ANGULARJS – THE NEWEST TECHNOLOGY IN CREATING WEB APPLICATIONS

Radu BUCEA-MANEA-TONIŞ
Hyperion University Bucharest, Calea Calarasi, No. 169, District 3,
Bucharest, Romania, Tel.: 0213274465, Fax: 0213216296,
Email: radub_m@yahoo.com

Abstract
This article is the result of searching and selecting new technologies that help programmer in developing web applications. It also represents a pleading for using it, showing its advantages and disadvantages. Alongside the article there are features regarding binding elements, modules, filters and directives. It is a synthesis and a guide of good practice for innovative programmers. All technical issues presented are supported by a case study.

Keywords: *AngularJS; JavaScript; MVC.*
JEL Classification: M15

Introduction

AngularJS is a complete JavaScript-based open-source front-end MVC (Model-View-Controller) framework created by Miško Hevery and maintained by a community of individuals and corporations[1]. AngularJS makes developing single-page web applications easy even for those not accustomed with JavaScript or JQuery. Even more, it provides the separation of application logic, data models, and views – a task that needed employment of big frameworks (ASP.NET, JavaEE with Tomcat, etc.) in the past. Also, it is suitable for service integration, and inherits dependency injection design pattern from aspect oriented programming.

---

[1] https://en.wikipedia.org/wiki/AngularJS.

AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and approximately 8,400 other sites out of 1 million tested in July 2015[2].

1. **How to Employ Angular in Web Apps** [Lerner, 2013]
We simply have to include angular.js in our HTML and explicitly set the ng-app attribute on an element in the DOM:

<div ng-app="myApp">

The ng-app attribute declares that everything inside of it belongs to this Angular app. Only the DOM elements inside the one with the ng-app attribute (eg. <div>) will be affected by Angular.

The controller is a function that adds additional functionality to the web app. Declaring the ng-controller attribute on a DOM element says that all of the elements inside of it belong to the controller:

<div ng-app="myApp" ng-controller="MyController">

The scope object is simply a JavaScript object whose properties are all available to the view and with which the controller can interact:

E.g.:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4
.8/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="SelectController">

<select  ng-model="select_country"  ng-options="country
for country in countries">
<option value="">Choose Country</option></select>
```

---

[2] libscore.com. Retrieved 2016-07-13.

```
</div>

<script>
var app = angular.module('myApp', [])
.controller('SelectController', function($scope) {
$scope.countries    =    ["Romania",    "Belgium",
"Washington"]; })
</script>

</body>
</html>
```
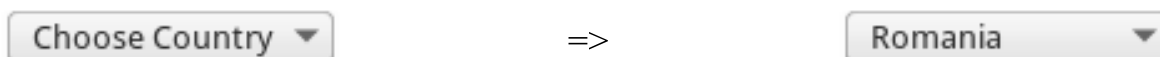
Choose Country ▼   =>   Romania ▼

Figure no. 1. AngularJS combo control

When we define custom behaviour for a web app, we set the execution context by designing methods in our controllers, and properties in the views.

There are two ways of updating the scope when the view modifies it, being sure that the view will immediately update when the scope changes:

1. By using expressions, ex. {{...}}

E.g.:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4
.8/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="LocController">

<ul>
<li ng-repeat="person in people"> {{person.name}} lives
in {{person.city}} </li></ul>
</div>
```

```
<script>
var app = angular.module('myApp', [])
.controller('LocController', function($scope) {
$scope.people = [
{name: "Radu", city: "Bucuresti"},
{name: "Emil", city: "Braila"} ]; })
</script>

</body>
</html>
```

- Radu lives in Bucuresti
- Emil lives in Braila

Figure no. 2. Result of ng-repeat directive with curly braces one way binding style

2. By using ng-bind directive, where language is defined in controller as scope attribute:

E.g.:
```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4
.8/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="LangController">

<p ng-bind="language"></p>

</div>

<script>
var app = angular.module('myApp', [])
.controller('LangController', function($scope) {
$scope.language="English"; })
</script>
```

```
</body>
</html>
```

English

Figure no. 3. One way binding with ng-bind directive

or you can use theng-model directive on HTML controls to bind the model to the view that provides a two-way binding between the model and the view:

E.g.:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4
.8/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="UserController">

Name: <input ng-model="user">
<p ng-bind="user"></p>

</div>

<script>
var app = angular.module('myApp', [])
.controller('UserController', function($scope) {
$scope.user = "Guest"; })
</script>

</body>
</html>
```

First time, the input control is filled with Guest,

Issue 3/2016

Figure no. 4. Autofilled input control from controller attribute

when we change the value to Radu, the paragraph element will show Radu, thus changing the value of the name scope attribute:

Figure no. 5. Setting controller attribute with input control value

The module is a container for the different parts of an application (controllers). We declare a module by using angular.module() method:

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

We need to pass two parameters to the method when we declare a module:
1) the name of the module we are creating;
2) the list of dependencies, otherwise known as injectables that can be used to define dependent modules.
Filters can be applied in view using binding curly braces:

E.g.:
```
<li ng-repeat="person in people | filter : 'Radu">
{{person.name | uppercase}} lives in {{person.city |
lowercase}} </li>
```
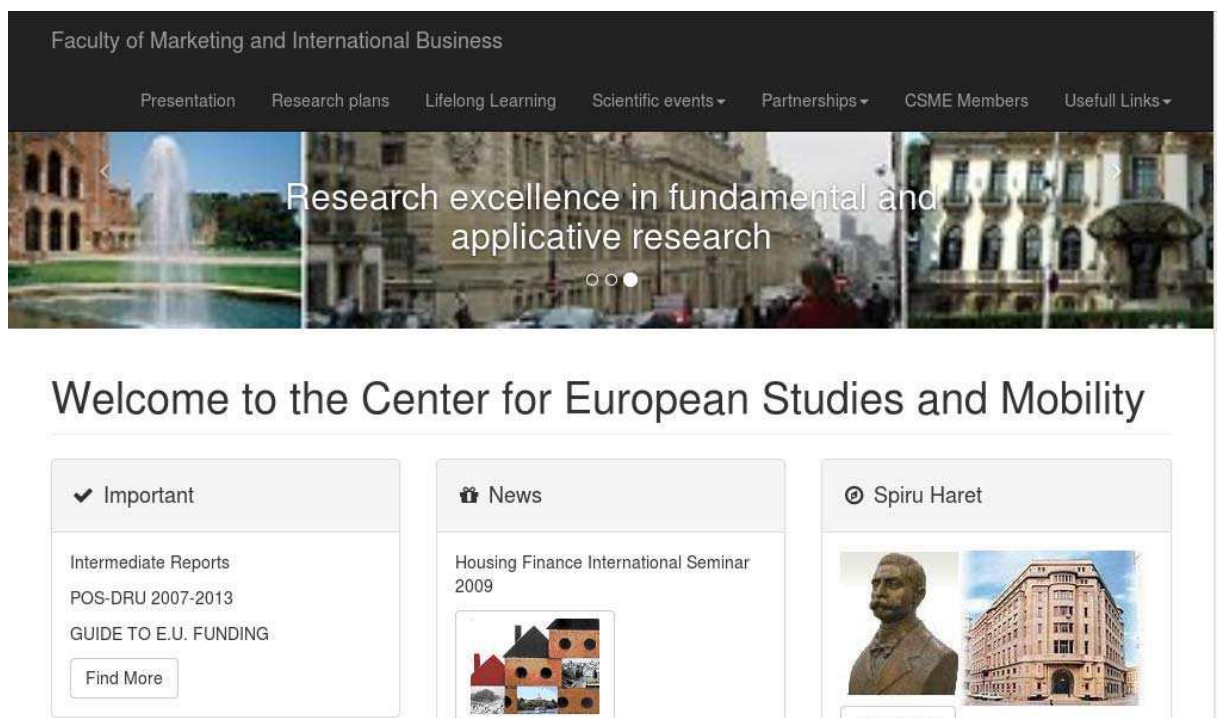
108

or in controller by using filter service:

```
E.g.:
var app = app.controller('NameController', ['$scope',
'$filter', function($scope, $filter) {
$scope.name = $filter('lowercase')('Radu');
}]);
```

2. **Case study**
We have created a new website, using bootstrap library (Figure no. 6):

Figure no. 6. Website home page

We have decided to generate dynamic content using AngularJS and JSON for the About web page:

Figure no. 7. The About page of the site was populated with data using AngularJS

The source code for the About page of the site reveals the technology used and a sample of it is presented in the following paragraph:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<meta name="description" content="CSME">
<meta name="author" content="Radu & Rocsana Manea">
<title>CSME</title>
<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
...
<!-- Angularjs -->
```

```
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4
.8/angular.min.js"></script>
</head>
<body>
...
<!-- Content Column -->
     <div              ng-app="myApp"              ng-
controller="AboutController">
<div class="col-md-9">
          <table width="100%"><tr>
          <td><h2>Center  for  European  Studies  and
Mobility</h2></td>
          <td><img          src="./images/siglaSc.jpg"
/></td></tr></table>
          <h4     ng-repeat="quote     in     quotes">
{{quote.no}} . {{quote.text}} </h4>
</div>
     </div>
<script>
var app = angular.module('myApp', [])
.controller('AboutController', function($scope) {
$scope.quotes = [
{no: "1", text: "Research, innovation, education"},
{no: "2", text: "Education integration of research in
lifelong learning"},
{no: "3", text: "Research excellence in fundamental and
applicative research"},
{no: "4", text: "Inovation methods, inovative models in
knowledge based economy"} ]; })
</script>
</div>

...
</body></html>
```

Conclusions

Web applications evolved from client server architecture to service oriented architecture and finally to microservices architecture pattern. AngularJS along side with Bootstrap (a CSS framework) and NodeJS just

fills the gap created by this transition, allowing developers creating robust and complex web applications, with minimum amount of time and effort. AngularJS can be downloaded from https://angularjs.org/.org/.

References
1. https://angularjs.org/.org/
2. https://en.wikipedia.org/wiki/AngularJS
3. Lerner, Ari, ng-book – The Complete Book on AngularJS (Fullstack, 2013).
4. www.libscore.com. Retrieved 2016-07-13