

# Advanced AngularJS Interview Questions with Answers

4 (1 votes)



Many developer has question like “[What is AngularJS?](#)”. In replay AngularJS is a MVC framework to design web applications. It helps to enrich HTML for a great extend. Using AngularJS you can build structural & [high performance based web applications](#). Directive is one of the most powerful [features in AngularJS](#). AngularJS comes with many in-built directives. Even you can write your own. AngularJS is powered by Google. Due this is a newly introduced technology, there are lot of vacancies for this skill. Are you among the one who is [Seeking a Job](#) in AngularJS. Look at below. Here I am sharing **most asked AngularJS top interview Questions** with Answers.

## What are the key features of AngularJS?

AngularJS is a latest technology introduced by Google. The key features of AngularJS are like Scope, Controller, View, Model, Directives, Validation, Filters, Data Binding, Services & Testable. AngularJS supports MVC & MVVM design pattern. Using AngularJS we can do massively Parallel Development. AngularJS Supports Single Page Applications design.

## Does Angular use the JQuery library?

Yes. AngularJS is an advanced language. It supports [JQuery](#) library.

## How to use ng-repeat in AngularJS?

Ng-repeat is a per-defined directive in AngularJS. Look at the Example below. Here I used ng-repeat to bind data from JavaScript. In sample data I have two fields name & designation. I am showing these records using ng-repeat in a HTML UL element.

```

<html ng-app="nameDesgApp">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.1/angular.min.js">
</script>
<script type="text/javascript">
var nameDesgApp = angular.module('nameDesgApp', []); nameDesgApp.controller('NameDesgCtrl',
function ($scope) {$scope.employees = [{ 'name': 'Swatio Rao', 'designation': 'Project
Manager' }, { 'name': 'Biswabhusan Panda', 'designation': 'Module Lead' }, { 'name': 'Kumar
Abhishek', 'designation': 'Senior Team Leader' }]; });
</script>
</head>
<body ng-controller="NameDesgCtrl">
<ul>
<li ng-repeat="emp in employees">{{emp.name}}
<p>{{emp.designation}}</p>
</li>
</ul>
</body>
</html>

```

## How to use Filter in AngularJS?

Filter is like Search in AngularJS. In the following code I have 3 records & a search box. If I am doing search filter keyword help to purify the results according to the Query.

```

<html ng-app="nameDesgApp">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.1/angular.min.js">
</script>
<script type="text/javascript">
var nameDesgApp = angular.module('nameDesgApp', []); nameDesgApp.controller('NameDesgCtrl',
function ($scope) {$scope.employees = [{ 'name': 'Swatio Rao', 'designation': 'Project
Manager' }, { 'name': 'Biswabhusan Panda', 'designation': 'Module Lead' }, { 'name': 'Kumar
Abhishek', 'designation': 'Senior Team Leader' }]; });
</script>
</head>
<body ng-controller="NameDesgCtrl">
Search: <input ng-model="query">
<ul class="phones">
<li ng-repeat="emp in employees | filter:query">{{emp.name}}
<p>
{{emp.designation}}</p>
</li>

```

```
</ul>  
</body>  
</html>
```

## In AngularJS how will you initialize a select box with options on page load?

Using AngularJS ng-init directive. Look at the example below.

```
<div ng-app="" ng-init="empName= 'Raghav'">  
<p>Name: <input type="text" ng-model="empName"></p>  
<p>You wrote: {{ empName }}</p>  
</div>
```

## What are the advantages of using AngularJS?

AngularJS is a latest Technology introduced by Google. It has several advantages in web development.

AngularJS supports MVC pattern. Using AngularJS we can do 2 ways data binding. AngularJS has pre-defined form validations. AngularJS supports both client & server communication. AngularJS supports Animations.

## How to use Controller in AngularJS?

Controller is a JavaScript constructor function. When Creating an application we need to provide initial value to the \$Scope object of a Controller. We can add property to \$Scope object. Before to create controller we have to create a module. Look at the example below how I created Controller in AngularJS.

```
var myApp = angular.module('myApp',[]);  
myApp.controller('myController', ['$scope', function($scope) {  
  $scope.welcome = 'Hello!';  
}]);
```

The property of a controller we can assign to the DOM object using ng-controller.

```
<div ng-controller="myController">  
  {{ welcome }}  
</div>
```

## What are the form Validations AngularJS provides?

Client Validations made easy using AngularJS. All input fields can have Required field, Minimum Length, Maximum Length, Matches a Pattern, Email, Number, URL & Custom Validations. Look at the Examples below.

### *Required Field Validation*

Using Required field validation we can prevent form submission with null value. I mean input fields need to be filled by the user. The syntax for required field validation is as follows.

```
<input type="text" required />
```

### *Minimum & Maximum field Length Validations*

To prevent the input field from less or excess number of characters we use Minimum & Maximum length validation. The AngularJS directive used for Minimum & Maximum length validations are ng-minlength & ng-maxlength. Look at the example below.

```
<input type="text" ng-minlength=5 />  
<input type="text" ng-maxlength=10 />
```

#### *Matches Pattern Validation*

AngularJS provides ng-pattern directive to validate regular expressions. Look at the example below.

```
<input type="text" ng-pattern="[a-zA-Z]" />
```

#### *Email Validation*

To valid an email id AngularJS provides ng-model directive. Using the following line of code we can validate an email id from any input field.

```
<input type="email" name="email" ng-model="user.email" />
```

#### *Number Validation*

To validate an input against Number we can use ng-model directive from AngularJS. Look at the example below.

```
<input type="number" name="personage" ng-model="user.age" />
```

#### *URL Validation*

To validate an input field for URL we can use the following syntax in AngularJS.

```
<input type="url" name="weblink" ng-model="user.facebook_url" />
```

## **How to create Directive using AngularJS?**

Directives are designed to use as markers with DOM elements. Markers can be attribute, element name, comment or CSS class. AngularJS provides many in-built directives. Few of them are ngModel, ngView or ngBind. The naming standard directive follows is camel-case. Much like controllers directives are registered on Module. To register a directive we can use module.directive. The time we create directive by default it restricted to attribute. In order to create directive which will triggered by element or class name we need to use restrict. The options are as follows.

- A – only matches attribute name
- E – only matches element name
- C – only matches class name

These restrictions can be combined as required.

AEC – it matches either attribute or element or class name

Let us look at the below example [how to create a directive using AngularJS](#).

```
myAngularApp = angular.module("myAngularApp", []);  
myAngularApp.directive('div', function() {
```

```

var directive = {};
directive.restrict = 'E'; /* using this we restrict this directive to elements */
directive.template = "My first directive: {{ EnterText }}";
return directive;
});

```

## What is 2 way data binding in AngularJS?

In UI Development to create a template we basically do one way data binding. In case of Angular it is a MVC framework. Internally using one way data binding we merge template & model to display in view. But in 2 way data binding there is a continuous communication in between model & view. Where template is just pointing to the view.

## What is the CDN link for AngularJS?

CDN is a permanent Global space where from we can access the various versions of library file to use directly in our web applications. For AngularJS you can refer the following CDN link.

## How to bind MySQL data to HTML5 using AngularJS?

In server end create a php file with MySQL query for the required data. Format it in the shape of JSON. Here I did this with the file name Customers\_JSON.php.

```

Customers_JSON.php{
  "Name" : "Rakesh Srivastab",
  "City" : "Lulea",
  "Country" : "Sweden"
}, {
  "Name" : "Biswabhusan Panda",
  "City" : "Mexico D.F.",
  "Country" : "Mexico"
}, {
  "Name" : "Kumar Abhishek",
  "City" : "Graz",
  "Country" : "Austria"
}, {
  "Name" : "Raveena Tondon",
  "City" : "Madrid",
  "Country" : "Spain"
}]

```

in App.js create a angular module & controller. Inside the controller use \$http.get method to fetch the data from JSON formatted php file. Look at the example below.

*App.js*

```
var myApp = angular.module('myApp', []);  
myApp.controller('Controller', function($scope,$http) {  
$http.get("Customers_JSON.php").success(function(response) {  
$scope.names = response;});});
```

Finally, In your index.html file use ng-repeat over UL element to bind the response data. Sharing the code below.

*Index.html*

```
<!DOCTYPE html>  
<html ng-app="myApp">  
<head>  
<title>How to bind MySql data using AngularJS?</title>  
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-rc.0/angular.min.js"></script>  
<script type="text/javascript" src="app.js"></script>  
</head>  
<body>  
<div ng-controller="Controller">  
<ul>  
<li ng-repeat="x in names">  
{{ x.Name + ', ' + x.Country }}  
</li>  
</ul>  
</div>  
</body>  
</html>
```

## How to use templateUrl in the Directive of AngularJS?

To implement one line of HTML template is ok for Directive. If we are going to develop a more functional control depending upon HTML for a better professional approach we need to use templateUrl. Look at the example below how I used a html page to display in side an AngularJS directive using templateUrl.

*index.html*

```
<!DOCTYPE html>  
<html ng-app="myApp">  
<head>  
<title>Using templateUrl in the Directive of AngularJS</title>  
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-rc.0/angular.min.js"></script>  
<script type="text/javascript" src="app.js"></script>  
</head>  
<body>
```

```
<div ng-controller="templateUrlDemo">
<div template-Url></div>
</div>
</body>
</html>
```

*app.jsApp-Template.htm*

```
<!DOCTYPE html>
<html ng-app>
<body>
This is a sample HTML file.
</body>
</html>
```

## Is AngularJS supports MVVM with MVC?

Yes. AngularJS is an Advanced JavaScript framework. It supports both MVVM & MVC design patterns.

## What is routes in AngularJS?

In web development single page application designing is more popular than multi page application designing. Let us assume you have 3 pages in your application mainpage, aboutus & contactus. Using older standard of web development you created 3 html pages mainpage.html, aboutus.html & contactus.html. Depending upon the user requests you are redirecting the user to the respective html page. As an framework designer think for a while if we can keep a container or view in main page where you can display the respective pages depending upon the user request, is not this is a better approach to classical web development. AngularJS Routing & Views did the same. Using Routing & Views in AngularJS you can easily create single page web application with no page refresh on a new page redirect. To achieve this technique ng-view is act like the container & routing helps to load each page in the ng-view depending upon the user request. The shape of request is like the url /mainpage or /aboutus. To have a [demo in AngularJS Routing](#) Look at the Example.

## What is the difference between compile & link functions in angularjs?

Compile function used for Template DOM manipulation. Hence manipulations that apply to all DOM clones of the template associated with the directive.

Link function used for registering DOM listeners. A \$watch() function allows a directive to be notified of instance scope property changes, which allows the directive to render an updated instance value to the DOM by copying content from the instance scope into the DOM.

## What is difference between Service & Factory?

Factory allows us to add some logic before creating the object we require. It differs from service in a way where it allows us to pass the function which factory then invokes & returns the result.

To create factory we use module.factory('factoryName', function); & for service we use

`module.service('serviceName', function);`. If you're using an object, you could use the factory provider. If you're using a class you could use the service provider.

Source of Image: <https://fiveoclockclub.com>