FULL STACK DEVELOPMENT USING OPEN-SOURCE TECHNOLOGY

By: Mrs. Nidhi Divecha (ME CMPN)

(UNIT 2)

ANGULAR JS

INTRODUCTION TO ANGULARJS

- Angular JS is an open-source JavaScript framework that is used to build single page web applications.
- It extends HTML with additional attributes and makes it more responsive to user actions.
- AngularJS is open source, completely free, and used by thousands of developers around the world.
- Angular Js is developed by Google.

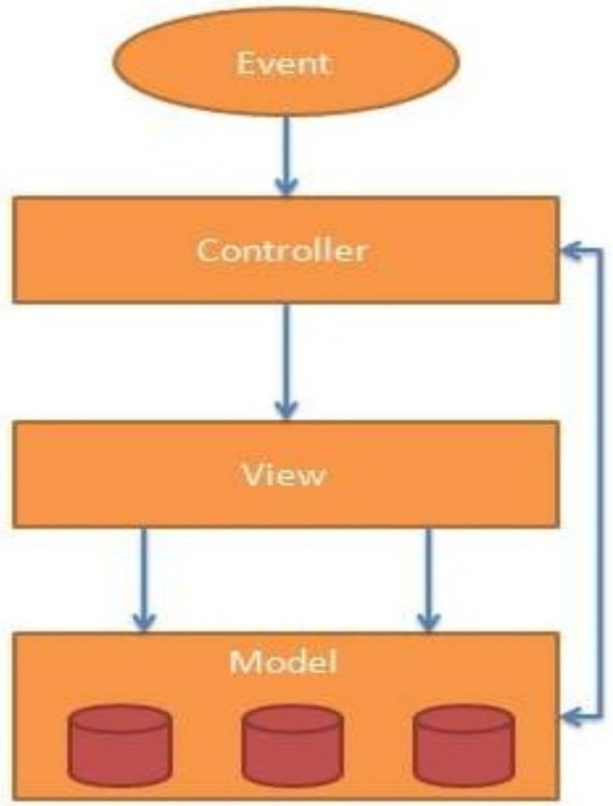
ANGULARJS BENEFITS

+ **Following are the advantages of AngularJS over other JavaScript frameworks:**

- **Dependency Injection:** Dependency Injection specifies a design pattern in which components are given their dependencies instead of hard coding them within the component.
- **Two-way data binding:** AngularJS creates a two-way data-binding between the select element and the orderProp model. orderProp is then used as the input for the orderBy filter.
- **Testing:** Angular JS is designed in a way that we can test right from the start. So, it is very easy to test any of its components through unit testing and end-to-end testing.
- **Model View Controller:** In Angular JS, it is very easy to develop application in a clean MVC way. You just have to split your application code into MVC components i.e. Model, View and the Controller.
- Directives, filters, modules, routes etc.

AngularJS MVC Architecture

MVC stands for Model View Controller. It is a software design pattern for developing web applications.



+ The MVC pattern is made up of the following three parts:

1. **Model:** It is responsible for managing application data. It responds to the requests from view and to the instructions from controller to update itself.
2. **View:** It is responsible for displaying all data or only a portion of data to the users. It also specifies the data in a particular format triggered by the controller's decision to present the data.
3. **Controller:** It is responsible to control the relation between models and views. It responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

AngularJS First Example

```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/an
gularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="">
  <p>Name : <input type="text" ng-model="name"></p>
  <h1>Hello {{name}}</h1>
</div>

</body>
</html>
```

```
Input something in the input box:

Name :
nidhi

Hello nidhi
```

- AngularJS applications are a mix of HTML and JavaScript. The first thing you need is an HTML page.

```
<!DOCTYPE html>
<html>
<head>
.
.
</head>
<body>
.
.
</body>
</html>
```

- Second, you need to include the AngularJS JavaScript file in the HTML page so we can use AngularJS

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular.min.js"></script>
</head>
<body>
.
.
</body>
</html>
```

AngularJS First Example

Following is a simple "Hello Word" example made with AngularJS. It specifies the Model, View, Controller part of an AngularJS app.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular.min.js"></script>
</head>
<body ng-app="myapp">
<div ng-controller="HelloController" >
<h2>Hello {{helloTo.title}} !</h2>
</div>

<script>
angular.module("myapp", [])
   .controller("HelloController", function($scope) {
       $scope.helloTo = {};
       $scope.helloTo.title = "World, AngularJS";
   } );
</script>
</body>
</html>
```

View Part

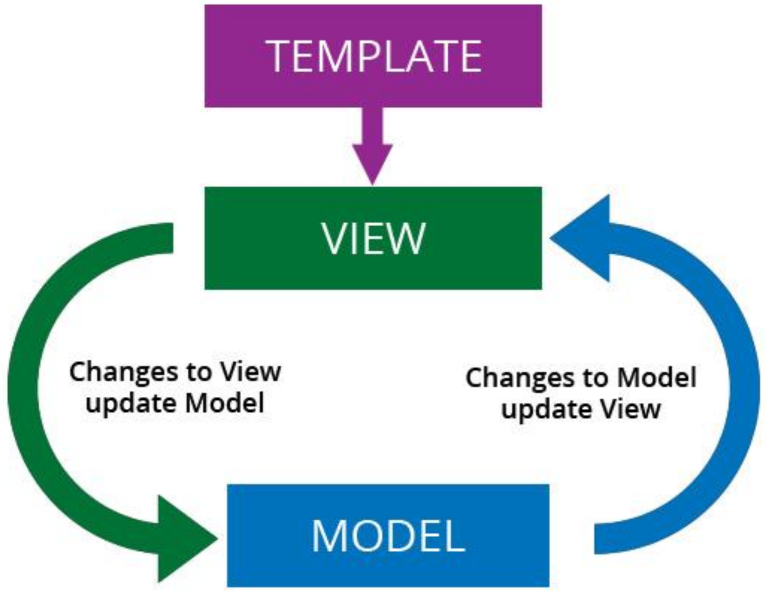
```
<div ng-controller="HelloController" >
<h2>Hello {{helloTo.title}} !</h2>
</div>
```

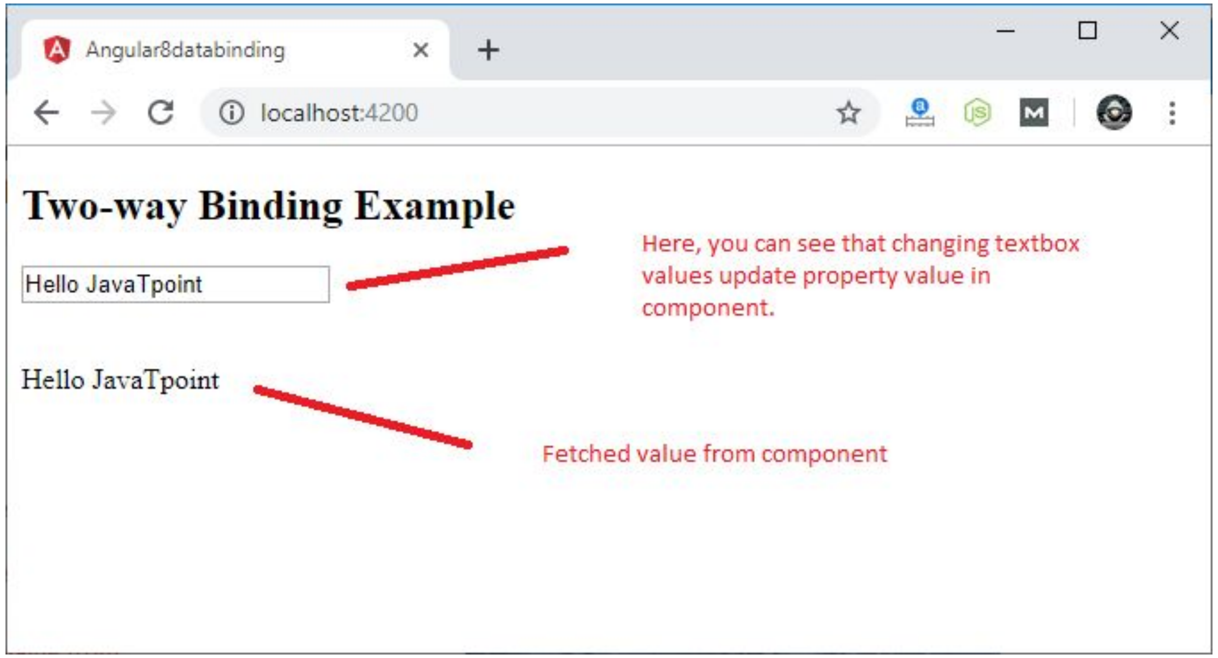
Controller Part

```
<script>
angular.module("myapp", [])
    .controller("HelloController", function($scope) {
        $scope.helloTo = {};
        $scope.helloTo.title = "World, AngularJS";
    });
</script>
```

AngularJS Data Binding

- It acts as a bridge between the view and business logic of the application.
- AngularJS follows Two-Way data binding model.
- Data binding in AngularJS is the synchronization between the model and the view.





Two-way Binding Example

Hello JavaTpoint

Here, you can see that changing textbox values update property value in component.

Hello JavaTpoint

Fetched value from component

You can check it by changing textbox value and it will be updated in component as well.

AngularJS Directives

- AngularJS lets you extend HTML with new attributes called Directives.
- AngularJS has a set of built-in directives which offers functionality to your applications.
- AngularJS also lets you define your own directives.
- AngularJS directives are extended HTML attributes with the **prefix ng-**
- **ng-app** – This directive starts an AngularJS Application.
- **ng-init** – This directive initializes application data.
- **ng-model** – This directive defines the model that is variable to be used in AngularJS.
- **ng-repeat** – This directive repeats HTML elements for each item in a collection.

Data Binding

- Example
-
```
<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>

</div>
```

- The {{ firstName }} expression, in the example above, is an AngularJS data binding expression.
- {{ firstName }} is bound with ng-model="firstName".

Data Binding

+ In the next example two text fields are bound together with two ng-model directives:

```
<div ng-app="" ng-init="quantity=1;price=5">

Quantity: <input type="number" ng-model="quanti
ty">
Costs:    <input type="number" ng-model="price"
>

Total in dollar: {{ quantity * price }}

</div>
```

ng-app directive

+ The ng-app directive starts an AngularJS Application.
+ It defines the root element.
+ It automatically initializes or bootstraps the application when the web page containing AngularJS Application is loaded.
+ It is also used to load various AngularJS modules in AngularJS Application.
+ In the following example, we define a default AngularJS application using ng-app attribute of a <div> element.

```
<div ng-app = "">
   ...
</div>
```

ng-init directive

- The ng-init directive initializes an AngularJS Application data. It is used to assign values to the variables.
- In the following example, we initialize an array of countries. We use JSON syntax to define the array of countries.
- <div ng-app = "" ng-init = "countries = [{locale:'en-US',name:'United States'},
-    {locale:'en-GB',name:'United Kingdom'}, {locale:'en-FR',name:'France'}]">
-    ...
- </div>

ng-model directive

- The ng-model directive defines the model/variable to be used in AngularJS Application. In the following example, we define a model named *name*.
- <div ng-app = "">
-    ...
-    <p>Enter your Name: <input type = "text" ng-model = "name"></p>
- </div>

ng-repeat directive

- The ng-repeat directive used on an array of objects:
- Example:

-
  ```
  <div ng-app="" ng-init="names=['Jani','Hege','Ka
  i']">
    <ul>
      <li ng-repeat="x in names">
        {{ x }}
      </li>
    </ul>
  </div>
  ```

AngularJS Controllers

- AngularJS controllers are used to control the flow of data of AngularJS application.
- A controller is defined using ng-controller directive.
- A controller is a JavaScript object containing attributes/properties and functions.
- Each controller accepts $scope as a parameter which refers to the application/module that controller is to control.

AngularJS Controller Example

```
<!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="myCtrl">
First Name: <input type="text" ng-model="firstName"><br>
Last Name: <input type="text" ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
</script>
</body>
</html>
```

+ The AngularJS application is defined by ng-app="myApp". The application runs inside the <div>.
+ The ng-controller="myCtrl" attribute is an AngularJS directive. It defines a controller.
+ The myCtrl function is a JavaScript function.
+ AngularJS will invoke the controller with a $scope object.
+ In AngularJS, $scope is the application object (the owner of application variables and functions).
+ The controller creates two properties (variables) in the scope (firstName and lastName).
+ The ng-model directives bind the input fields to the controller properties (firstName and lastName).

Module in AngularJS

- A module in AngularJS is a container of the different parts of an application such as controller, service, filters, directives, factories etc.
- You can think of a module as a Main () method in other types of applications.

Example: Create Application Module

```
<!DOCTYPE html>
<html >
<head>
    <script src="~/Scripts/angular.js"></script>
</head>
<body ng-app="myApp">

<script>
        var myApp = angular.module('myApp', []);

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

- In the above example, the angular.module() method creates an application module, where the first parameter is a module name which is same as specified by ng-app directive.
- The second parameter is an array of other dependent modules [].
- In the above example we are passing an empty array because there is no dependency.

The following example demonstrates creating controller module in myApp module.

```
<!DOCTYPE html>
<html >
<head>
    <script src="~/Scripts/angular.js"></script>
</head>
<body ng-app="myApp">
    <div ng-controller="myController">
        {{message}}
    </div>
    <script>
        var myApp = angular.module("myApp", []);

        myApp.controller("myController", function ($scope) {
            $scope.message = "Hello Angular World!";
        });
    </script>
</body>
</html>
```