

The AngularJS Components

There are three main component of AngularJS are following below -

ng-app – This directive defines AngularJS based web application to HTML.

ng-model – This directive binds the values of AngularJS application data to HTML input controls.

ng-bind – This directive binds the AngularJS Application data to HTML tags.

AngularJS Directives

AngularJS directives are basically used to extend the HTML attributes called Directives. They starts with prefix ng-. We will discuss following directives-

The Ng-App Directive

The ng-app directive is used to initializes an AngularJS application. ng-app is the root element of the Angular application. It is automatically initialize when the application is loaded.

Syntax : <div ng-app = ""> ... </div>

The ng-init Directive : The ng-init directive is used to initializes application data or value or you can say put the value to the variable.

Syntax : <div ng-app="" ng-init="Name='Juhi'"></div>

Example

```
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
</script>
<body>
<div ng-app="" ng-init="Name='Juhi'">
<p>Type your last Name :</p>
<p>Name: <input type="text" ng-model="Name"></p>
<p>Hello {{ Name }}</p>
</div>
</body> </html>
```

Ng-Model Directive

The ng-model directive is used binds the value of HTML controls (like input) to application data.

Syntax : <div ng-app = ""> ... <p>Type Name: <input type = "text" ng-model = "Name"></p> </div>

The ng-repeat Directive

The ng-repeat directive is used to iterate the html elements for each item in a collection. In following example, we've iterated over array of cities.

Syntax : <div ng-app="" ng-init="cities=['Delhi','Noida','Gurgaon']">

```
<ul> <li ng-repeat="city in cities">
{{ city }}
</li> </ul> </div>
```

ng-repeat Example

```
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
</script>
<body>
<div ng-app="" ng-init="cities=['Delhi','Noida','Gurgaon']">
<ul>
<li ng-repeat="city in cities">
{{ city }}
</li> </ul> </div> </body> </html>
```

AngularJS – Expressions

Angular Expression are used to bind the directive data to HTML. Angular expression are putting inside the double braces.

For Example

```
<input type="text" name = "name" ng-model="name">
{{name}}
```

Expressions are used to bind application data to html. Expressions are written inside double braces like {{expression}}. Expressions behaves in same way as ng-bind directives. You can use express in the

place of ng-bind. AngularJS application expressions are like JS expressions and put the data where they are used.

AngularJS Example

```
<!DOCTYPE html>
<html>
<script src="angular.min.js"></script>
<body>
<div ng-app="">
<p>Simple Angular Expression: {{ 38 + 2 }}</p>
</div> </body> </html>
```

AngularJS Example

```
<!DOCTYPE html>
<html> <script src="angular.min.js"></script>
<body>
<div ng-app="" ng-init="a=10; b=20;">
<p>Simple Angular Expression: {{ a*b }}</p>
</div> </body> </html>
```

AngularJS Example

```
<!DOCTYPE html>
<html> <script src="angular.min.js"></script>
<body>
<div ng-app="" ng-init="a=10; b=20;">
<p>Simple Angular Expression: {{ a*b }}</p>
</div> </body> </html>
```

AngularJS Sum Example

```
<!DOCTYPE html>
<html> <script src="angular.min.js"></script>
<body>
<div ng-app="">
<h1>Calculate sum</h1>
<input type="number" ng-model="a">
<input type="number" ng-model="b">
<p>Simple Angular Sum: {{a+b}}</p>
</div> </body> </html>
```

AngularJS Objects

AngularJS objects are like JavaScript objects so you can easily access via dot(.) operator

AngularJS Objects Example

```
<!DOCTYPE html>
<html> <script src="angular.min.js"></script>
<body>
<div ng-app="" ng-init="student={Name:'Brown',RollNo:38}">
<p>The name is {{ student.Name }}</p>
<p>The Roll Number is {{ student.RollNo }}</p>
</div> </body> </html>
```

AngularJS Arrays

AngularJS arrays are like JavaScript arrays so you can easily access via index.

AngularJS Arrays Example

```
<!DOCTYPE html>
<html>
<script src="angular.min.js"></script>
<body>
<div ng-app="" ng-init="names=['uk','kk','khan','umar']">
<p>The third result is {{ names[2] }}</p>
</div>
</body> </html>
```

Note: angular.min.js file must be in the same folder or use this url <http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js> but in this case internet must be connected.

AngularJS Controller

The AngularJS Controller basically JavaScript constructor function that is used to pass the AngularJS Scope. In other word controller is used to control the data in the whole application.

AngularJS Controller

This example explain the concept of the angular controller with point to point explanation.

AngularJS Controller Example

```
<html> <head>
<title>Angular JS Controller Example</title>
<script src = "angular.min.js"></script>
</head> <body>
<h2>AngularJS Sample Controller Application</h2>
<div ng-app = "ukApp" ng-controller = "ukController">
Enter first name: <input type = "text" ng-model = "name.FName"><br><br>
Enter last name: <input type = "text" ng-model = "name.LName"><br>
<br> Your Name: {{name.fullN()}} </div> <script>
var mainApp = angular.module("ukApp", []);
mainApp.controller('ukController', function($scope) {
$scope.name = {
FName: "Umar",
LName: "Farooque",
fullN: function() {
var studentObject;
studentObject = $scope.name;
return studentObject.FName + " " + studentObject.LName;
} }; });
</script>
</body> </html>
```

Explanation of Controller Example

The angular Application always starts from directive ng-app so ng-app="ukApp" is the starts point and end where the div is end. The ng-controller="ukController" attribute is an AngularJS directive that is used for defining the controller. The ukController function is a JavaScript function with \$scope, this is dependency injection that's built into AngularJS. AngularJS will invoke the controller with a \$scope object. \$scope.name is property of studentController object. FName and LName are two properties of scope.name object. They have some default value. fullN is the function of \$scope.name object which return the full name including first name and last name.

Note: We can also defined the controller object in separate JavaScript and use that particular file in HTML.

AngularJS Ajax

AngularJS provides the \$http service that is basically used to communicate with the remote HTTP servers via the browser's XMLHttpRequest object or via JSONP. In other word \$http service is used to send the ajax call to remote server. Ajax call must be in the same server.

Important Point About \$Http

AngularJS \$http is a Angular service for reading data from servers. \$http.get(url) is a function that is used to read the data from server.

Example

```
<!DOCTYPE html>
<html>
<script src= "angular.min.js"></script>
<body>
<div ng-app="ukApp" ng-controller="ukController">
{{data}}
</div>
<script>
```

```

var app = angular.module('ukApp', []);
app.controller('ukController', function($scope, $http) {

$http.get("http://www.ptutorial.com/angularjs/data3.php")
.success(function(response) {$scope.data = response;});
});
</script> </body> </html>

```

Explanation

You can simply execute this example by deploying the angularjshttp.html and the data3.php file on the server. AngularJS will invoke ukController with a \$scope and \$http object. \$scope is the application object. \$http is an XMLHttpRequest object for requesting external server data. \$http.get() method is used to get the data from server (). If success, the controller assign the response data to the data property. [Http://www.ptutorial.com/angularjs/data3.php](http://www.ptutorial.com/angularjs/data3.php) File Data

AngularJS Filter

As the name filter can be used to transform the data. For example formatting a string to the lowercase. Filter can be used through the directives or expression by using the pipe (|) operator. Here Is Some Common Filter Used In AngularJS

Filters - Filter Description

Currency - Change number to currency format

Filter - Select a subset of items from an array

lowercase - Change a string to lower case

orderBy - Orders an array by an expression

uppercase - Change a string to upper case

Adding Filters To Expressions

You can add filter to the expression by using pipe operator (|) forward by filter

Uppercase Filter Example

```

<div ng-app="ukApp" ng-controller="ukController">
<p>The name is {{ name | uppercase }}</p>
</div>

```

Lowercase Filter Example

```

<div ng-app="ukApp" ng-controller="ukController">
<p>The name is {{ name | lowercase }}</p>
</div>

```

Currency Filter Example

```

<div ng-app="ukApp" ng-controller="costCtrl">
<input type="number" ng-model="rs">
<input type="number" ng-model="qua">
<p>Total = {{ (rs * qua) | currency }}</p>

</div>

```

OrderBy Filter

To order subjects by marks, we've used orderBy marks.

OrderBy Filter Example

```

<ul>
<li ng-repeat = "subject in student.subjects | orderBy:'marks'">
{{ subject.name + ', marks:' + subject.marks }}
</li> </ul>

```

AngularJS Forms

AngularJS provides the feature to bind the HTML form data or input fields to the model object. You can use these feature to bind the form data to model. Bind text field using ng-model

AngularJS Text Field

You can simply bind text field using ng-model as shown following below-

```

<!DOCTYPE html>
<html lang="en-US">
<script src="angular.min.js"></script>
<body>

```

```

<div ng-app="">
<p>Enetr Name : <input type="text" ng-model="name"></p>
<p style="color:red";>Hello {{name}}</h2></p></p>
</div> </body> </html>

```

AngularJS Binding Radio Buttons

You can easily bind radio buttons just like text boxes. If you are using group of radio button use same ng-model name.

```

<!DOCTYPE html>
<html lang="en-US">
<script src="angular.min.js"></script>
<body>
<div ng-app="">
<p>Select gender</p>
<form>
Male: <input type="radio" ng-model="myForm.gen" value="Male"><br/>
Female: <input type="radio" ng-model="myForm.gen" value="Female">
</form>
You are
<p style="color:blue";> {{myForm.gen}}</h2></p></p>
</div> </body> </html>

```

Binding Select Boxes

```

<html> <head>
<script src= "angular.min.js"></script>
</head>
<body ng-app="app">
<div ng-controller="Test">
Select Name: <select ng-model="name">
<option ng-repeat="item in items" value="{{item}}">{{item}}</option>
</select>
<p>Selected Name is : {{name}}</p>
</div>
<script>
var app = angular.module('app',[]);
app.controller('Test',function($scope){
$scope.items = ['umar','khan','farooque','sonu','singh']
});
</script> </body> </html>

```