Modern Development of E-business Application with AngularJS

Pistol, L.; Bucea Manea Tonis, R. & Bucea Manea Tonis, R.

Abstract: *Modern E-business web applications are now developed using new technologies such as NodeJS, AngularJS and Bootstrap. The article identifies and solves practical issues regarding the JSON exchanged messages between NodeJS and AngularJS within SMEs E-business environment.*

Key words: *JSON, AngularJS, NodeJS, E-Business, REST*

Authors´ data: Univ.Prof.Dr. **Pistol**, L[uminita]*; Univ.Lecturer.Dr **Bucea-Manea-Tonis**, R[adu]**, Univ.Lecturer.Dr **Bucea-Manea-Tonis**, R[ocsana]***, *Spiru Haret University, Ion Ghica 13, 030045, Bucharest, Romania,**Hyperion University, Călărași 169, 030615, Bucharest, Romania, ***Spiru Haret University, Ion Ghica 13, 030045, Bucharest, Romania, luminita_pistol@yahoo.com, DPN: 64130, radub_m@yahoo.com, DPN:23039, rocsense39@yahoo.com, DPN:23040.

This Publication has to be referred as: Pistol, L[uminita]; Bucea Manea Tonis, R[adu] & Bucea Manea Tonis, R[ocsana] (2017). Modern Development of E-business Application with AngularJS, Chapter 16 in DAAAM International Scientific Book 2017, pp.209-216, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-12-9, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2017.16

1. Introduction

Cross-border business-to-business (B2B) e-commerce portals offer important advantages for early-mover adopters. The advantages refer at amplifying the inverted U shape effect of early-mover (Deng, 2016). After a scientific analysis the authors recommends that these platforms to be develop in bootstrap technology with java functionalities, based on AngularJS and NodeJs.

Software developers enjoy modern technologies like NodeJS, JSON (JavaScript Object Notation), AngularJS. In cross site scripting, they encounter some difficulties regarding the JSON middleware, using Angular/Node Javascript on client-server architecture.

The authors noticed from practice that new technologies like AngularJS and NodeJS are not fully documented, many design flows being neglected by the end users' perspective.

The authors embraced this Javascript technology and tried to promote it making it suitable even in the most unfriendly environment, providing solutions for the encountered errors.

The result of the authors' research and their practical attempt was a dynamic web report with full AJAX search capabilities on a SMEs economical data structure.

After (Lamb, 2016), AngularJS has some advantages, comparing with JQuery:

- RESTfull API - we may create an AngularJS controller module that will consume a REST service, by using $http get service method.
- MVC Pattern Support - 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.
- Templating - we may specify the html that will be appended or replaced when we define a custom directive by using an inline template.
- Two way data binding - we can use the `ng-model` directive on HTML controls to bind the model to the view, that provides a two-way binding between the model and the view.
- Dependency management - we need to pass a list of dependencies, known as injectables when we declare a module.
- Deep-link routing - we can enable users to bookmark and share specific pages by using the $routeProvider
- Form validation, eg:
  $scope.fields = [
  {placeholder: 'Username', isRequired: true},
  {placeholder: 'Password', isRequired: true}];
- Integration test runner - it was designed to be testable during the loops of lifecycle software development, having even a unit test runner (Karma) and an end-to-end test framework (Protractor). With the Automated QUality Analysis(AQUA) AngularJS in very easy to be integrated in any step of the software lifecycle.

2. Case Study – Filtering a Romanian SMES data structure with AngularJS

It had been used NodeJS runtime environment to feed a JSON response for a client AngularJS application:

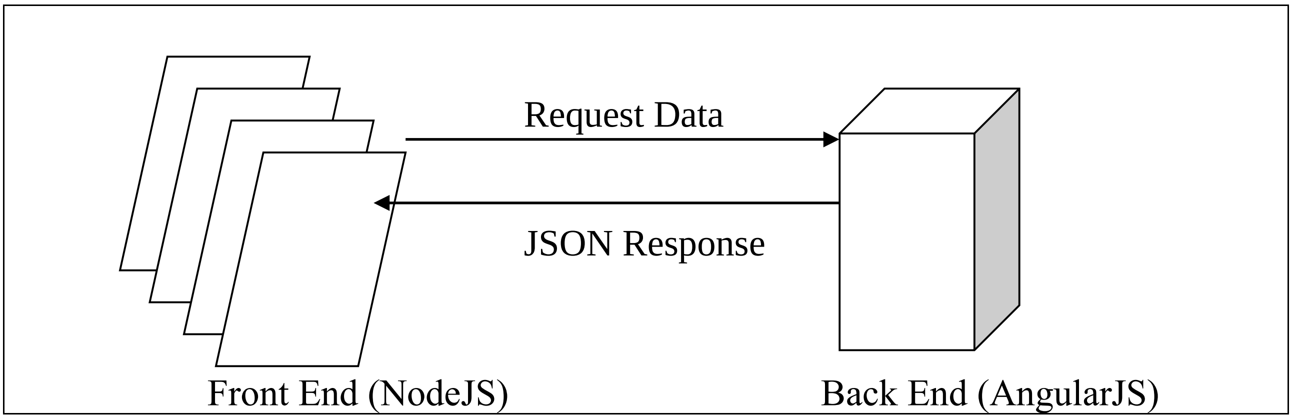


Fig. 1. Front End - Back End communication between NodeJS and AngularJS

Unlike in the case of XML, JSON files are a convenient option for storing data - JSON files being less cluttered and easy-to-read-making information easily to be accessed by the users (Mayernick etal., 2015). JSON can either be represented as a hash of properties and values (Object), for eg.

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

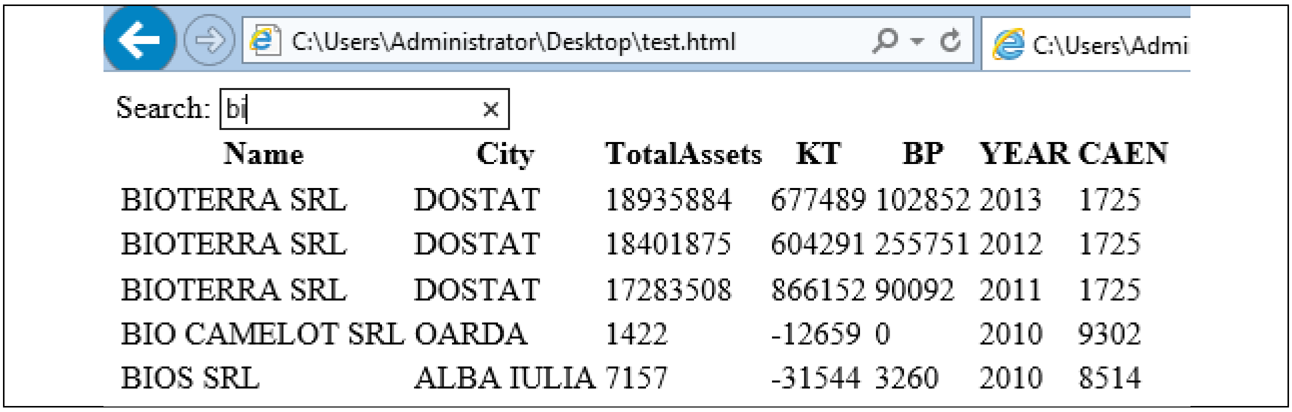
or as a list of values (Array), for eg. (Bucea-Manea-Tonis, 2016)

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

The server script is inspired from (Tutorial Point, 2016) and adapted to the SMEs data structure.

```
//load the http module and store the returned HTTP
instance into a http variable
var http = require("http");

//create a server instance and then bind it to port 8089
using the listen method associated with the server
instance
http.createServer(function (request, res) {
   // Send the HTTP header
   // HTTP Status: 200 : OK
   // Content Type: text/plain
   // Website you wish to allow to connect
    res.setHeader('Access-Control-Allow-Origin',
'http://localhost:8089');

    // Request methods you wish to allow
    res.setHeader('Access-Control-Allow-Methods', 'GET,
POST, OPTIONS, PUT, PATCH, DELETE');

    // Request headers you wish to allow
    res.setHeader('Access-Control-Allow-Headers', 'X-
Requested-With,content-type');

    // Set to true if you need the website to include
cookies in the requests sent
    // to the API (e.g. in case you use sessions)
    res.setHeader('Access-Control-Allow-Credentials',
true);

    res.setHeader('Content-Type', 'application/json');
    res.end(JSON.stringify({ "records":[
    {"Name":"AXEL COMPANY
SRL","City":"ABRUD","TotalAssets":"3023661","KT":"774274"
,"BP":"272622","YEAR":"2013","CAEN":"5152"},
    {"Name":"DORA LACT
SRL","City":"AIUD","TotalAssets":"170245","KT":"72446","B
P":"86814","YEAR":"2013","CAEN":"1725"},
    {"Name":"BIOTERRA
SRL","City":"DOSTAT","TotalAssets":"18935884","KT":"67748
9","BP":"102852","YEAR":"2013","CAEN":"1725"},
    {"Name":"AGRA OARDA
SRL","City":"OARDA","TotalAssets":"562319","KT":"241077",
"BP":"48852","YEAR":"2013","CAEN":"1713"},
    {"Name":"AXEL COMPANY
SRL","City":"ABRUD","TotalAssets":"2554304","KT":"671730"
,"BP":"162494","YEAR":"2012","CAEN":"5152"},
    {"Name":"DORA LACT
SRL","City":"AIUD","TotalAssets":"87328","KT":"9368","BP"
:"8655","YEAR":"2012","CAEN":"1725"},
    {"Name":"DANI
SRL","City":"TOTOI","TotalAssets":"44286","KT":"4873","BP
":"30915","YEAR":"2012","CAEN":"1725"},
    {"Name":"BIOTERRA
SRL","City":"DOSTAT","TotalAssets":"18401875","KT":"60429
1","BP":"255751","YEAR":"2012","CAEN":"1725"},
    {"Name":"AGRA OARDA
SRL","City":"OARDA","TotalAssets":"518773","KT":"202300",
"BP":"92680","YEAR":"2012","CAEN":"1713"},
    {"Name":"AGRO ITAL
SRL","City":"OIEJDEA","TotalAssets":"893389","KT":"245128
8","BP":"636","YEAR":"2012","CAEN":"1713"},
    {"Name":"DORA LACT
SRL","City":"AIUD","TotalAssets":"155971","KT":"714","BP"
:"26500","YEAR":"2011","CAEN":"1725"},
    {"Name":"DANI
SRL","City":"TOTOI","TotalAssets":"116561","KT":"35788","
BP":"3298","YEAR":"2011","CAEN":"1725"},
    {"Name":"BIOTERRA
SRL","City":"DOSTAT","TotalAssets":"17283508","KT":"86615
2","BP":"90092","YEAR":"2011","CAEN":"1725"},
    {"Name":"AGRA OARDA
SRL","City":"OARDA","TotalAssets":"486761","KT":"124449",
"BP":"83525","YEAR":"2011","CAEN":"1713"},
    {"Name":"AGRO ITAL
SRL","City":"OIEJDEA","TotalAssets":"891254","KT":"245192
3","BP":"14114","YEAR":"2011","CAEN":"1713"},
    {"Name":"GLADIOLA COMPROD
SRL","City":"BLANDIANA","TotalAssets":"50624","KT":"20091
","BP":"4777","YEAR":"2011","CAEN":"1713"},
    {"Name":"BIO CAMELOT
SRL","City":"OARDA","TotalAssets":"1422","KT":"-
12659","BP":"0","YEAR":"2010","CAEN":"9302"},
    {"Name":"BLUE COMPANY SRL","City":"ALBA
IULIA","TotalAssets":"116089","KT":"-100292","BP":"-
82481","YEAR":"2010","CAEN":"9271"},
    {"Name":"BEST SOUND RADIO
ENERGY","City":"CUGIR","TotalAssets":"17897","KT":"-
33769","BP":"5310","YEAR":"2010","CAEN":"9220"},
    {"Name":"BIOS SRL","City":"ALBA
IULIA","TotalAssets":"7157","KT":"-
31544","BP":"3260","YEAR":"2010","CAEN":"8514"}
..............................................................................................................................................
    ] }));
    }).listen(8089);
```

The case study is based on data regarding Romanian SME's, and contains the company names, the city and the financial data (total assets, total capital, gross profit, and CAEN - classification of national economic activities), available online at http://www.smesonline.eu/indexen.html. In the figure 2 there is a sample of Romanian SMEs. It was also implemented a filter on this specific data structure that can be applied to any available field.



Search: bi

Name	City	TotalAssets	KT	BP	YEAR	CAEN
BIOTERRA SRL	DOSTAT	18935884	677489	102852	2013	1725
BIOTERRA SRL	DOSTAT	18401875	604291	255751	2012	1725
BIOTERRA SRL	DOSTAT	17283508	866152	90092	2011	1725
BIO CAMELOT SRL	OARDA	1422	-12659	0	2010	9302
BIOS SRL	ALBA IULIA	7157	-31544	3260	2010	8514

Fig. 2. Filter on Company name on Romanian SMEs data

AngularJS makes developing web application easy even for those not accustomed with Javascript or JQuery, providing the separation of application logic, data models, and views (MVC).

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, by using expressions {{…}} or by using ng-bind directive, for eg. `<p>{{language}}</p>` instead of `<p ng-bind="language"></p>`

Filters can be applied in view using binding curly braces or in controller by using filter service, Eg:

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

We decided to use the first alternative, as may be seen in the client code bellow, adapted after (w3schools.com, 2016).

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

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

<label>Search: <input ng-model="searchText"></label>
<table>

<tr><th>Name</th><th>City</th><th>TotalAssets</th><th>KT<
/th><th>BP</th><th>YEAR</th><th>CAEN</th></tr>
  <tr ng-repeat="x in myData | filter:searchText"">
```

```
<td>{{x.Name}}</td><td>{{x.City}}</td><td>{{x.TotalAssets
}}</td><td>{{x.KT}}</td><td>{{x.BP}}</td><td>{{x.YEAR}}</
td><td>{{x.CAEN}}</td>
  </tr>
</table>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("http://localhost:8089/").then(function
(response) {
      $scope.myData = response.data.records;
  });
});
</script>

</body>
</html>
```

The $http service is a core Angular service that facilitates communication with the remote HTTP servers via the browser's XMLHttpRequest object or via JSONP. The $http service is a function that takes a URL used to generate an HTTP request and returns a response object with data and status properties. (Angularjs, 2016)

Comments - Errors that may appear in RESTfull applications:
1. Error: Access to restricted URI denied
Motive: A resource makes a cross-origin HTTP request when it requests a resource from a different domain than the one which the first resource itself serves.
Solution: The Cross-Origin Resource Sharing standard works by adding new HTTP headers that allow servers to describe the set of origins that are permitted to read that information using a web browser (Perrier, 2016).

```
  // Website you wish to allow to connect
   res.setHeader('Access-Control-Allow-Origin', 'http://localhost:8089');
   // Request methods you wish to allow
   res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT,
PATCH, DELETE');
   // Request headers you wish to allow
   res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type');
   // Set to true if you need the website to include cookies in the requests sent
   // to the API (e.g. in case you use sessions)
   res.setHeader('Access-Control-Allow-Credentials', true);
```

2. Error: CORS header 'Access-Control-Allow-Origin' does not match
'http://localhost:8089
Motive: Firefox and Chrome require exact domain specification in Access-Control-Allow-Origin header. For servers with authentication, these browsers do not allow "*" in this header (WebDAV, 2016)
Solution: Use IE instead

3. Conclusion

This paper aims to demonstrate the advantages of the new wave of web technologies and provide some practical solutions to avoid errors in cross site scripting. The article reveals the advantages of using these technologies for cross-border business-to-business (B2B) e-commerce portals.

4. References

Angularjs, (2016) Available from: https://docs.angularjs.org/api/ng/service/$http Accessed: 2016-10-12.

Bucea-Manea-Tonis, R. (2016). Angular JS – The Newest Technology in Creating Web Applications. Annals of "Spiru Haret". Economic Series, 16(3). Retrieved from: http://anale.spiruharet.ro/index.php/economics/article/view/745

Bucea-Manea-Tonis, R. (2016), Romanian SMEs database Available from: http://www.smesonline.eu/indexen.html Accessed: 2016-10-10.

IT Hit WebDAV System (2016), Cross-Origin Requests (CORS) in Internet Explorer, Firefox, Safari and Chrome, Available from: https://www.webdavsystem.com/ajax/programming/cross_origin_requests Accessed: 2016-10-5.

Lamb, M. jQuery vs. AngularJS: A Comparison and Migration Walkthrough, (2015) Available from: https://www.airpair.com/angularjs/posts/jquery-angularjs-comparison-migration-walkthrough#2-migration-walkthrough

Mayernick, M. & Nichols H. (2015). How to Use JSON files in Node.js, Available from: https://www.codementor.io/nodejs/tutorial/how-to-use-json-files-in-node-js Accessed: 2016-10-9.

Perrier, J-Y (2016). HTTP access control (CORS), Available from: https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS, Accessed: 2016-9-12.

Tutorials point, (2016). Simply easy learning, Available from: https://www.tutorialspoint.com/nodejs/nodejs_first_application.htm Accessed: 2016-10-12.

Ziliang, D. & Zeyu, W., (2016) Early-mover advantages at cross-border business-to-business e-commerce portals, Journal of Business Research, 69(3), pp. 6002-6011

W3schools, (2016) Available from: http://www.w3schools.com/angular/tryit.asp?filename=try_ng_customers_json, Accessed: 2016-10-15.