

AngularJS – Part 1

Agenda

- Introduction to AngularJS
- Adding reference of AngularJS
- Creating a very simple example

Tools

-

Pre-requisite

-

Version	Last updated	Comments	Modified By
V1.0	23-05-2016		Bilal Shahzad

Brief Introduction

AngularJS is a JavaScript Framework for client side development.

It gives behavior to our (static) HTML. It reduces your code and makes development fast.

It introduces different attributes and new tags to give behavior. We call them “directives”.

AngularJS is a MVVM (Model View-View Model) framework. To make things easier, you may consider it like MVC framework.

A module in angular is like a component which handles one or more pages functionality. You will have to create at least one “module” before starting your work. You use “angular.module()” function to create a module.

Once a module is created, you can use it to create controller. Controller is a function which manages data (ViewModel) and communication with services. You use “ng-controller” directive to mention which part of the page is going to use which controller.

A Service is a code which will contain your business logic (or any reusable logic).

When you apply “ng-app” directive on some element, “angular” start parsing that element and all child elements inside it. You provide some module name to “ng-app”.

When it finds “ng-controller” during parsing, it creates a context for that specific portion.

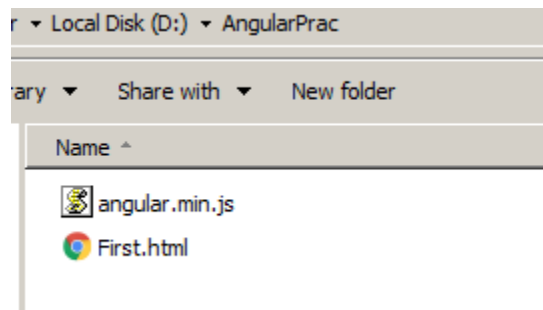
AngularJS supports Two-Way data binding. It means if you make any change in data in your HTML (e.g. changing a value in textbox, selecting some item in dropdown), it will automatically populate value in attached variable/objects in controller (\$scope object). And if you make any change in variable/objects in \$scope, it will automatically update DOM to reflect change in attached elements.

Step by Step Walkthrough

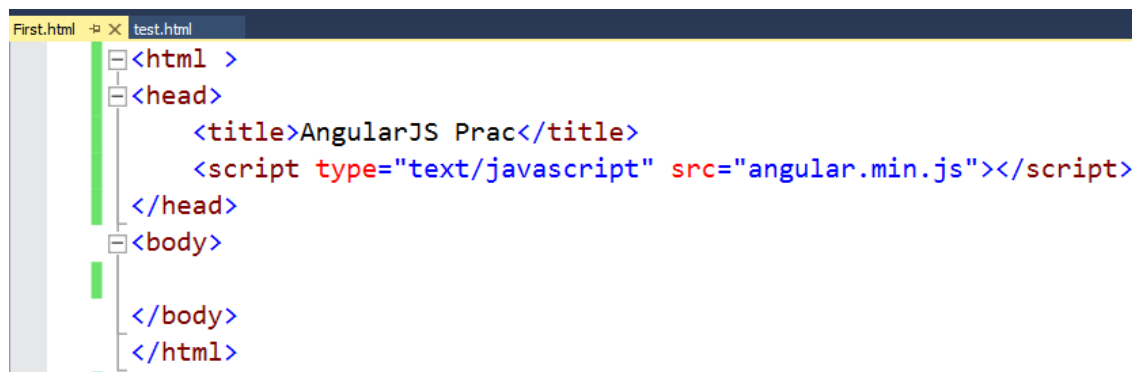
- 1- First, you will have to download AngularJS file. To do this, go to <https://angularjs.org/>
- 2- Click on “Download AngularJS 1” to download required JS file.



- 3- Create a folder somewhere with some name (let say D:\AngularPrac) and copy “angular.min.js” file in that folder. Also create a new html file (e.g. First.html)



- 4- Open “First.html” in any editor (e.g. Visual Studio, Notepad++) and add following HTML. Here we are just adding reference of our “angular.min.js” file which is in same folder.

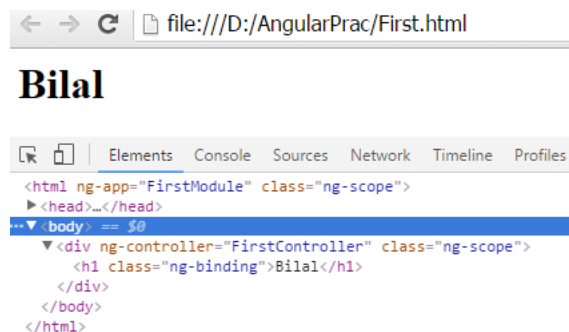


- 5- Now make following changes in “First.html” file. “angular” is a global object which comes when we add reference of angular library.
- In Line 2, we are telling the module we are going to use in current page by using “ng-app” directive.
 - In Lin 7, we are creating our module “FirstModule” by using “module” function. Second parameter is the list of modules on which “FirstModule” depends on. When you are creating a module, you need to pass second parameter even if there is no dependency for now. When you don’t provide second parameter, it means you are trying to get already created module. Once module is created, its reference will be stored in “app” variable.
 - In Line 8, we are creating a controller named “FirstController” which is referring to a function. “\$scope” is the parameter which will be used to pass different type of between controller & html (view).
 - In Line 9, we’ve added a property “name” in our “\$scope” variable.
 - In Line 14, we are using “ng-controller” to specify a context for our div. It means any property which we’ve defined in that specific controller (using \$scope) can be accessed here.
 - In Line 15, we’ve used “Binding Expression” to show the value of “name” property. When parser will read this expression, it will look for “name” property inside “\$scope” object of “FirstController”.



```
1
2 <html ng-app="FirstModule">
3 <head>
4   <title>AngularJS Prac</title>
5   <script type="text/javascript" src="angular.min.js"></script>
6   <script>
7     var app = angular.module("FirstModule", []);
8     app.controller("FirstController", function ($scope) {
9       $scope.name = "Bilal";
10    })
11  </script>
12 </head>
13 <body>
14   <div ng-controller="FirstController">
15     <h1>{{name}}</h1>
16   </div>
17 </body>
18 </html>
```

- 6- Double click on “First.html” page to run it in browser. Check if you see something like this or not.



7- Now make following changes in “First.html”.

- “ng-model” is used for form controls. You use the name of variable which you want to attach with that form control. When you will make any change in control, variable will be updated automatically. And when you will make change in variable (in \$scope), control state will be changed automatically.
- “ng-repeat” is used to repeat specific element of an array. We face many situations when we want to repeat same element multiple times but with different data. For example, we want to load cities in a drop down. Now a drop down contains “option” element where value and text will be different for each city.
- So when we’ll make any change in textbox, value of “name” variable in controller will be changed as both are attached. And on all places where “name” is used in “view” will be updated (for example inside h1 tag Line 20).
- We’ve used “ng-repeat” on <option> tag. So this option tag will be generated for all values of cities. We’ve also used “ng-model” with <select> tag. It means whenever you will select an item in dropdown, corresponding “value” of that item will be loaded in “selectedItem” variable in \$scope of controller.

```

1  <script>
2
3  var app = angular.module("FirstModule", []);
4  app.controller("FirstController", function ($scope) {
5      $scope.name = "Bilal";
6      $scope.cities = [
7          { cid: 1, name: 'Lahore' },
8          { cid: 2, name: 'Karachi' },
9          { cid: 3, name: 'ISB' }];
10     $scope.selectedItem = "";
11 })
12 </script>
13 </head>
14 <body>
15     <div ng-controller="FirstController">
16         <input type="text" ng-model="name" /><h1>{{name}}</h1>
17         <br />
18         <select ng-model="selectedItem">
19             <option ng-repeat="c in cities" value="{{c.cid}}">{{c.name}}</option>
20         </select>
21         <span>{{selectedItem}}</span>
22     </div>
23 </body>
24 </html>

```

Here we are creating an array of objects (cid,name)

A 'variable' which will be bound with some element in DOM

8- Run the file and see if it is working as expected. Try to change text in textbox. Try to select some city from dropdown. Inspect the HTML of the page.

Tasks for Practice

Once you are done with above tutorial. Try to work on these tasks.

- 1- Create another property in “\$scope” object and show it on the page using binding expression “{{}}” inside the DIV which has ng-controller attribute. Also use same binding expression outside the DIV. Run page and see if it works or not.
- 2- Add a new property “studnets” of type Array in \$scope (just like cities). This property will contain objects. Each object will contain (ID,Name,Age). Assign some dummy objects (just like we did with cities).
- 3- Now write code in your view side to iterate “students” array using “ng-repeat” and show students in a table. Hint: You need to apply “ng-repeat” on <tr> tag as each student will be represented in a row.
- 4- In same code, create another controller with name “SecondController”. In this controller function, add some property to “\$scope” object. Then create another DIV in your page and refer this controller using “ng-controller” directive on this new DIV. Note: New DIV should be outside of existing DIV. Show the property added in “SecondController” \$scope object inside DIV using binding expression.

Useful Links

<https://angularjs.org/>

<http://www.w3schools.com/angular/>

<http://www.tutorialspoint.com/angularjs/>