

What is AngularJS?

AngularJS is a framework to build large scale and high performance web application while keeping them as easy-to-maintain. Following are the features of AngularJS framework.

- AngularJS is a powerful JavaScript based development framework to create RICH Internet Application RIARIA.
- AngularJS provides developers options to write client side application using JavaScript using JavaScript in a clean MVC ModelViewControllerModelViewController way.
- Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
- AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache License version 2.0.

What is data binding in AngularJS?

Data binding is the automatic synchronization of data between model and view components. ng-model directive is used in data binding.

What is scope in AngularJS?

Scopes are objects that refer to the model. They act as glue between controller and view.

What are the controllers in AngularJS?

Controllers are JavaScript functions that are bound to a particular scope. They are the prime actors in AngularJS framework and carry functions to operate on data and decide which view is to be updated to show the updated model based data.

What are the services in AngularJS?

AngularJS come with several built-in services. For example \$http service is used to make XMLHttpRequests (Ajax calls). Services are singleton objects which are instantiated only once in app.

What are the filters in AngularJS?

Filters select a subset of items from an array and return a new array. Filters are used to show filtered items from a list of items based on defined criteria.

Explain directives in AngularJS.

Directives are markers on DOM

elements such as elements, attributes, css, and more such as elements, attributes, css, and more.

These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives ng-bind, ng-model, etc ng-bind, ng-model, etc to perform most of the task that developers have to do.

Explain templates in AngularJS.

Templates are the rendered view with information from the controller and model. These can be a single file like index.html or multiple views in one page using "partials".

What is routing in AngularJS?

It is concept of switching views. AngularJS based controller decides which view to render based on the business logic.

What is deep linking in AngularJS?

Deep linking allows you to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.

What are the advantages of AngularJS?

Following are the advantages of AngularJS.

- AngularJS provides capability to create Single Page Application in a very clean and maintainable way.
- AngularJS provides data binding capability to HTML thus giving user a rich and responsive experience.
- AngularJS code is unit testable.
- AngularJS uses dependency injection and make use of separation of concerns.
- AngularJS provides reusable components.
- With AngularJS, developer writes less code and gets more functionality.
- In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.
- AngularJS applications can run on all major browsers and smart phones including Android and iOS based phones/tablets.

What are the disadvantages of AngularJS?

Following are the disadvantages of AngularJS.

- Not Secure – Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.
- Not degradable – If your application user disables JavaScript then user will just see the basic page and nothing more.

Which are the core directives of AngularJS?

Following are the three core directives of AngularJS.

- ng-app – This directive defines and links an AngularJS 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.

Explain AngularJS boot process.

When the page is loaded in the browser, following things happen:

- HTML document is loaded into the browser, and evaluated by the browser. AngularJS JavaScript file is loaded; the angular global object is created. Next, JavaScript which registers controller functions is executed.
- Next AngularJS scans through the HTML to look for AngularJS apps and views. Once view is located, it connects that view to the corresponding controller function.
- Next, AngularJS executes the controller functions. It then renders the views with data from the model populated by the controller. The page gets ready.

What is MVC?

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

- **Model** – It is the lowest level of the pattern responsible for maintaining data.
- **View** – It is responsible for displaying all or a portion of the data to the user.
- **Controller** – It is a software Code that controls the interactions between the Model and View.

Explain ng-app directive.

ng-app directive defines and links an AngularJS application to HTML. It also indicate the start of the application.

Explain ng-model directive.

ng-model directive binds the values of AngularJS application data to HTML input controls. It creates a model variable which can be used with the html page and within the container controlforexample,divforexample,div having ng-app directive.

Explain ng-bind directive.

ng-bind directive binds the AngularJS Application data to HTML tags. ng-bind updates the model created by ng-model directive to be displayed in the html tag whenever user input something in the control or updates the html control's data when model data is updated by controller.

Explain ng-controller directive.

ng-controller directive tells AngularJS what controller to use with this view. AngularJS application mainly relies on controllers to control the flow of data in the application. 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.

How AngularJS integrates with HTML?

AngularJS being a pure javaScript based library integrates easily with HTML.

Step 1 – Include angularjs javascript libray in the html page

```
<head>
  <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
</head>
```

Step 2 – Point to AngularJS app

Next we tell what part of the HTML contains the AngularJS app. This done by adding the ng-app attribute to the root HTML element of the AngularJS app. You can either add it to *html*element or *body* element as shown below:

```
<body ng-app = "myapp">
</body>
```

Explain ng-init directive.

ng-init directive initializes an AngularJS Application data. It is used to put values to the variables to be used in the application.

Explain ng-repeat directive.

ng-repeat directive repeats html elements for each item in a collection.

What are AngularJS expressions?

Expressions are used to bind application data to html. Expressions are written inside double braces like {{ expression}}. Expressions behave in same way as ng-bind directives. AngularJS application expressions are pure JavaScript expressions and outputs the data where they are used.

Explain uppercase filter.

Uppercase filter converts a text to upper case text.

In below example, we've added uppercase filter to an expression using pipe character. Here we've added uppercase filter to print student name in all capital letters.
Enter first name: <input type = "text" ng-model = "student.firstName">
Enter last name: <input type = "text" ng-model = "student.lastName">
Name in Upper Case: {{student.fullName() | uppercase}}

Explain lowercase filter.

Lowercase filter converts a text to lower case text.

In below example, we've added lowercase filter to an expression using pipe character. Here we've added lowercase filter to print student name in all lowercase letters.
Enter first name: <input type = "text" ng-model = "student.firstName">
Enter last name: <input type = "text" ng-model = "student.lastName">
Name in Upper Case: {{student.fullName() | lowercase}}

Explain currency filter.

Currency filter formats text in a currency format.

In below example, we've added currency filter to an expression returning number using pipe character. Here we've added currency filter to print fees using currency format.
Enter fees: <input type = "text" ng-model = "student.fees">
fees: {{student.fees | currency}}

Explain filter filter.

filter filter is used to filter the array to a subset of it based on provided criteria.
In below example, to display only required subjects, we've used subjectName as filter.

Enter subject: <input type = "text" ng-model = "subjectName">

Subject:

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

Explain orderby filter.

orderby filter orders the array based on provided criteria.

In below example, to order subjects by marks, we've used orderBy marks.

Subject:

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

Explain ng-disabled directive.

ng-disabled directive disables a given control.

In below example, we've added ng-disabled attribute to a HTML button and pass it a model. Then we've attached the model to an checkbox and can see the variation.

```
<input type = "checkbox" ng-model = "enableDisableButton">Disable Button
<button ng-disabled = "enableDisableButton">Click Me!</button>
```

Explain ng-show directive.

ng-show directive shows a given control.

In below example, we've added ng-show attribute to a HTML button and pass it a model. Then we've attached the model to a checkbox and can see the variation.

```
<input type = "checkbox" ng-model = "showHide1">Show Button  
<button ng-show = "showHide1">Click Me!</button>
```

Explain ng-hide directive.

ng-hide directive hides a given control.

In below example, we've added ng-hide attribute to a HTML button and pass it a model. Then we've attached the model to a checkbox and can see the variation.

```
<input type = "checkbox" ng-model = "showHide2">Hide Button  
<button ng-hide = "showHide2">Click Me!</button>
```

Explain ng-click directive.

ng-click directive represents a AngularJS click event.

In below example, we've added ng-click attribute to a HTML button and added an expression to updated a model. Then we can see the variation.

```
<p>Total click: {{ clickCounter }}</p></td>  
<button ng-click = "clickCounter = clickCounter + 1">Click Me!</button>  
|
```

How angular.module works?

angular.module is used to create AngularJS modules along with its dependent modules.

Consider the following example:

```
var mainApp = angular.module("mainApp", []);
```

Here we've declared an application **mainApp** module using angular.module function. We've passed an empty array to it. This array generally contains dependent modules declared earlier.

How to validate data in AngularJS?

AngularJS enriches form filling and validation. We can use **\$dirty** and **\$invalid** flags to do the validations in seamless way. Use **novalidate** with a form declaration to disable any browser specific validation.

Following can be used to track error.

- **\$dirty** – states that value has been changed.
- **\$invalid** – states that value entered is invalid.
- **\$error** – states the exact error.

Explain ng-include directive.

Using AngularJS, we can embed HTML pages within a HTML page using ng-include directive.

```
<div ng-app = "" ng-controller = "studentController">  
  <div ng-include = "main.htm"></div>  
  <div ng-include = "subjects.htm"></div>  
</div>
```

How to make an ajax call using Angular JS?

AngularJS provides **http** control which works as a service to make ajax call to read data from the server. The server makes a database call to get the desired records. AngularJS needs data in JSON format. Once the data is ready, **http** control which works as a service to make ajax call to read data from the server. The server makes a database call to get the desired records. AngularJS needs data in JSON format. Once the data is ready, **http** can be used to get the data from server in the following manner:

```
function studentController($scope, $http) {
  var url = "data.txt";
  $http.get(url).success(function(response) {
    $scope.students = response;
  });
}
```

What is use of \$routeProvider in AngularJS?

\$routeProvider is the key service which set the configuration of urls, maps them with the corresponding html page or ng-template, and attaches a controller with the same.

What is \$rootScope?

Scope is a special JavaScript object which plays the role of joining controller with the views. Scope contains the model data. In controllers, model data is accessed via scopeobject.scopeobject.rootScope is the parent of all of the scope variables.

What is scope hierarchy in AngularJS?

Scopes are controllers specific. If we define nested controllers then child controller will inherit the scope of its parent controller.

```
<script>
```

```
var mainApp = angular.module("mainApp", []);
mainApp.controller("shapeController", function($scope) {
  $scope.message = "In shape controller";
  $scope.type = "Shape";
});
mainApp.controller("circleController", function($scope) {
  $scope.message = "In circle controller";
});
</script>
```

Following are the important points to be considered in above example.

- We've set values to models in shapeController.
- We've overridden message in child controller circleController. When "message" is used within module of controller circleController, the overridden message will be used.

What is a service?

Services are JavaScript functions and are responsible to do specific tasks only. Each service is responsible for a specific task for example, http is used to make ajax call to get the server data. http is used to make ajax call to get the server data. route is used to define the routing information and so on. Inbuilt services are always prefixed with \$ symbol.

What is service method?

Using service method, we define a service and then assign method to it. We've also injected an already available service to it.

```
mainApp.service('CalcService', function(MathService){
  this.square = function(a) {
    return MathService.multiply(a,a);
  }
})
```

});

What is factory method?

Using factory method, we first define a factory and then assign method to it.

```
var mainApp = angular.module("mainApp", []);
mainApp.factory('MathService', function() {
    var factory = {};
    factory.multiply = function(a, b) {
        return a * b;
    }
    return factory;
});
```

What are the differences between service and factory methods?

factory method is used to define a factory which can later be used to create services as and when required whereas service method is used to create a service whose purpose is to do some defined task.

Which components can be injected as a dependency in AngularJS?

AngularJS provides a supreme Dependency Injection mechanism. It provides following core components which can be injected into each other as dependencies.

- value
- factory
- service
- provider
- constant

What is provider?

provider is used by AngularJS internally to create services, factory etc. during config phasephaseduringwhichAngularJSbootstrapsitselfphaseduringwhichAngularJSbootstrapsitself. Below mention script can be used to create MathService that we've created earlier. Provider is a special factory method with a method get which is used to return the value/service/factory.

```
//define a module
var mainApp = angular.module("mainApp", []);

...
//create a service using provider which defines a method square to return square of a number.
mainApp.config(function($provide) {
    $provide.provider('MathService', function() {
        this.$get = function() {
            var factory = {};
            factory.multiply = function(a, b) {
                return a * b;
            }
            return factory;
        };
    });
});
```

What is constant?

constants are used to pass values at config phase considering the fact that value cannot be used to be passed during config phase.

```
mainApp.constant("configParam", "constant value");
```

Is AngularJS extensible?

Yes! In AngularJS we can create custom directive to extend AngularJS existing functionalities. Custom directives are used in AngularJS to extend the functionality of HTML. Custom directives are defined using "directive" function. A custom directive simply replaces the element for which it is activated. AngularJS application during bootstrap finds the matching elements and do one time activity using its compile method of the custom directive then process the element using link method of the custom directive based on the scope of the directive.

On which types of component can we create a custom directive?

AngularJS provides support to create custom directives for following type of elements.

- **Element directives** – Directive activates when a matching element is encountered.
- **Attribute** – Directive activates when a matching attribute is encountered.
- **CSS** – Directive activates when a matching css style is encountered.
- **Comment** – Directive activates when a matching comment is encountered.

What is internationalization?

Internationalization is a way to show locale specific information on a website. For example, display content of a website in English language in United States and in Danish in France.

How to implement internationalization in AngularJS?

AngularJS supports inbuilt internationalization for three types of filters currency, date and numbers. We only need to incorporate corresponding js according to locale of the country. By default it handles the locale of the browser. For example, to use Danish locale, use following script

Descriptive

1. What is PrimeFaces?

Ans: PrimeFaces is a lightweight JSF component library with one JAR file, which needs no configuration and does not contain any required external dependencies.

2. What are the namespace for PrimeFaces components?

Ans: namespace for PrimeFaces components is as follows:

```
xmlns:p="http://primefaces.org/ui"
```

For PrimeFaces Mobile, the namespace is as follows:

```
xmlns:p="http://primefaces.org/mobile"
```

3. How to implement Internationalization in PrimeFaces?

Ans: For Internationalization, first we need to specify the resource bundle definition under the application tag in faces-config.xml, as follows:

```
<application>
  <locale-config>
    <default-locale>en</default-locale>
    <supported-locale>tr_TR</supported-locale>
  </locale-config>
  <resource-bundle>
    <base-name>messages</base-name>
    <var>msg</var>
  </resource-bundle>
</application>
```

A resource bundle would be a text file with the .properties suffix that would contain the locale-specific messages. So, the preceding definition states that the resource bundle messages_{localekey}.properties file will reside under classpath and the default value of localekey is en.

4. What are PrimeFaces dependencies?

Ans: PrimeFaces only requires a JAVA 5+ runtime and a JSF 2.x implementation as mandatory dependencies. There're some optional libraries for certain features.

5. How to configure PrimeFaces?

Ans: PrimeFaces does not require any mandatory configuration and follows configuration by exception pattern of Java EE. Here is the list of all configuration options defined with a context-param such as;

```
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>bootstrap</param-value>
</context-param>
```

1. Which is/are includes of panel

- a. toggling
- b. closing
- c. built-in pop-up menu
- d. AJAX event listeners
- e. All of the above

2. Which of the default value of toggleSpeed attribute

- a) a.100 milliseconds.
- b) b. 1000 milliseconds.
- c) c.10000 milliseconds.
- d) d. none of the above

3. AccordionPanel is a container component that provides the ability to group

- A. Multiple tabs.
- B. b. Single tabs.

4. The scrollPanel component provides customizable scroll bars instead of the browser's scrolls.

- a. False
- A. b.True

5. Which of the component includes the features such as dynamic content, loading, orientations, and programmatically managing tabs.

- a. a.layoutView
- b. gridView
- c. c.panelView
- d. d.tabView

6. The By default Orientation of the tabs is/are

- a) bottom
- b) top
- c) left
- d) right

7. Which is not includes the Transition effects

- a. blind
- b. put
- c. puff
- d. fade

8. Which component provides the Nested layouts

- a. grid unit component
- b. panel unit component
- b) layout unit component
- c) None of the above