

AngularJS Interview Questions

 codespaghetti.com/angularjs-interview-questions/

Angular JS

Angular JS Interview Questions, Best AngularJS Programs and Examples to help you ace your next Job interview.



Table of Contents:

CHAPTER 1: Top 5 Angular JS Interview Questions

CHAPTER 2: Top 5 Mistakes AngularJS Dev. Makes

CHAPTER 3: What Is Angular JS

CHAPTER 4: Advantages Of Angular JS

CHAPTER 5: Angular JS Interview Questions

CHAPTER 6: Angular JS Directive Questions

CHAPTER 7: Angular JS Security Questions

CHAPTER 8: Angular JS Testing Interview Questions

CHAPTER 9: AngularJS VS JQuery

CHAPTER 10: AngularJS VS ReactJS

CHAPTER 11: AngularJS Interview Resources

CHAPTER 12: Summary

CHAPTER 13: AngularJS Interview Questions PDF

Importance of AngularJS

AngularJS is a very popular framework among developers. Small to large companies are using angularJS. If your going for an interview as a front end developers then get ready to answer angular JS questions.

In this guide i have collected and curated the best angularJS interview questions. Go through these questions and you will increase your chances of success in interviews. Before we dive deep into this guide. Let's see what are the top 5 questions asked in angularJS interviews

Top 5 AngularJS Interview Questions



Top 5 Mistakes Angular Js Developers Make

These are the top 5 mistakes developers make while developing applications using AngularJS

1. Heavy reliance on \$scope & not using controller instead
2. Overusing \$broadcast and \$emit
3. Overusing \$watch
4. Failing to Test effectively
5. Hacking the DOM

What is Angular Js

- AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly.
- AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.
- AngularJS provides developers options to write client side application (using JavaScript) in a clean MVC way.
- Application written in AngularJS is cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.

Question: What Are Advantages Of Using AngularJS Framework?

Some of the key advantages of using AngularJS framework are:

- It provides an excellent experience to the end user.
- It free's the developers from having to register callbacks manually or write repetitive low-level DOM manipulation tasks.
- By separating DOM manipulation from app logic, it makes code modular and easy to test.
- It supports two-way data binding.

Question: What Are The Main Features Of AngularJS ?

Some of the most prominent feature of AngularJS are.

Data-binding – Handles synchronization of data across model, controllers, and view.

Scope – Object representing the model, acts as a glue layer between controller and view.

Controllers – JS functions bound to the scope object.

Services – Substitutable objects that are wired together using dependency injection. e.g. \$location service.

Filters – Formats the value of an expression for displaying to the user. e.g., uppercase, lowercase.

Directives – These are extended HTML attributes start with the “ng-” prefix. e.g., ng-app directive used to initialize the angular app.

Templates – HTML code including AngularJS specific elements and attributes.

Routing – It’s an approach to switch views.

MVC pattern – A design pattern made up of three parts.

- **Model** – Represents data, could be static data from a JSON file or dynamic data from a database.
- **View** – Renders data for the user.
- **Controller** – Gives control over the model and view for collating information to the user.

Deep linking – Enables the encoding of the application state in the URL and vice versa.

Dependency injection – A design pattern to let the components injected into each other as dependencies

Question: What Is A Digest Cycle in AngularJS?

During every digest cycle, all new scope model values are compared against the previous values. This is called dirty checking. If change is detected, watches set on the new model are fired and another digest cycle executes. This goes on until all models are stable.

The digest cycle is triggered automatically but it can be called manually using “.\$apply()”.

Question: What Are Ways Of Communication Between Modules Of Application, Using AngularJS Functionality?

The common ways of communication are:

- Using events or services
- By assigning models on **\$rootScope**
- Directly between controllers using **ControllerAs** and other forms of inheritance
- Directly between controllers using **\$parent, \$\$childHead, \$\$nextSibling**

Question: How Digest Cycle Can Be Decreased?

It can be decreased by decreasing the number of watchers. To do this you can:

- Use web worker
- Work in batches
- Cache DOM

- Remove unnecessary watchers
 - Use one-time Angular binding
-

Question: Explain The Steps Involved In Boot Process For AngularJS?

Whenever a web page loads in the browser, following steps execute in the background.

- First, the HTML file containing the code gets loaded into the browser. After that, the JavaScript file mentioned in the HTML code gets loaded. It then creates a global object for angular. Now, the JavaScript which displays the controller functions gets executed.
 - In this step, AngularJS browses the complete HTML code to locate the views. If the same is available, then Angular links it to the corresponding controller function.
 - In this step, AngularJS initiates the execution of required controller functions. Next, it populates the views with data from the model identified by the controller. With this the page is ready.
-

Question: Do AngularJS Provide Any Security Features?

AngularJS provides built-in protection from the following security flaws.

- It prevents cross-site scripting attacks: Cross-site scripting is a technique where anyone can send a request from client side and can get the confidential information easily.
 - It prevents HTML injection attacks.
 - It prevents XSRF protection for server-side communication: “Auth token” mechanism can handle it. When the user logs for the first time a user id and password is sent to the server, and it will, in turn, return an auth token. Now, this token does the authentication in the future transactions.
-

Question: 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.
-

Question:

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

```
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";
});
```

Question: What Is Provider?

provider is used by AngularJS internally to create services, factory etc. during config phase(phase during which AngularJS bootstraps itself).

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;
        };
    });
});
```

Question: What Are Filters? Explain Different Filters Provided By AngularJS?

An AngularJS Filter changes or transforms the data before passing it to the view. These Filters work in combination with AngularJS expressions or directives. AngularJS uses pipe character ("|") to add filters to the expressions or directives. For example:

```
<p> {{ bid | currency }} </p>
```

The above example is an expression enclosed in the curly braces. The filter used in this expression is `currency`. Also important to note that filters are case-sensitive. AngularJS provides following filters to transform data.

currency – It is used to format a number to a currency format.

date – It is required to format a date to a specified format.

filter – It chooses a subset of items from an array.

json – It formats an object to a JSON string.

limitTo – Its purpose is to create an array or string containing a specified number of elements/characters.

The elements are selected, either from the beginning or the end of the source array or string. This depends on the value and sign (positive or negative) of the limit.

lowercase – This filter converts a string to lower case.

number – It formats a number as text.

orderBy – It enables to sort an array. By default, sorting of strings happens alphabetically. And sorting of numbers is done numerically.

It also supports a comparator function where we can define what will be counted as a match or not.

uppercase – This filter converts a string to upper case

Question: What Are Compile, Pre, And Post Linking In AngularJS?

Compile – It collects an HTML string or DOM into a template and creates a template function. It can then be used to link the scope and the template together.

AngularJS uses the `compile` function to change the original DOM before creating its instance and before the creation of scope. Before discussing the `Pre-Link` and the `Post-Link` functions let's see the `Link` function in detail.

Link – It has the duty of linking the model to the available templates. AngularJS does the data binding to the compiled templates using `Link`. Following is the `Link` syntax.

```
link: function LinkFn(scope, element, attr, ctrl){}
```

Where each of the four parameters is as follows-

Scope – It is the scope of the directive. **element** – It is the DOM element where the directive has to be applied.

attr- It is the Collection of attributes of the DOM element.

ctrl – It is the array of controllers required by the directive.

AngularJS allows setting the link property to an object also. The advantage of having an object is that we can split the link function into two separate methods called, pre-link and post-link.

Post-Link – Execution of Post-Link function starts after the linking of child elements. It is safer to do DOM transformation during its execution. The post-link function is suitable to execute the logic.

Pre-Link – It gets executed before the child elements are linked. It is not safe to do DOM transformation. As the compiler linking function will fail to locate the correct elements.

It is good to use the pre-link function to implement the logic that runs when AngularJS has already compiled the child elements. Also, before any of the child element's post-link functions have been called.

Question: Explain AngularJS Application Life-Cycle?

Understanding the life cycle of an AngularJS application makes it easier to learn ,design and implement the code. AngularJS App life cycle consists of following three phases

-Bootstrap -Compilation -Runtime

These three phases of the life cycle occur each time a web page of an AngularJS application gets loaded into the browser. Let's learn about each of the three stages in detail:

The Bootstrap Phase – In this phase, the browser downloads the AngularJS javascript library. .

The Compilation Phase – The second phase of the AngularJS life cycle is the HTML compilation stage. Initially, when a web page loads in the browser, a static form of the DOM gets loaded.

During the compilation phase, this static DOM gets replaced with a dynamic DOM which represents the app view.

There are two main steps – first, is traversing the static DOM and collecting all the directives.

These directives are now linked to the appropriate JavaScript functionality which lies either in the AngularJS built-in library or custom directive code. The combination of directives and the scope, produce the dynamic or live view.

The Runtime Data Binding Phase – This is the final phase of the AngularJS application. It remains until the user reloads or navigates to a different web page.

At this point, any changes in the scope get reflected in the view, and any changes in the view are directly updated in the scope, making the scope the single source of data for the

view.

This shows that AngularJS behaves differently from traditional methods of binding data.

The traditional methods combine a template with data, received from the engine and then manipulate the DOM each time there is any change in the data.

However, AngularJS compiles the DOM only once and then links the compiled template as necessary, making it much more efficient than the traditional methods.

Question: What is Link Function And How It Differs From Compile?.

The link function combines the directives with a scope to produce a live view. The link function is responsible for instance DOM manipulation and for registering DOM listeners.

The compile function is responsible for template DOM manipulation as well as the collection of all the directives.

Question: When a scope is terminated, two similar “destroy” events are fired. What are they used for, and why are there two?

The first one is an AngularJS event, “\$destroy”, and the second one is a jqLite / jQuery event “\$destroy”.

The first one can be used by AngularJS scopes where they are accessible, such as in controllers or link functions.

Consider the two below happening in a directive’s postLink function. The AngularJS event:

```
scope.$on('$destroy', function () { // handle the destroy, i.e. clean up. });
```

```
And element.on('$destroy', function () { // respectful jQuery plugins already have this handler. // angular.element(document.body).off('someCustomEvent'); });
```

The jqLite / jQuery event is called whenever a node is removed, which may just happen without scope teardown.

Question: List a few ways to improve the performance of AngularJS application.

The two officially recommended methods are disabling debug data and enabling strict DI mode.

The first one can be enabled through the \$compileProvider:

```
myApp.config(function
($compileProvider)
{ $compileProvider.debugInfoEnabled(false);
});
```

That tweak disables appending scope to elements, making scopes inaccessible from the console. The second one can be set as a directive:

The performance gain lies in the fact that the injected modules are annotated explicitly, hence they don't need to be discovered dynamically.

You don't need to annotate yourself, just use some automated build tool and library for that.

Two other popular ways are:

Using one-time binding where possible. Those bindings are set, e.g. in “{{ ::someModel }}” interpolations by prefixing the model with two colons.

In such a case, no watch is set and the model is ignored during digest.

```
Making $httpProvider use applyAsync:
myApp.config(function ($httpProvider) {
$httpProvider.useApplyAsync(true);
});
```

which executes nearby digest calls just once, using a zero timeout.

Question: What are the DOM and BOM in Angular JS?

The DOM is the Document Object Model. It's the view part of the UI. Whatever we change in page elements is reflected in the DOM.

BOM is the Browser Object Model, which specifies the global browser objects like window, local storage, and console.

Questions: What Are Controllers In AngularJS?

Controllers are simple JavaScript functions that are bound to a particular scope. They are the prime actors in AngularJS framework and carry functions to operate on data.

As the name says they control how data flows from the server to HTML.

For example here is simple “User” controller which provides data via “UserName” and “UserCode” property and Add/ Update logic to save the data to database.

```
function User($scope)
{
    $scope.UserName = "Arthur";
    $scope.UserCode = "2111";
    $scope.Add = function () {
    }
    $scope.Update = function () {
    }
}
```

Questions: What Is "ng-controller"?

"ng-controller" is a directive. Controllers are attached to the HTML UI by using the "ng-controller" directive tag and the properties of the controller are attached by using "ng-model" directive.

For example below is a simple HTML UI which is attached to the "User" controller via the "ng-controller" directive and the properties are binded using "ng-model" directive.

```
<div ng-controller="User">
<input type="text" id="UserName" ng-model="UserName"/><br />
<input type="text" id="UserCode" ng-model="UserCode"/>
</div>
```

Questions: What Are Expressions In AngularJS?

AngularJS expressions are unit of code. Which 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.

The below expression adds two constant values.

```
{{1+1}}
```

The below expression multiplies quantity and cost to get the total value.

```
The value total cost is {{ quantity * cost }}
```

The below expression displays a controller scoped variable.

```
<div ng-controller="UserV">
The value of Customer code is {{UserCode}}
</div>
```

```
The value of User code is {{UserCode}}
```

Question: How To initialize AngularJS application data?

We can use “ng-init” directive to do this. In the example below “ng-init” directive is used to initialize the “pi” value.

```
<body ng-app="userApp" ng-init="p=3.14">
The value of p is {{p}}
</body>
```

Question: How To Define Scope In AngularJS?

“\$scope” is an object instance of a controller. “\$scope” object instance is created when “ng-controller” directive is encountered.

For example, we have two controllers “ScopeFunction” and “TestFunction”. In both controllers we have a “ControllerName” variable.

```
function ScopeFunction($scope)
{
$scope.ControllerName = "ScopeFunction";
}
function TestFunction($scope)
{
$scope.ControllerName = "TestFunction";
}
```

Now to attach the above controllers to HTML we should use “ng-controller” directive. For instance you can see below how “ng-controller” directive attaches “ScopeFunction” with “div1” tag and “Testfunction” with “div2” tag.

```
<div id="div1" ng-controller="ScopeFunction">
Instance of {{ControllerName}} created
</div>
<div id="div2" ng-controller="TestFunction">
Instance of {{ControllerName}} created
</div>
```

AngularJS Directives



Question: What Are Directives?

Directives are special markers on a DOM element that tell the html compiler to attach a specified behavior to the DOM element. Directives start with ng-prefix. Some of the built-in directives include ngClass, ngApp, ngRepeat, ngModel, ngBind and ngInit

Question: What Are Different Types of Directives?

There are four types of directives:

- Comment directives
 - CSS class directives
 - Attribute directives
 - Element directives
-

Question: What Are Different Ways To Invoke A Directive?

There are four different ways to invoke a directive in an angular application. They are as follows.

1) As an attribute:

```
<span my-directive></span>
```

2) As a class:

```
<span class="my-directive: expression;"></span>
```

3) As an element:

```
<my-directive></my-directive>
```

4) As a comment:

```
<!-- directive: my-directive expression -->
```

Question: How Can We Create A Custom Directive In Angular?

If we want to create our own custom Angular directive and attach it with HTML elements as shown in the below code.

To create a custom directive we need to use the “directive” function to register the directive with angular application. When we call the “register” method of “directive” we need to specify the function which will provide the logic for that directive.

For example in the below code we have created a copy right directive and it returns a copy right text.

Please note “app” is an angular application object which has been explained in the previous sections.

```
app.directive('companyCopyRight', function ()
{
return
{
template: '@CopyRight questpond.com '
};
});
```

Question: What Kind Of Naming Conventions Is Used For Custom Directives?

For angular custom directive the best practice is to follow camel casing and that also with atleast two letter's. In camel case naming convention we start with a small letter, followed by a capital letter for every word.

Some example of camel cases are “loopCounter” , “isValid” and so on.

So when you register a custom directive it should be with camel case format as shown in the below code “companyCopyRight”.

```
app.directive('companyCopyRight', function ()
{
return
{
template: '@CopyRight questpond.com '
};
});
```

AngularJS Custom Directives Resources ,Examples & code samples

- <https://code.tutsplus.com/tutorials/mastering-angularjs-directives--cms-22511>
 - <https://www.codeschool.com/blog/2015/03/06/digging-advanced-angularjs-directives/>
 - <https://docs.angularjs.org/guide/directive>
 - <https://github.com/huseyinbabal/mastering-angularjs-directives>
 - <https://gist.github.com/umidjons/6669708>
-

AngularJs Custom Directives Video Tutorial

Angular JS Security Interview Questions

Question: What Are The Security Features Provided By AngularJS?

AngularJS provides protection from the following security flaws.

- It prevents cross-site scripting attacks: Cross-site scripting is a technique where anyone can send a request from client side and can get the confidential information easily.
 - It prevents against HTML injections.
 - It prevents XSRF(Cross-site request forgery) protection for server-side communication.
-

Question: What Are The Web Application Security Risks That A Web Developer Should Avoid While Doing Development Using AngularJS?

Following are the most critical web application development flaws that a developer should take care of:

- Injection attack.
- Broken Authentication and Session Management.
- Cross-Site Scripting (XSS)

- Insecure direct object references.
- Security Misconfiguration.
- Sensitive Data Exposure.
- Cross-Site Request Forgery (CSRF).
- Missing Function Level Access Control.
- Using components that possess vulnerabilities.
- In validated redirects and forwards.

Testing AngularJS

Question: How to test Angular JS application?

Since Angular JS facilitates separation of client-side components with the help of inbuilt dependency injection (DI) management.

it has testing support for every component e.g. Angular Controller, Angular Service, etc. It's an always good idea of testing client-side components before deployment.

Here are some frameworks which can be used for testing AngularJS applications.

Jasmine - is a behavior-driven development framework for testing JavaScript code.

A major feature of this framework is that it does not depend on any JavaScript framework and neither it needs DOM to test the code.

This provides for clean syntax using which we can write tests. Jasmine also provides functions for structuring the test using assertions.

Find More information at <http://jasmine.github.io/2.4/introduction.html>

Protractor - is an end-to-end testing framework for Angularjs applications.

This framework runs tests against the application running in a real browser.

This is a NodeJS program and uses Jasmine for its test syntax.

Find more information at <http://angular.github.io/protractor/#/api>.

Karma - is a JavaScript command line tool. This tool is used to use the web server to load the application code and test it.

This tool runs on NodeJS and is available as a NPM package.

Find more information at <http://karma-runner.github.io/0.13/index.html>

AngularJS vs jQuery

Comparison Chart Between AngularJS And JQuery?

Function	AngularJS	JQuery
Abstracts the DOM	Yes	Yes
Unit Test Runner	Yes	Yes
Deferred Promises	Yes	Yes
JSONP/AJAX	Yes	Yes
RESTFUL API	No	Yes
MVC Pattern Support	No	Yes
Templating	No	Yes
Form Validating	No	Yes
Localization	No	Yes

WHAT IS JQUERY?

jQuery is a DOM manipulation library that makes it easier to use JavaScript on your website. jQuery simplifies the things for developers.

It takes the complex code that would be required to make AJAX calls or manipulate the DOM and wraps them into simple methods you can call with a single line of JavaScript.

Here are some of the key technical features that are available in the jQuery library:

- HTML/DOM manipulation
- CSS manipulation
- Effects and animations
- Cross-browser compatibility
- AJAX/JSONP
- Event handling

jQuery can be used in conjunction with other frameworks, including AngularJS. In fact, AngularJS is built off of an implementation of jQuery called jqLite.

Since jQuery has no real structure, the developer has full freedom to build projects as they see fit.

However the lack of structure also means it's easier to fall into the trap of "spaghetti code,"

which can lead to confusion on larger projects with no clear design direction or code maintainability. For these situations, a framework like AngularJS can be a big help.

When To Use jQuery

If you are looking for a lightweight but powerful tool for DOM manipulation. jQuery will serve you the best in this regard. It is designed to help you manipulate the DOM with JavaScript.

Whether that involves building a website from scratch alongside vanilla JavaScript, or complementing an existing framework with a jQuery plugin,

jQuery shines when you require flexibility or see a particular feature that you wish to include in your app, like a slider. It's also great for quickly scripting solutions that work to test an idea.

WHAT IS ANGULARJS?

AngularJS is a JavaScript framework that was specifically designed to help developers build SPAs in accordance with best practices for web development.

By providing a structured environment for building SPAs, the risk of producing “spaghetti code” is greatly reduced.

AngularJS also provides a suite of features that help the developer do more with less code than would be required with “vanilla JavaScript” (developer speak for plain, simple JavaScript without the use of any libraries) and jQuery.

Some of the key features AngularJS adds on top of jQuery are:

- Angular directives
- Templating
- Dependency injection
- Two-way data binding
- Support for MVC
- RESTful API
- Form validation

It goes without saying that AngularJS also abstracts the DOM, has a system for managing event handlers, and supports AJAX/JSONP.

When To Use ANGULARJS

If you are looking for a full featured framework for developing web applications from scratch.

AngularJS gives you everything you need to build the client-side of an application.

It will also make it easier to keep your web project organized and modular to avoid repeating code.

Advanced features like two-way data binding and dependency injection allow you to quickly create visually stunning and engaging applications.

AngularJS vs ReactJS

React JS is a library that deals with views, AngularJS is a full fledge framework.

The primary difference between AngularJS and ReactJS lies in state management. Angular has data-binding bundled in by default, whereas React is generally augmented by Redux to give unidirectional data flow and work with immutable data.

Both are opposing approaches and there is no consensus on which is better: mutable/data binding or immutable/unidirectional.

Here is a Comparison chart

Technology	React JS	Angular JS	Angular 2
Author	Facebook Community	Google	Google
Type	Open Source JS library	Fully-featured MVC framework	Fully-featured MVC framework
Tool Chain	High	Low	High
Language	JSX	JavaScript, HTML	TypeScript
Learning Curve	Low	High	Medium
Packaging	Strong	Weak	Medium
Rendering	Server Side	Client Side	Server Side
App Architecture	None, combined with Flux	MVC	Component-Based
Data Binding	Uni-Directional	Bi-Directional	Bi-Directional
DOM	Virtual DOM	Regular DOM	Regular DOM
Latest Version	15.4.0 (November 2016)	1.6.0	2.2.0 (November 2016)

Scalability:

Angular is easy to scale thanks to its design as well as a powerful CLI. React is testable and therefore scalable compared to other frameworks.

Dependency Injection:

Dependency injection is a bone of contention as it is contrary to the prevalent React paradigm of functional programming and immutability.

But some sort of dependency injection is almost unavoidable in data-binding environments, because it aids in decoupling (and thereby mocking and testing) where there is no separate data-layer architecture.

Angular supports dependency injection and one great advantage is the ability to have different life cycles for different stores.

Some of the common React paradigms deploy some kind of global app state that maps to disparate components, however it is conducive to the introduction of bugs when cleaning

the global state on component unmount.

On the other hand, a store that is created on component mount and is seamlessly available to the component's children –is a more useful and often neglected concept.

This is out of the box in Angular, but quite easily replicated with MobX as well.

Simplicity + Code length:

React is quite easy and simple to understand but it takes quite some time to set up a project in React.

Angular on the other hand, is not simple by any means. Its inherent complexity sometimes causes confusion and Angular specific 3rd party libraries and syntax.

Overall, both frameworks provide a robust set of tools for quality, scalable, reactive web-based applications.

For those who prefer to code in classic JS, React may find more favors, but for those looking at a more mature and sophisticated solution, AngularJS might be your best bet.

Angular JS Interview Questions From Around the Web:

In this section you will find a list of excellent AngularJS resources.

Best AngularJS Interview Resources:

- <https://gainfromhere.com/learn-angularjs-framework-courses-online/>
- <https://angularjs.org/>
- <https://www.codeschool.com/courses/shaping-up-with-angularjs>
- <https://egghead.io/>

AngularJS Interview Questions:

- <http://www.techbeamers.com/latest-angularjs-interview-questions-answers/>
- <https://www.toptal.com/angular-js/interview-questions>
- <https://www.codementor.io/blog/angularjs-interview-questions-sample-answers-du1081n7p>
- <https://www.janbasktraining.com/blog/angularjs-interview-questions-answers/>
- <http://www.codespaghetti.com/interview-questions/>

Keys To Interview Success

Angular provides a way for developers to quickly build modern web applications. Angular was created by Google and is now operated as an open source project.

Angular JS is getting famous day by day. In order to get hired as front end developer. You need to prepare angularJS interview questions.

Programming interviews are not easy and your only hope to survive is to prepare very seriously.

Understand the core concepts about angularJS. What are the pros and cons of using angularJS in your projects. And how an angularJS application can be tested.

In this guide all of these concepts are covered in details.

ANGULARJS FACTS

www.codespaghetti.com

HAS TWO-WAY DATABINDING

Models and views can be bound to update each other.

MINIFIED FILE SIZE WITHOUT DEPENDENCIES

Smaller file size means faster load times.

MODELS ARE PLAIN JAVA SCRIPT OBJECTS

Easier data transfer between the application and REST-Services or the LocalStorage. It also reduces complexity from getter- and setter functions.

SUPPORTS ROUTING AND DEEP LINKING

Routing allows interaction with the browser url. This is very important for a good user experience if you are building one-page apps.

SUPPORTS ONE- WAY DATA BINDING

Views can be bound to update automatically when an observable object changes.

HAS AUTOMATIC VIEW FILTERING

Has a mechanism to filter views based on scope variables (for example when searching) without the need to write much extra code.

IN A NUTSHELL

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly.

AngularJS Interview Questions PDF

About The Author

References

- <https://www.upwork.com/hiring/development/angularjs-vs-jquery/>
- <https://www.airpair.com/angularjs/posts/jquery-angularjs-comparison-migration-walkthrough>
- <https://www.rishabhsoft.com/blog/reactjs-vs-angularjs>
- <http://www.dotnetcurry.com/angularjs/1255/unit-testing-angularjs-using-jasmine-karma>
- <https://scotch.io/tutorials/testing-angularjs-with-jasmine-and-karma-part-1>