

Angular Interview Questions

Download PDF

Print

Use the filter to quickly find topics like components, DI, RxJS/observables, change detection, forms, routing, and performance.

Filter (e.g., DI, RxJS, change detection, NgModule, routing)...

Showing 62 of 62

1. What is Angular and what is it used for?

Angular is a popular open-source framework for building dynamic web applications. It is used for developing complex, scalable and maintainable web applications that are capable of running fast and responsive to user interactions.

2. What is a Component in Angular?

A component in Angular is a basic building block of the framework that represents a reusable UI element. Components are responsible for rendering the UI and controlling the flow of data in an Angular application.

3. What is two-way data binding in Angular?

Two-way data binding in Angular is a feature that allows changes in the UI to immediately reflect the underlying data model and vice versa. This is achieved by binding the data model to the UI element and automatically updating the data model whenever the UI element changes and vice versa.

4. What is Angular CLI and what is it used for?

Angular CLI is a powerful command-line interface tool that makes it easy to generate, build, and deploy Angular applications. It can be used for generating new components, services, and other parts of an Angular application, as well as for building and deploying the application to a production environment.

5. What is lazy-loading in Angular and how does it improve performance?

Lazy-loading in Angular is a technique for loading only the necessary parts of an application on-demand, instead of loading everything at once. This results in faster initial load times and improved overall performance, as the browser only has to download and render the parts of the application that are actually being used.

6. What is AOT (Ahead-of-Time) Compilation in Angular and why is it important?

Ahead-of-Time (AOT) Compilation is a feature in Angular that compiles the application's components and templates into efficient JavaScript code during the build process. This results in faster initial load times, as the browser does not have to interpret the Angular templates and components at runtime. AOT also catches any syntax or semantic errors during the build process, instead of at runtime, making the application more stable and less prone to runtime errors.

7. What is dependency injection in Angular and how does it work?

Dependency injection is a design pattern in Angular that allows components to receive instances of services or other dependencies from a central location. Angular uses a Dependency Injection (DI) system to manage the creation and injection of dependencies into components. This makes it easy to manage and maintain the dependencies needed by the components, and also makes it easy to switch between different implementations of a dependency without having to modify the components that use it.

8. What is the difference between Angular and AngularJS?

Angular and AngularJS are two different frameworks. AngularJS (also known as Angular 1.x) is the original version of the framework, while Angular is a complete rewrite of AngularJS. Angular is faster, more feature-rich, and has a more modern architecture, while AngularJS is simpler and has a smaller learning curve. Angular is a more powerful framework overall, but AngularJS is still widely used for simple applications.

9. What is the difference between reactive and template-driven forms in Angular?

Reactive forms and template-driven forms are two different approaches to handling forms in Angular. Reactive forms use a programmatic approach to building and managing forms, while template-driven forms rely on Angular's template syntax to define the form structure. Reactive forms provide more control over the form and its validation logic, while template-driven forms are easier to get started with and are better suited for simple forms.

10. What is RxJS and how is it used in Angular?

RxJS is a library for reactive programming in JavaScript. It is used in Angular to handle asynchronous data streams and provide an efficient and elegant way to handle events, such as user interactions, HTTP requests, and timer events. RxJS makes it easy to manage complex data flows and provides a flexible and powerful way to handle asynchronous data in Angular applications.

11. What is the role of Angular Modules in an Angular application?

Angular Modules are used to organize the components, directives, and services in an Angular application into cohesive blocks of functionality. Each module exports a set of declarations that are used by the components and services within the module, and also



imports the required dependencies from other modules. This allows for easy separation of concerns and better maintainability of the codebase.

12. What is the role of the @NgModule decorator in Angular?

The @NgModule decorator is used to define an Angular module in a TypeScript class. It allows you to specify the components, directives, and services that belong to the module, as well as any imports and exports that the module requires. The @NgModule decorator is essential for creating modular, maintainable, and reusable code in Angular applications.

13. What is the difference between Components and Directives in Angular?

Components and Directives are two different types of building blocks in Angular. Components are the basic building blocks of an Angular application and are used to define the UI and behavior of a part of the application. Directives are used to extend the behavior of an HTML element, and are used to add dynamic behavior to the HTML elements. Components are a type of directive with a template, while Directives can be used without a template.

14. What is the difference between Observables and Promises in Angular?

Observables and Promises are two different approaches to handling asynchronous data in Angular. Promises are a one-time representation of the result of an asynchronous operation, while Observables are a stream of data that can be subscribed to and provide notifications over time. Observables are more flexible and powerful than Promises, and are used extensively in Angular for handling asynchronous data streams, such as user interactions and HTTP requests.

15. What is the role of the RouterModule in Angular?

The RouterModule is used to configure and manage the navigation between different parts of an Angular application. It provides a powerful set of features for defining routes, handling navigation, and passing data between components. The RouterModule is an essential part of building large, complex Angular applications, and is used to provide a seamless navigation experience for the end-user.

16. What is the difference between one-way data binding and two-way data binding in Angular?

In one-way data binding, data is passed from the component to the template, but changes in the template are not reflected back in the component. In two-way data binding, changes in both the component and the template are reflected in each other in real-time. This is achieved by using the ngModel directive in combination with a property binding in Angular.

17. What is lazy loading in Angular and why is it important?

Lazy loading is a design pattern in Angular that allows for deferring the loading of parts of the application until they are actually needed. This helps to reduce the initial load time of the application and improve performance. Lazy loading is important in Angular because it allows for building large, complex applications that can still perform well and provide a good user experience.

18. What is a guard in Angular and how is it used?

A guard in Angular is a type of service that is used to control access to routes in the application. Guards can be used to implement security or authorization logic, or to prevent navigation to certain routes based on certain conditions. Guards are implemented as classes that implement the CanActivate, CanDeactivate, or CanLoad interfaces, and they can be added to routes using the canActivate, canDeactivate, or canLoad properties of the route configuration object.

19. What is dependency injection in Angular and how does it work?

Dependency injection is a design pattern in Angular that allows for decoupling the creation of objects from their usage. In Angular, dependencies are declared in the constructor of a class and are automatically provided by the Angular injector at runtime. This allows for writing modular, reusable code, and makes it easier to unit test components and services in isolation.

20. What is AOT (Ahead-of-Time) compilation in Angular and why is it important?

AOT (Ahead-of-Time) compilation is a process in Angular that compiles the application at build time, rather than at runtime. This results in faster application load times and improved performance, because the application code is already compiled and optimized when it is loaded in the browser. AOT compilation is also more secure, because it eliminates the risk of code injection attacks. AOT compilation is important in Angular because it allows for building high-performance, secure, and scalable applications.

21. What is a Directive in Angular and how is it different from a Component?

A Directive in Angular is a reusable component that adds behavior to an existing DOM element. Directives are used to manipulate the DOM and can be used to create custom HTML elements, change the appearance and behavior of existing elements, or add event listeners. Components are a type of directive, but with a template and defined behavior, that define a reusable UI component. The difference between Directives and Components is that Components have their own templates and styles, while Directives don't have a template, but manipulate an existing element.

22. What is the role of the NgModule in Angular and why is it important?

The NgModule is a mechanism in Angular that allows for organizing an application into cohesive blocks of functionality. An NgModule defines a compilation context for the components and services that it declares, and also specifies the imported and exported modules

 and providers. The ngModule is important in Angular because it provides a way to structure an application into reusable and maintainable modules, and it also provides a way to manage dependencies and the lifecycle of the application.

23. What is the role of the RouterModule in Angular and how does it work?

The RouterModule is a module in Angular that provides routing functionality for the application. The RouterModule is used to configure and activate the Angular router, which is responsible for navigating between routes and displaying the appropriate component for the active route. The RouterModule uses a configuration object to define the available routes, and it uses the routerLink directive and the Router service to navigate between routes.

24. What is the difference between a Template-Driven Form and a Reactive Form in Angular?

A Template-Driven Form is a form in Angular that is built using template directives and two-way data binding. Template-Driven forms are easy to use, but can become complex and difficult to maintain for large forms. A Reactive Form, on the other hand, is a form in Angular that is built using the reactive approach, where form data is represented as a model and can be manipulated programmatically. Reactive forms are more flexible and powerful, but can be more difficult to use and require a better understanding of Angular.

25. What is the difference between an Observable and a Promise in Angular?

An Observable is a stream of values in Angular that can emit multiple values over time, and can also be cancelled. Observables are used for asynchronous programming and are often used for real-time data updates or for handling multiple events. A Promise is a single value in Angular that is returned asynchronously and can only be resolved or rejected once. Promises are often used for performing single async operations and provide a simpler API compared to observables.

26. What is Angular's change detection mechanism and how does it work?

Angular uses a mechanism called change detection to detect and update the values of the components in the application. Change detection runs in a loop and checks the values of the components for changes, and if any changes are detected, it updates the values and re-renders the components. By default, change detection runs on the main UI thread, but can also be run manually or on a separate render thread for performance optimization. Change detection in Angular uses a strategy called OnPush, which means that change detection only runs when an input property of a component changes, or when an event is triggered from the component.

27. What is the purpose of the Angular CLI and what are its features?

The Angular CLI is a command-line interface tool that can be used to create, test, and build Angular applications. The Angular CLI provides a number of features that make it easier to develop Angular applications, such as generating components, services, and modules, running unit tests, and building the application for production. The Angular CLI also provides a way to configure and run development servers, and can be used to add and manage dependencies and packages in an Angular application.

28. What is lazy loading in Angular and why is it important?

Lazy loading is a technique in Angular where the application only loads the necessary components and modules for the current route, and delays loading other components and modules until they are actually needed. This can improve the performance and startup time of an Angular application, as well as reduce the size of the application. Lazy loading is important in Angular because it can lead to a more efficient and responsive application, and can also make it easier to manage the size and complexity of the application.

29. What is the role of the ActivatedRoute in Angular and how is it used?

The ActivatedRoute is a class in Angular that is used to represent the current activated route in the application. The ActivatedRoute provides information about the current route, including the route parameters, query parameters, and the data associated with the route. The ActivatedRoute is used to fetch information about the current route and to display the appropriate component for the route. The ActivatedRoute can also be used to subscribe to changes in the route, and to navigate to other routes programmatically.

30. What is the difference between Angular and React, and which one should you use for your project?

Angular and React are both popular JavaScript libraries for building web applications. Angular is a full-fledged framework, while React is a JavaScript library. Angular provides a complete solution for building web applications, with a strong focus on modularity, testing, and performance, while React is focused on building reusable UI components. The choice between Angular and React depends on the specific requirements and goals of a project. If a project requires a more comprehensive solution, Angular may be the better choice, while if a project requires more flexibility and a focus on UI, React may be a better choice. Ultimately, the best choice will depend on the skills of the development team, the complexity of the project, and the specific requirements of the project. Both Angular and React have large and supportive communities, so finding help and resources for either library should not be an issue.

31. What is the difference between Angular components and directives?

Angular components and directives are both used to extend the functionality of HTML elements in an Angular application. However, components are a higher-level concept in Angular and are used to represent self-contained and reusable parts of the user interface, while directives are a lower-level concept used to extend the behavior of HTML elements. Components have a defined template and are responsible for fetching and displaying data, while directives are used to manipulate the DOM or add behavior to existing elements. Components can also use directives to extend their functionality, but directives cannot use components.

32. What is the purpose of the ngOnInit lifecycle hook in Angular?



The `ngOnInit` lifecycle hook is a method in Angular that is called after the component has been initialized and its data-bound properties have been checked for the first time. The `ngOnInit` lifecycle hook is a good place to perform any additional setup or configuration for the component, such as fetching data from a remote server, or initializing values for the component's properties. The `ngOnInit` lifecycle hook is called once during the lifetime of the component and is called after the constructor of the component has been executed.

33. What is the purpose of the RouterModule in Angular and how is it used?

The `RouterModule` is a module in Angular that provides the necessary services and directives for navigating between different routes in an Angular application. The `RouterModule` is used to configure the routes for the application and to associate each route with a component. The `RouterModule` also provides directives such as the `RouterLink` and `RouterOutlet` directives, which are used to navigate to different routes and to display the components for each route. The `RouterModule` is used in the root module of the application, and can be imported and configured with the routes for the application.

34. What is the difference between two-way data binding and one-way data binding in Angular?

Two-way data binding in Angular refers to the synchronization of data between the model and the view, where changes to the model are reflected in the view and vice versa. One-way data binding refers to the synchronization of data in one direction, either from the model to the view or from the view to the model. In Angular, two-way data binding is achieved using a combination of property binding and event binding, while one-way data binding can be achieved using either property binding or event binding alone. Two-way data binding is useful in cases where the user can interact with the application and update the model, while one-way data binding is useful in cases where the model is being updated in response to changes in the application state.

35. What is the purpose of the HttpClientModule in Angular and how is it used?

The `HttpClientModule` is a module in Angular that provides the `HttpClient` service, which is used to make HTTP requests to a remote server. The `HttpClientModule` is used to configure the `HttpClient` service and make it available for use in an Angular application. The `HttpClientModule` is imported in the root module of the application and can be used in any component in the application to make HTTP requests. The `HttpClient` service is a powerful tool for making asynchronous requests and receiving responses from a remote server, and it is used in Angular to interact with APIs and retrieve data from a server. The `HttpClient` service supports a wide range of HTTP methods, including GET, POST, PUT, DELETE, and more.

36. What is the purpose of the ReactiveFormsModule in Angular and how is it used?

The `ReactiveFormsModule` is a module in Angular that provides a way to build and manage reactive forms. Reactive forms provide a way to build complex forms with a reactive approach, where the form state is managed in the component class and updates are made in real-time as the user interacts with the form. The `ReactiveFormsModule` is used to import the necessary directives and services for building reactive forms, and it should be imported in the module that will use the reactive forms. The `ReactiveFormsModule` provides several classes and APIs for building forms, including the `FormControl`, `FormGroup`, and `FormArray` classes, as well as the `formControl`, `formGroup`, and `formArray` directives.

37. What is the purpose of the ngIf directive in Angular and how is it used?

The `ngIf` directive in Angular is used to conditionally render an element in the template based on the value of a boolean expression. The `ngIf` directive is used to add or remove an element from the DOM based on a condition, and it provides a way to conditionally display content in the template. The `ngIf` directive is used in the template as an attribute on the element to be conditionally rendered, and the expression passed to the `ngIf` directive is evaluated to determine whether the element should be displayed or not. For example, the following code uses the `ngIf` directive to conditionally display a message based on the value of a variable: `<p *ngIf='showMessage'>Hello, Angular!</p>`

38. What is the purpose of the ngFor directive in Angular and how is it used?

The `ngFor` directive in Angular is used to dynamically render a list of items in the template. The `ngFor` directive is used to loop over a list of items and generate HTML elements for each item in the list. The `ngFor` directive is used in the template as an attribute on an element that will be used as a container for the dynamically generated elements, and the expression passed to the `ngFor` directive specifies the list of items to be rendered. For example, the following code uses the `ngFor` directive to display a list of names: `<li *ngFor='let name of names'>{{ name }}</li>`

39. What is the purpose of the ngClass directive in Angular and how is it used?

The `ngClass` directive in Angular is used to dynamically set the CSS class for an element based on a condition. The `ngClass` directive is used to conditionally apply one or more CSS classes to an element based on the value of an expression. The `ngClass` directive is used in the template as an attribute on the element to which the CSS classes should be applied, and the expression passed to the `ngClass` directive specifies the CSS classes to be applied and when they should be applied. For example, the following code uses the `ngClass` directive to apply a 'highlight' class to a text element based on the value of a variable: `<p [ngClass]='{ highlight: isHighlighted }'>Hello, Angular!</p>`

40. What is the purpose of the ngStyle directive in Angular and how is it used?

The `ngStyle` directive in Angular is used to dynamically set the styles for an element based on a condition. The `ngStyle` directive is used to conditionally apply one or more styles to an element based on the value of an expression. The `ngStyle` directive is used in the template as an attribute on the element to which the styles should be applied, and the expression passed to the `ngStyle` directive specifies the

 styles to be applied and when they should be applied. For example, the following code uses the `ngStyle` directive to apply a font-size style to a text element based on the value of a variable: `<p [ngStyle]='{ "font-size": fontSize + "px" }'>Hello, Angular!</p>`

41. What is the purpose of the `HttpClientModule` in Angular and how is it used?

The `HttpClientModule` in Angular is used to make HTTP requests from the application to a server. The `HttpClientModule` provides a way to make HTTP requests using the `HttpClient` service, which is a powerful and flexible way to communicate with a server. The `HttpClientModule` should be imported in the module that will use the `HttpClient` service, and once it is imported, the `HttpClient` service can be injected into the components that need to make HTTP requests. For example, the following code imports the `HttpClientModule` and uses the `HttpClient` service to make a GET request to a server.

42. What is the purpose of the `FormsModule` in Angular and how is it used?

The `FormsModule` in Angular is used to create forms in the application. The `FormsModule` provides a way to create template-driven forms or reactive forms, which are forms that are driven by the data model and not by the template. The `FormsModule` should be imported in the module that will use forms, and once it is imported, the forms can be created using the directives and services provided by the `FormsModule`. For example, the following code imports the `FormsModule` and creates a template-driven form in a component.

43. What is the purpose of the `ReactiveFormsModule` in Angular and how is it used?

The `ReactiveFormsModule` in Angular is used to create reactive forms in the application. The `ReactiveFormsModule` provides a way to create forms that are driven by the data model and not by the template. The `ReactiveFormsModule` should be imported in the module that will use reactive forms, and once it is imported, the reactive forms can be created using the `FormBuilder` service and the `FormControl`, `FormGroup`, and `FormArray` classes provided by the `ReactiveFormsModule`. For example, the following code imports the `ReactiveFormsModule` and creates a reactive form in a component.

44. What is lazy loading in Angular?

Lazy loading is a feature in Angular that allows you to load modules only when they are needed, rather than loading all modules at once when the application starts. This can significantly improve the application's startup time and reduce the memory footprint. In Angular, you can implement lazy loading by using the `loadChildren` property in the routing configuration. The `loadChildren` property should be set to the module's path and the module should be loaded using the `import` statement, wrapped in a function that returns a promise.

45. What is the purpose of the `ngOnInit` method in Angular?

The `ngOnInit` method is a lifecycle hook in Angular that is called after the component has been initialized and its data-bound properties have been checked for the first time. This is a good place to perform any additional setup for the component, such as fetching data from a remote server or setting up subscriptions. The `ngOnInit` method is called once and only once during the component's lifetime, making it a good place to perform one-time setup logic.

46. What is the difference between `ngFor` and `ngForOf` in Angular?

The `ngFor` and `ngForOf` are two similar directives in Angular for rendering a list of items. The main difference between them is that `ngFor` is a legacy directive, while `ngForOf` is a more recent addition to Angular. The recommended way of using these directives is to use the `ngForOf` directive, which is more flexible and has better performance. The `ngForOf` directive takes an expression that should evaluate to an array or an iterable, while the `ngFor` directive only takes an array. The `ngForOf` directive is also more flexible, as it can be used with any object that implements the `Iterable` or `Array`-like interface.

47. What is the role of a service in Angular?

A service in Angular is a class that provides a specific functionality that can be shared across multiple components. Services are typically used to encapsulate logic that is not specific to a single component, such as data access, logging, or business logic. Services can be injected into components using dependency injection, allowing the component to access the service's methods and properties. Services can also be injected into other services.

48. What is the purpose of the `RouterModule` in Angular?

The `RouterModule` is a module in Angular that provides routing functionality to an application. It allows you to configure the routes for your application, map the URL to a component, and navigate between different components. The `RouterModule` provides directives such as `routerLink` and `routerOutlet`, which are used to build navigation links and display the content of a routed component, respectively. The `RouterModule` also provides services such as the `Router` service, which is used to programmatically navigate between routes. The `RouterModule` is typically imported in the root module of an application, allowing the entire application to use routing.

49. What is the purpose of the `HttpClientModule` in Angular?

The `HttpClientModule` is a module in Angular that provides a simple way to make HTTP requests. It is a lower-level API for making HTTP requests, compared to higher-level abstractions such as the Angular `Http` module. The `HttpClientModule` provides the `HttpClient` service, which is a simple way to make HTTP requests using the `GET`, `POST`, `PUT`, `DELETE`, and other HTTP methods. The `HttpClientModule` also provides utility functions for transforming data, such as `JSON.stringify`, and error handling. The `HttpClientModule` is typically imported in the root module of an application, allowing the entire application to use the `HttpClient` service.

50. What is the purpose of the `FormsModule` in Angular?



The FormsModule is a module in Angular that provides two-way data binding and validation for forms. It provides directives such as ``ngModel`` and ``ngSubmit``, as well as services such as the FormBuilder and Validators services, which are used to build and validate forms in Angular. The FormsModule also provides the ``ngForm`` directive, which is used to bind a form to the component, and the ``ngControl`` directive, which is used to bind form controls to the component. The FormsModule is typically imported in the module where the forms are used, allowing components in that module to use the forms functionality.

51. What is the purpose of the ReactiveFormsModule in Angular?

The ReactiveFormsModule is a module in Angular that provides a reactive approach to building and validating forms. It provides a way to build and manage form controls and form groups using a functional approach, as opposed to the template-driven approach provided by the FormsModule. The ReactiveFormsModule provides the FormControl, FormGroup, and FormBuilder classes, which are used to build and manage reactive forms in Angular. The ReactiveFormsModule is typically imported in the module where the reactive forms are used, allowing components in that module to use the reactive forms functionality.

52. What is the purpose of the BrowserModule in Angular?

The BrowserModule is a module in Angular that is required for applications that run in a web browser. It provides services and directives that are specific to running Angular in a web browser, such as the Title service, which is used to set the document title, and the Meta service, which is used to set metadata on the document. The BrowserModule also provides the ``NgModule`` decorator, which is used to define an Angular module, and the ``ngDoBootstrap`` function, which is used to bootstrap the root component of an application. The BrowserModule is typically imported in the root module of an application, allowing the entire application to use the services and directives provided by the BrowserModule.

53. What is the purpose of the HttpClientModule in Angular?

The HttpClientModule is a module in Angular that provides a way to make HTTP requests. It provides the HttpClient service, which is a powerful and flexible way to make HTTP requests in Angular. The HttpClient service provides methods for making GET, POST, PUT, DELETE, and other types of HTTP requests, as well as methods for manipulating HTTP headers, URL parameters, and request bodies. The HttpClientModule also provides the ``HttpClient`` interceptor system, which is used to perform tasks such as error handling and request/response transformation before and after HTTP requests are made. The HttpClientModule is typically imported in the module where HTTP requests are made, allowing components in that module to use the HttpClient service.

54. What is the purpose of the RouterModule in Angular?

The RouterModule is a module in Angular that provides routing functionality for applications. It provides the Router service, which is used to navigate between routes and components, as well as the RouterLink and RouterOutlet directives, which are used to bind links to routes and render routes in the component template, respectively. The RouterModule also provides the ``Routes`` array, which is used to define the routes for an application, and the ``forRoot`` method, which is used to configure the RouterModule for a module. The RouterModule is typically imported in the module where routing is used, allowing components in that module to use the routing functionality.

55. What is a component in Angular and what are its main parts?

A component in Angular is a class that defines a reusable UI element, such as a button, a form, or a section of a page. A component consists of three main parts: a component class, a component template, and component-specific metadata. The component class defines the behavior of the component, such as what happens when a button is clicked. The component template defines the HTML structure of the component and how it is rendered. The component-specific metadata, which is defined using the `@Component` decorator, ties the component class and template together and provides additional information about the component, such as the selector used to render the component in the HTML, the styles used to style the component, and the inputs and outputs used to pass data between components.

56. What is a directive in Angular and what are the different types of directives?

A directive in Angular is a class that adds behavior to an existing DOM element or an element created by a component. There are three types of directives in Angular: component directives, attribute directives, and structural directives. Component directives are the most common type of directive and are used to create components. Attribute directives modify the appearance or behavior of an existing element. Structural directives modify the structure of the DOM by adding or removing elements. Examples of structural directives are ``ngIf`` and ``ngFor``. Directives are used to extend the functionality of components and to create reusable UI elements.

57. What is an Observable in Angular and why do we use it?

An Observable in Angular is a stream of data that can emit multiple values over time. Observables are used in Angular to handle asynchronous data, such as network requests, user input, and timer events. Observables provide a way to subscribe to and receive data from the stream, and they also provide a way to handle errors and completion events. Observables are used in Angular because they make it easy to handle asynchronous data and provide a way to react to changes in the data over time.

58. What is the difference between Angular and AngularJS?

Angular and AngularJS are two different frameworks for building web applications. AngularJS is the original version of the framework and was released in 2010. Angular is a complete rewrite of AngularJS and was first released in 2016. Angular is a more powerful and flexible framework than AngularJS, with improved performance, a simpler syntax, and a more modular architecture. Angular also supports mobile

✂ devices and progressive web applications out-of-the-box. While AngularJS applications can be upgraded to Angular, the process is not trivial and typically involves a significant amount of code changes.

59. What is a directive in Angular and how does it work?

A directive in Angular is a custom attribute or element that can modify the behavior and appearance of an HTML element. Directives are used in Angular to add custom logic to the template. Directives can be used to create reusable components, implement dynamic behavior, and bind data to elements in the template. Directives are implemented as classes in Angular, and they can access the DOM, the component instance, and other directives on the element. Directives work by extending the HTML with custom functionality, and they can be used to manipulate the DOM, add event handlers, and interact with the component data and logic.

60. What is a service in Angular and what is it used for?

A service in Angular is a class that provides reusable and independent functionality to other parts of the application, such as components. Services are used in Angular to centralize common logic and data that can be used across multiple components. Services can access data from a back-end API, perform complex calculations, and share data between components. Services are created using the Angular dependency injection system, and they can be injected into components and other services. Services are used in Angular to make the application more modular, testable, and scalable.

61. What is the RouterModule in Angular and how does it work?

The RouterModule in Angular is a module that provides the routing functionality for the application. The RouterModule allows you to define routes for different parts of the application and map them to components. When a user navigates to a specific route, the RouterModule matches the route to the corresponding component and displays it in the browser. The RouterModule also provides navigation methods, such as navigate and navigateByUrl, that can be used to programmatically navigate between routes. The RouterModule uses observables and the location service to implement the routing functionality and provide a seamless user experience.

62. What is AOT compilation in Angular and why is it important?

AOT (Ahead-of-Time) compilation in Angular is a process of compiling the application at build time, before the application is run in the browser. AOT compilation allows Angular to optimize the application for faster rendering, smaller download size, and improved security. AOT compiled applications start faster and run smoother, because the compiled templates and components are included in the application bundle and do not need to be recompiled at runtime. AOT compiled applications are also more secure, because the compiled templates and components cannot be tampered with or executed with malicious code at runtime. AOT compilation is important in Angular because it provides a better user experience and improved security for the application.

Tip: When answering Angular questions, mention module/lazy-loading strategy, change detection strategy (Default vs. OnPush), and how you manage state (services, signals, RxJS, or NgRx).

SEARCH

Search topics...

Go

INTERVIEW TOPICS

Java	Q&A
Java 8	Q&A
Spring	Q&A
Hibernate	Q&A
React JS	Q&A
SQL	Q&A
MongoDB	Q&A
JavaScript	Q&A
Kafka	Q&A
Design Patterns	Q&A

 Microservices	Q&A
REST API	Q&A
Behavioral	Q&A

QUICK ACTIONS

- Start System Design
- Practice Coding
- Earn a Certificate

RECOMMENDED READING

[Learn Java 17 Programming](#)

[Learn JavaScript Quickly](#)

[SQL QuickStart Guide](#)

[System Design Interview](#)

See all →

SPONSORED

Ads help keep content free.

NEED HELP?

Have a topic request or correction? We'd love to hear from you.

Contact Us

THANK YOU!

Your comments and complements are always the inspiration that keeps us motivated.

ABOUT

[PROJECTS](#)

[ABOUT US](#)

[BLOG](#)

[AWARDS](#)

ADDRESS



Tech Venture

 Sterling, Virginia, 20166, US

 info@techventurellc.com

 641 451 3463

 641 451 3463

FOLLOW US

