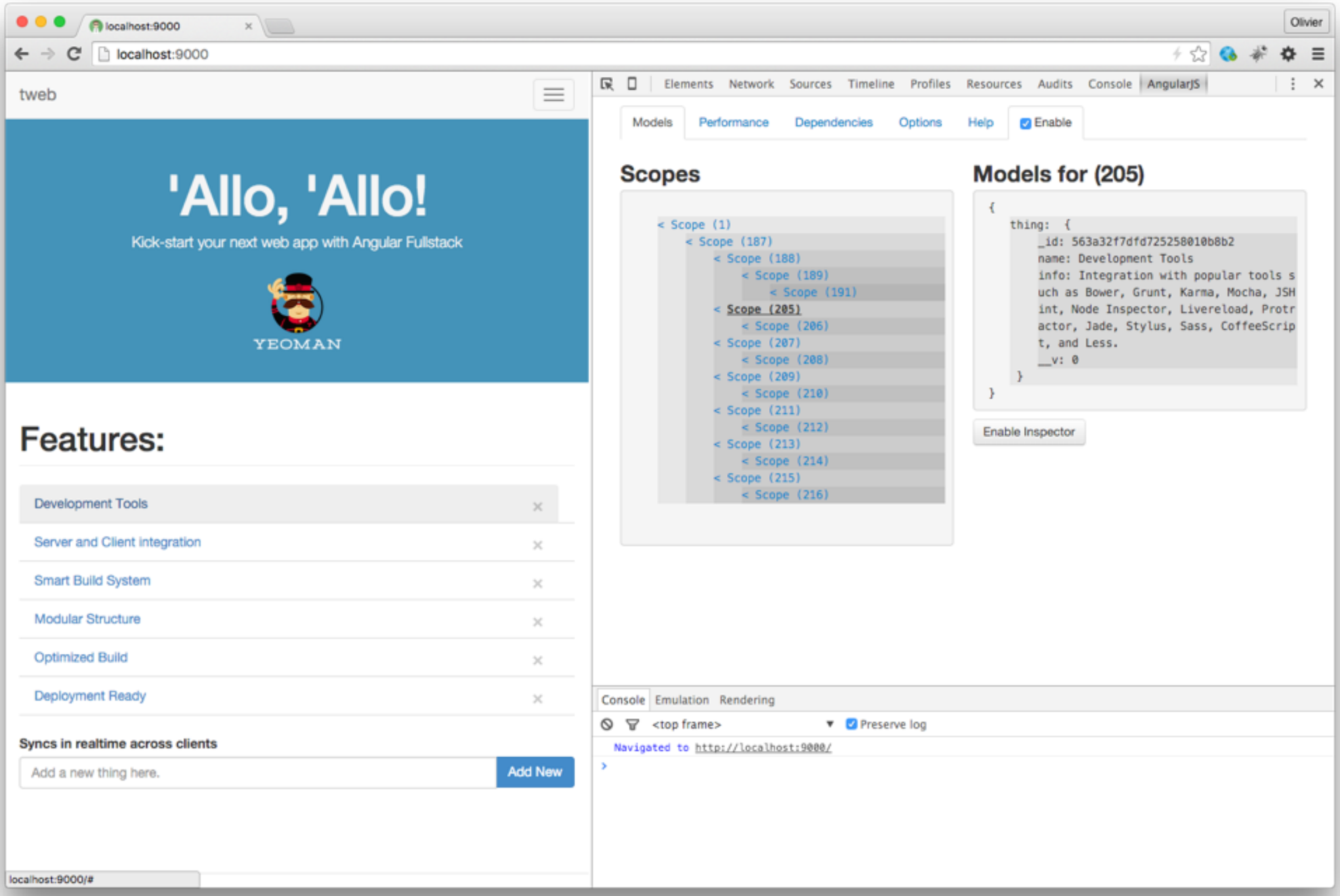
```
<!-- Add your site or application content here -->
<!-- uiView:  -->
▼<div ui-view class="ng-scope">
   <!-- ngInclude: "components/navbar/navbar.html" -->
  ▶<div ng-include=""components/navbar/navbar.html"" class="ng-scope">…</div>
  ▼<header id="banner" class="hero-unit ng-scope">
    ▶<div class="container">…</div>
   </header>
  ▼<div class="container ng-scope">
     ::before
    ▼<div class="row">
       ::before
      ▼<div class="col-lg-12">
         <h1 class="page-header">Features:</h1>
         <!-- ngRepeat: thing in awesomeThings -->
        ▼<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">
           ::before
          ▼<li>
            ▼<a href="#" tooltip="Integration with popular tools such as Bower, Grunt, Karma,
            Mocha, JSHint, Node Inspector, Livereload, Protractor, Jade, Stylus, Sass,
            CoffeeScript, and Less." class="ng-scope ng-binding">
                "Development Tools"
                <button type="button" ng-click="deleteThing(thing)" class="close">×</button>
              </a>
           </li>
           ::after
         </ul>
         <!-- end ngRepeat: thing in awesomeThings -->
        ▶<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">…</ul>
         <!-- end ngRepeat: thing in awesomeThings -->
        ▶<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">…</ul>
         <!-- end ngRepeat: thing in awesomeThings -->
        ▶<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">…</ul>
         <!-- end ngRepeat: thing in awesomeThings -->
        ▶<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">…</ul>
         <!-- end ngRepeat: thing in awesomeThings -->
        ▶<ul ng-repeat="thing in awesomeThings" class="nav nav-tabs nav-stacked col-md-4
        col-lg-4 col-sm-6 ng-scope">…</ul>
         <!-- end ngRepeat: thing in awesomeThings -->
```

```
html  body  div  div  div  div  ul.nav.nav-tabs.nav-stacked.col-md-4.col-lg-4.col-sm-6.ng-scope
```

```
element.style {
}
.nav-          navs.scss:76
tabs {
   border-bottom:▶1px
      solid ☐#ddd;
}
.nav {          navs.scss:9
   margin-bottom: 0;
   padding-left: 0;
   list-style:▶none;
}
  grid-framework.scss:1
.col-xs-1, .col-sm-1,
.col-md-1, .col-lg-1,
.col-xs-2, .col-sm-2,
.col-md-2, .col-lg-2,
.col-xs-3, .col-sm-3,
.col-md-3, .col-lg-3,
.col-xs-4, .col-sm-4,
.col-md-4, .col-lg-4,
.col-xs-5, .col-sm-5,
.col-md-5, .col-lg-5,
.col-xs-6, .col-sm-6,
.col-md-6, .col-lg-6,
.col-xs-7, .col-sm-7,
.col-md-7, .col-lg-7,
.col-xs-8, .col-sm-8,
.col-md-8, .col-lg-8,
.col-xs-9, .col-sm-9,
.col-md-9, .col-lg-9,
.col-xs-10, .col-sm-
10, .col-md-10, .col-
lg-10, .col-xs-11,
.col-sm-11, .col-md-
11, .col-lg-11, .col-
xs-12, .col-sm-12,
.col-md-12, .col-lg-12
{
   position:
      relative;
   min-height: 1px;
```

Console   Emulation   Rendering

```
<top frame>  ▼  ☑ Preserve log
Navigated to http://localhost:9000/
>
```


localhost:9000
Olivier
localhost:9000
tweb
'Allo, 'Allo!
Kick-start your next web app with Angular Fullstack
YEOMAN
Features:
Development Tools
Server and Client integration
Smart Build System
Modular Structure
Optimized Build
Deployment Ready
Syncs in realtime across clients
Add a new thing here.
Add New
localhost:9000/#
Elements Network Sources Timeline Profiles Resources Audits Console AngularJS
Models Performance Dependencies Options Help Enable
Scopes
< Scope (1)
< Scope (187)
< Scope (188)
< Scope (189)
< Scope (191)
< Scope (205)
< Scope (206)
< Scope (207)
< Scope (208)
< Scope (209)
< Scope (210)
< Scope (211)
< Scope (212)
< Scope (213)
< Scope (214)
< Scope (215)
< Scope (216)
Models for (205)
{
thing: {
_id: 563a32f7dfd725258010b8b2
name: Development Tools
info: Integration with popular tools such as Bower, Grunt, Karma, Mocha, JSHint, Node Inspector, Livereload, Protractor, Jade, Stylus, Sass, CoffeeScript, and Less.
__v: 0
}
}
Enable Inspector
Console Emulation Rendering
<top frame>
Preserve log
Navigated to http://localhost:9000/

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controller>
  {{ msg }}
  <a ng-click="greet"></a>
</div>
```

Services

Modules

What is a Controller?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- An AngularJS controller is used to **initialize a scope** and to **attach behavior** (functions) to it.

This will create a new scope, which will be managed by an instance of a controller named “SpicyController”.

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

We can access the scope properties and invoke the functions.

```
<div ng-controller="SpicyController">
 <input ng-model="customSpice">
 <button ng-click="spicy('chili')">Chili</button>
 <button ng-click="spicy(customSpice)">Custom spice</button>
 <p>The food is {{spice}} spicy!</p>
</div>
```

This adds a controller named “SpicyController” to our “spicyApp2” module.

This initializes the “customSpice” and the “spice” properties (they will be available in the view).

```
var myApp = angular.module('spicyApp2', []);

myApp.controller('SpicyController', ['$scope', function($scope) {
    $scope.customSpice = "wasabi";
    $scope.spice = 'very';

    $scope.spicy = function(spice) {
        $scope.spice = spice;
    };
}]);
```

This adds function to the scope. It will be available in the view.

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

How does AngularJS implement MVC?

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

M

Scopes chain

```
{
  msg : 'hello',
  greet : function() {}
}
```

C

Controller

```
$scope.msg = 'hello';
$scope.greet = function() {};
```

V

HTML **page** or **fragment** with **directives, expressions** and **filters**

```
<div ng-controll
  {{ msg }}
  <a ng-cli greet"></a>
</div>
```

Services

Modules

What is a Service?

- AngularJS services are **singleton objects** that can be injected in controllers and that provide some functionality.
- It is a good practice to keep **controllers small**. For this reason, most of the complex behavior should be delegated to a service.
- A good example is the code that deals with **AJAX** requests.
- AngularJS provides a list of **built-in services**.
- You can implement **your own services**.

service

$anchorScroll
$animate
$animateCss
$cacheFactory
$compile
$controller
$document
$exceptionHandler
$filter
$http
$httpBackend
$httpParamSerializer
$httpParamSerializerJQLike
$interpolate
$interval
$locale
$location
$log
$parse
$q
$rootElement
$rootScope
$sce
$sceDelegate
$templateCache
$templateRequest
$timeout
$window
$xhrFactory

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

AJAX, Security & Same Origin Policy

Activity 1: JSON-P

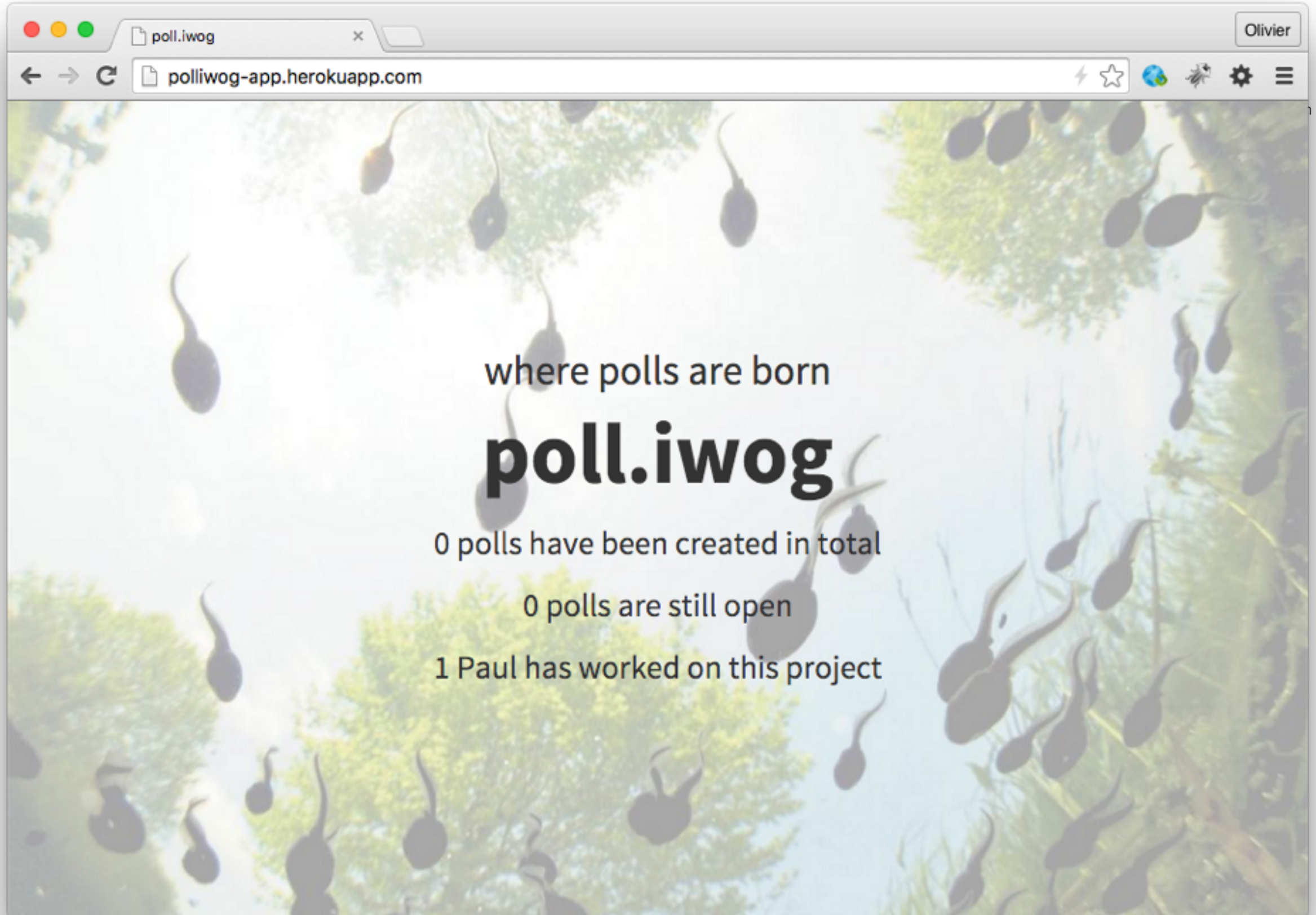
heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- **Start by reading and analyzing in details the following article:**
  - http://schock.net/articles/2013/02/05/how-jsonp-really-works-examples/
- **Prepare answers to the following questions:**
  - What is the **problem** addressed by JSON-P? Illustrate with a concrete example.
  - What needs to be done on the **client side** in order to implement JSON-P (explain what happens at the **lowest level** and how libraries can help)?
  - What needs to be done on the **server side** in order to implement JSON-P?
- **You have 15 minutes to prepare yourself. I will ask a few students to present their solutions.**
- **Other helpful resources:**
  - https://developer.github.com/v3/#json-p-callbacks
  - https://johnnywey.wordpress.com/2012/05/20/jsonp-how-does-it-work/
  - http://www.uitrick.com/javascript/jsonp-and-its-usages/

Activity 2: CORS

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- **Start by reading and analyzing in details the following article:**
  - http://www.eriwen.com/javascript/how-to-cors/
  - Prepare answers to the following questions:
  - What is the **problem** addressed by CORS? Illustrate with a concrete example.
  - What needs to be done on the **client side** in order to implement CORS?
  - What needs to be done on the **server side** in order to implement CORS?
  - Illustrate the process with a sequence diagram.
- **You have 15 minutes to prepare yourself. I will ask a few students to present their solutions.**
- **Other helpful resources:**
  - https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS
  - http://www.w3.org/TR/cors/#introduction
  - http://www.html5rocks.com/en/tutorials/cors/


poll.iwog
Olivier
polliwog-app.herokuapp.com
where polls are born
poll.iwog
0 polls have been created in total
0 polls are still open
1 Paul has worked on this project

```
HTTP/1.1 200 OK
Content-type: text/html
...
```

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

```
if poll_count == 1
  h3 #{poll_count} poll has been created in total
else
  h3 #{poll_count} polls have been created in total
if active_count == 1
  h3 #{active_count} poll is still open
else
  h3 #{active_count} polls are still open
h3 1 Paul has worked on this project
```

views/stats.jade

```
GET / HTTP/1.1
Host: ...
Accept: text/html
```

```
// All other routes are redirected to landing page
app.get('/*', function stat_view(req, res) {
    var Poll = require('./api/poll/poll.model');
    Poll.find(function (err, polls) {
        if(err) {return res.status(500).send(err); }
        var active_count = polls.filter(function (x) { return x.active; }).length;
        res.render('stats', {poll_count: polls.length, active_count: active_count});
      });
  });
```

routes.js

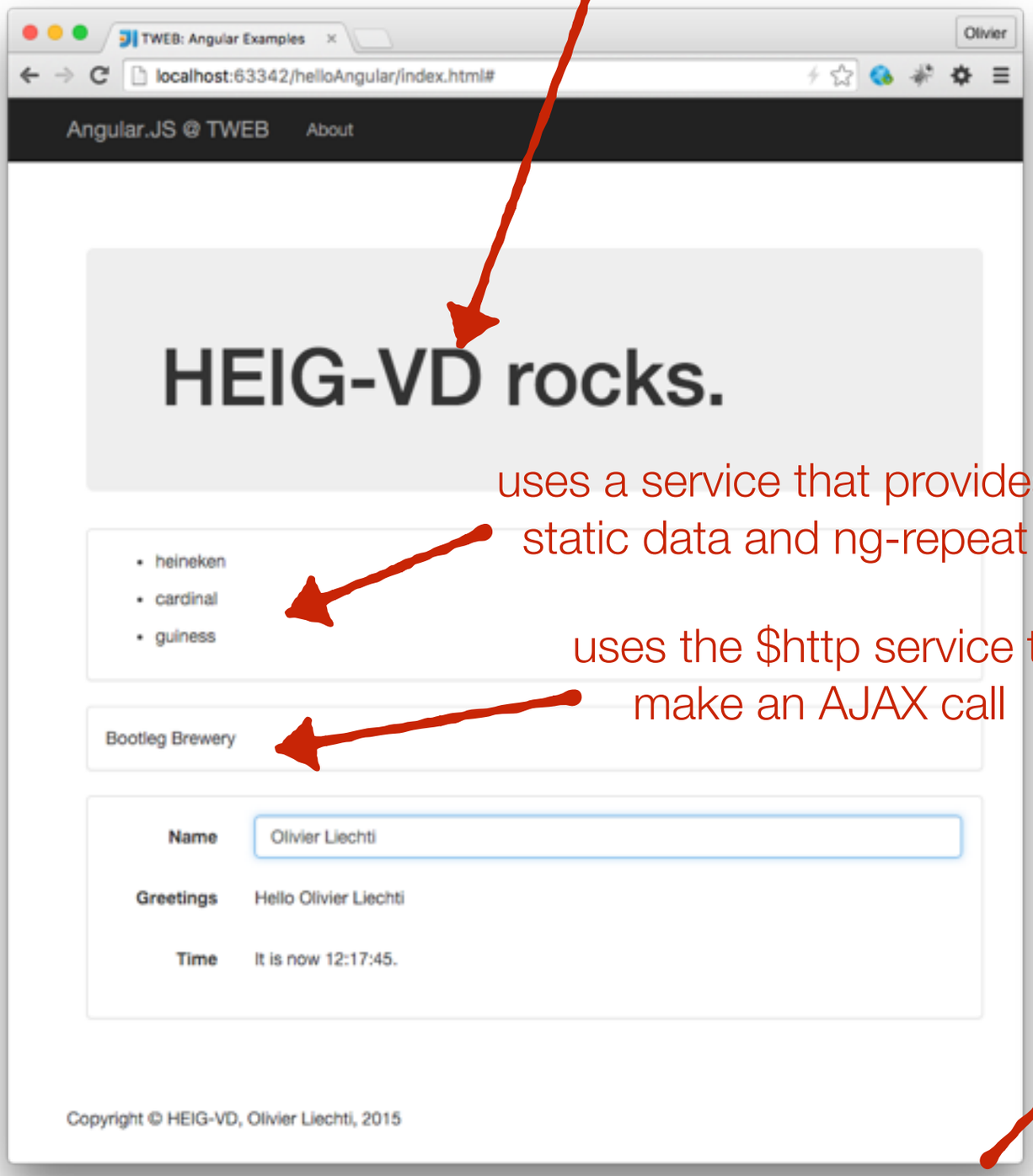
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

https://github.com/SoftEng-HEIGVD/Teaching-HEIGVD-TWEB-Examples-Angular

Binding from scope to view
(expression displays scope property)

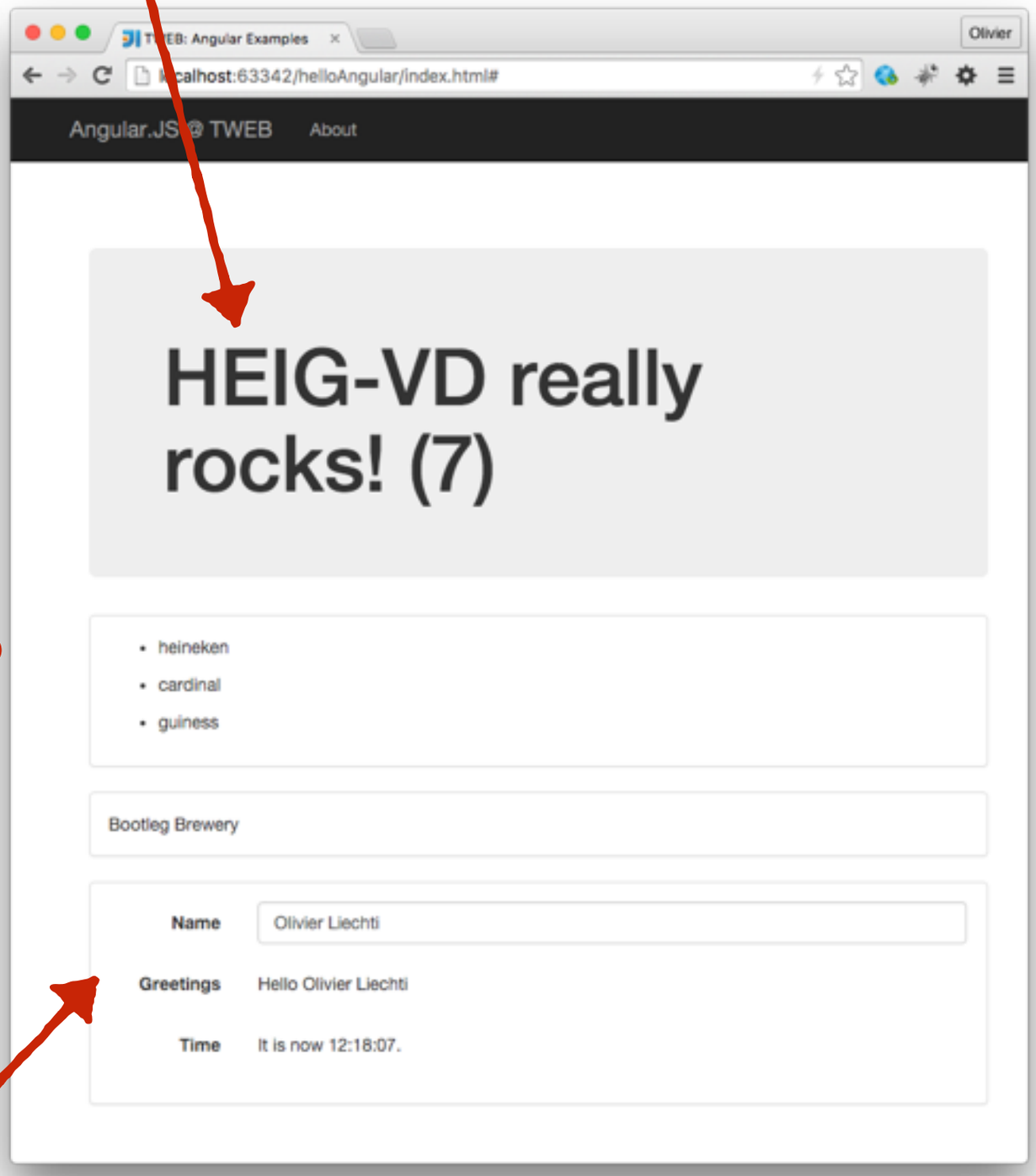
ng-click calls a function attached to the scope

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud



uses a service that provides static data and ng-repeat

uses the $http service to make an AJAX call



Double binding with ng-model: input field updates the scope and expression displays updated value in Greetings