

AngularJS: Tutorial: Part 4: Directive Example



Posted by Alejandro G. Carlstein Ramos Mejia on November 2, 2015 November 3, 2015 About Programming / AngularJS / JavaScript

In [Part 1](#), we went over the basics on AngularJS. In [Part 2](#), we went over AngularJS and Charts. In [Part 3](#), we went over common mistakes when developing using AngularJS. In this part of the tutorial, I wish to share with you a small directive I am building. I apologize in advance if you see my code a little messy. I have a shortage of free time so I built this in a hurry. I am not including the back-end such as PHP, MySQL, and REST using JSON. That would be part of another tutorial.

Let's begin with this draft of project. The idea is to use JSON as the model that will indicate the view to use a template which itself uses the UI feature called accordion in a nested way to display all the work experience I have:

Let's begin with the JSON, workExperience.json:

```
[
{
  "IT Consultant - Vestal, New York (June 2014 - Present)":
  {
    "<p><i>Self-Employed</i></p><p>Providing services to the company Hourglass Applications. Involved in a variety of applications and research development.</p><p><u>Project Examples:</u></p>" : [
      { "Jog 101" :
        "Designed, coded, and tested using Swift with Core Data, GeoLocation, MapKit, and MediaPlayer."},
      { "Browser Alexia" :
        "Designed and coded in collaboration using Objective-C and Swift with UIWebView, WKWebView and JavaScript injection."}
    ]}
},
{
  "The Raymond Corporation - Greene, New York (March 2012 - May 2014)":
  {
    "<p><i>Programmer Analyst IV</i></p><p>The Raymond Corporation is a Toyota material handling solution provider company.<br>Acted as project leader designing, coding, testing, documenting and maintaining large complex programs. Provided leadership, support, and technical guidance to all levels of personnel over technical, programming, and system related issues.</p><p><u>Project Examples:</u></p>" : [
      { "IBM WebSphere Cast Iron" :
        "Designed, coded, tested, documented and maintained integrations which synchronize information between different databases such as AS400, MS SQL Servers, and cloud solutions such as Oracle CRM on Demand and Big Machines using SOAP and REST web services."},
      { "Suppliers Center" :
        "Maintained and expanded website running on Apache Tomcat 5.5 using JavaScript, jQuery, and JSP with functionalities that included accessing multiple databases (MS SQL Server, AS400, and MS Access 97), reporting, plus providing email services."},
    ]}
}
```

```

{ "Asaichi Strips" :
  "Designed, coded, tested, documented and maintained website running on ColdFusion
8.0 and 9.0.2 accessing databases (MS SQL Server and AS400) and providing export
services (CSV, XLS, and PDF). Agile programming approach similar to SCRUM was used when
building this system."},
{ "Contractor Manager System" :
  "Redesigned RMIS for Human Resources department, coded using PrimeFaces framework,
tested, documented and maintained website running on Apache Tomcat 7.0/8.0 with access
to MS SQL 2012 Server database."},
{ "HR E-Application" :
  "Modified website running on ColdFusion 8.0 with functionalities that included
accessing to MS SQL 2008 and email services."},
{ "Atlassian Jira and Confluence" :
  "Used Groovy to export tickets from TrackIT! into Jira and MDR structure into
Confluence."},
{ "PLM Report" :
  "This is a lifecycle management report system. Worked in collaboration to build the
user interface using JavaScript, jQuery, and CFML in the back end. This website was
running on ColdFusion 8.0 server."},
{ "Microsoft SharePoint" :
  "Export documentation from Atlassian Confluence into Microsoft SharePoint."}

}}
},

```

```

{ "Hourglass Applications - Vestal, New York (May 2011 - March 2012)":
  { "<p><i>Programmer and Developer</i></p><p>Involved in an application development for
Android phones and Nook tablets using Java and Eclipse, in partnership with David W.
Corrigan, MS (Phone: 607.239.2702).</p><p><u>Project Examples:</u></p>" : [
    { "IBM WebSphere Cast Iron" :
      "Designed, coded, tested, documented and maintained integrations which synchronize
information between different databases such as AS400, MS SQL Servers, and cloud
solutions such as Oracle CRM on Demand and Big Machines using SOAP and REST web
services."},
    { "Suppliers Center" :
      "Maintained and expanded website running on Apache Tomcat 5.5 using JavaScript,
jQuery, and JSP with functionalities that included accessing multiple databases (MS SQL
Server, AS400, and MS Access 97), reporting, plus providing email services."},
    { "Asaichi Strips" :
      "Designed, coded, tested, documented and maintained website running on ColdFusion
8.0 and 9.0.2 accessing databases (MS SQL Server and AS400) and providing export
services (CSV, XLS, and PDF). Agile programming approach similar to SCRUM was used when
building this system."},
    { "Contractor Manager System" :
      "Redesigned RMIS for Human Resources department, coded using PrimeFaces framework,
tested, documented and maintained website running on Apache Tomcat 7.0/8.0 with access
to MS SQL 2012 Server database."},
    { "HR E-Application" :
      "Modified website running on ColdFusion 8.0 with functionalities that included
accessing to MS SQL 2008 and email services."},
    { "Atlassian Jira and Confluence" :
      "Used Groovy to export tickets from TrackIT! into Jira and MDR structure into
Confluence."},

```

```

{ "PLM Report" :
  "This is a lifecycle management report system. Worked in collaboration to build the
  user interface using JavaScript, jQuery, and CFML in the back end. This website was
  running on ColdFusion 8.0 server."},
{ "Microsoft SharePoint" :
  "Export documentation from Atlassian Confluence into Microsoft SharePoint."}

]}
},

```

```

{ "Broome Community College - Binghamton, New York (January 2007 - December 2008)":
  { "<p><i>Computer Lab Student Support</i></p><p>Maintained a positive learning
  environment in the computer laboratory and provided instructional support service to
  credit students. Provide support to students on topics such as Java, Visual Basic,
  Visual C# .NET, Visual C++ .NET, Structured Programming, Object Oriented Programming,
  Microprocessors and Assembly Language Programming, Data Structures, Problem Solving and
  Communication Tools, and Digital Logic.</p>" : []}
},

```

```

{ "Fusion Development, Inc. - Coral Springs, Florida (December 2004 - December 2006)":
  { "<p><i>Project Manager in Charge of Applications Development</i></p><p>Maintained a
  positive learning environment in the computer laboratory and provided instructional
  support service to credit students. Provide support to students on topics such as
  Visual Basic, Visual C# .NET, Visual C++ .NET, Structured Programming, Object Oriented
  Programming, Microprocessors and Assembly Language Programming, Data Structures,
  Problem Solving and Communication Tools, and Digital Logic.</p><p><u>Project Examples:
  </u></p>" : [
    { "13angel" :
      "Acted as project leader by designing, testing, and documenting a 3D visualization
      tarot software built on Visual C++ and Direct 3D. Also, designed, coded, tested and
      documented CD recording system for software distribution using JavaScript, PHP, and
      Linux command line for CD-ROM recording."},
    { "Eurocoach, Inc." :
      "Eurocoach is a company that restores and fabricates Ferraris, racing and Italian
      cars. This website was built using Macromedia Flash, Macromedia Fireworks, JavaScript
      and PHP."},
    { "Tic-Tac-Toe 3D for BluePepper Technologies, Inc." :
      "Acted as project leader by designing, testing and documenting 3D tic-tac-toe for
      BlackBerry market built using C."},
    { "Fusion Development, Inc." :
      "Website build using Macromedia Flash, Macromedia Fireworks, JavaScript, PHP and
      MySQL."},
    { "Tango Advertising Banners" :
      "Designed advertising banners using Macromedia Fireworks for Tango Group in Miami,
      Florida."}
  ]}
},

```

```

{ "BluePepper Technologies, Inc. - Coral Springs, Florida (January 2003 - December
2004)":
  { "<p><i>Software Designer</i></p><p>Developed, designed, and wrote PHP programs that
  were instrumental in the success of four projects related with telephone and wireless

```

```

technology; also, maintenance of the company website. Performed installation,
maintenance, configuration, and repair PC Networks. </p><p><u>Project Examples:</u>
</p>" : [
  { "MTeam" :
    "Instant conferencing for business subscribers. URL: <a
href=\"http://bptweb.com/mteam/index.html\"
target=\"_blank\">http://bptweb.com/mteam/index.html</a>\"},
    { "Ping" :
      "Instant conferencing for young adults and teens. URL: <a
href=\"http://bptweb.com/ping/index.html\"
target=\"_blank\">http://bptweb.com/ping/index.html</a>\"},
    { "SvMS" :
      "Short voice message services. URL: <a href=\"http://bptweb.com/svms/index.html\"
target=\"_blank\">http://bptweb.com/svms/index.html</a>\"},
    { "PASS" :
      "Patient automatic scheduling services."},
    { "BluePepper Technologies, Inc" :
      "Maintained website using JavaScript, PHP, and MySQL.<br>All projects interfaces
were built using JavaScript, PHP, MySQL, Macromedia Flash, and Macromedia Fireworks."}
  ]}
},

{ "Apex Voice Communications - Coral Springs, Florida (October 2001 - December 2004)":
  { "<p><i>Web Programmer and Designer</i></p><p>Started in the company creating,
developing and maintaining the APEX Reseller Program website using ASP, JavaScript,
Macromedia Flash and Macromedia Fireworks with functionalities that included database
access (ODBC to MS SQL and MS Access), email services, callback, and others. Performed
development of telephone applications and installation; maintenance, configuration and
repair of PC Networks.</p>" : []}
},

{ "Ecconet S.A. - Capital Federal, Buenos Aires, Argentina (March 2000 - June 2001)":
  { "<p><i>Technical Support</i></p><p>Maintenance, configuration, and repair of PC
Networks.</p>" : []}
}

]

```

Please notice that in your database the information would not be organized in this way. I am doing in purposed because I don't wish to give a solid idea on how the database looks like:

Second, lets go to our template, accord.html:

```

<uib-accordion close-others="true">

  <div ng-repeat="companies in datasource" ng-init="companiesIndex = $index">

    <div ng-repeat="(a, b) in companies">

      <uib-accordion-group is-open="status.openMain[companiesIndex]" >

        <uib-accordion-heading>
          {{'\{:a\}\'}} <!-- Company Name -->
          <i class="pull-right glyphicon"
            ng-class="{ 'glyphicon-chevron-down': status.openMain[companiesIndex],
'glyphicon-chevron-right': !status.openMain[companiesIndex]}">
            </i>
        </uib-accordion-heading>

        <div ng-repeat="(c, d) in b">
          <p ng-bind-html="c"></p> <!-- Company Description -->

          <uib-accordion close-others="false">

            <div ng-repeat="(e, f) in d">

              <div ng-repeat="(g,h) in f">

                <uib-accordion-group is-open="status.openSub[e]" >

                  <uib-accordion-heading>
                    {{'\{:g\}\'}} <!-- Project Name -->
                    <i class="pull-right glyphicon"
                      ng-class="{ 'glyphicon-chevron-down': status.openSub[e], 'glyphicon-
chevron-right': !status.openSub[e]}"></i>
                  </uib-accordion-heading>
                  <p ng-bind-html="h"></p> <!-- Project Description -->

                </uib-accordion-group>

              </div>
            </div>

          </uib-accordion>
        </div>

      </uib-accordion-group>
    </div>

  </uib-accordion>
</div>

```

In accord, we are using “ng-bind-html” which allows us to include HTML in our JSON to be displayed.

You need to have the angular-sanitize.js library in order to use it. Also, notice I am using one-time binding “::” indicator, in this way we reduce the number for \$watchers being used.

Next, is the motor of our directive and controller, index.js:

```
(function () {
  'use strict';

  var accMod = angular.module('acc', ['ngAnimate', 'ui.bootstrap']);

  accMod.directive('accord', function () {

    var scope = {
      datasource: '='
    };

    return {
      templateUrl: 'accord.html',
      scope: scope
    };

  });

  var mod = angular.module('ui.bootstrap.app', ['acc', 'ngSanitize']);

  mod.controller('workExperienceController', ['$scope', '$http', function($scope, $http)
  {

    $http.get('workExperience.json')
      .then(function(res){
        $scope.workExperienceDataSource = res.data;
      });

  }]);

}());
```

Short and to the point. I created a module where I administrate the directive related with accordion UI feature, and another module which takes care of all the modules of my application. By doing this, I can reuse code that later I inject in other projects. Also, I can little by little build my own angular library.

Here is the main page, index.html:

```

<!doctype html>
<html ng-app="ui.bootstrap.app">
  <head>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular.js">
</script>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular-
animate.js"></script>
    <script src="https://code.angularjs.org/1.2.0-rc.3/angular-sanitize.js"></script>
    <script src="http://angular-ui.github.io/bootstrap/ui-bootstrap-tpls-0.14.3.js">
</script>
    <script src="index.js"></script>
    <link href="http://netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css"
rel="stylesheet">
  </head>
  <body>

    <div>

      <p><b>WORK EXPERIENCE</b></p>
      <div ng-controller="workExperienceController">
        <div accord datasource="workExperienceDataSource"></div>
      </div>

    </div>

  </body>
</html>

```

In the future, I will create a directive that will iterate through all the JSON and build nested accordions automatically.

For now, this JSON will create a 2 level nested accordion as displayed below.

Finally, the working example:

© 2015, Alejandro G. Carlstein Ramos Mejia. All rights reserved.