100+ Most Asked Angular QnA

Made By:

CommonJobs

1. **What is Angular?**
   Angular is a popular open-source JavaScript framework developed by Google for building dynamic web applications.

2. **What are the key features of Angular?**
   The key features of Angular include two-way data binding, dependency injection, modular architecture, component-based development, and a powerful template system.

3. **What is the latest version of Angular?**
   As of my knowledge cutoff in September 2021, the latest version of Angular is Angular
   12. However, please note that there may be newer versions available beyond that.

4. **What is the difference between AngularJS and Angular?**
   AngularJS, also known as Angular 1.x, is an older version of Angular. Angular, starting from version 2, is a complete rewrite of AngularJS with improved performance and enhanced features.

5. **What is TypeScript?**
   TypeScript is a superset of JavaScript that adds static typing and other advanced features to JavaScript. Angular is written in TypeScript.

6. **What is the purpose of the "ng-app" directive in AngularJS?**
   The "ng-app" directive is used to define the root element of an AngularJS application.
   It initializes the application and auto-bootstraps the AngularJS framework.

7. **What is a component in Angular?**
   A component in Angular is a reusable building block that encapsulates the template,
   logic, and styling required to render a part of the user interface.

8. **What is a template in Angular?**
   A template in Angular is an HTML file that defines the structure and layout of a component. It includes placeholders and binding expressions to dynamically display
   data.

9. **What is data binding in Angular?**
   Data binding is a mechanism in Angular that establishes a connection between the component and the template, allowing the synchronization of data between them.

10. **What are the different types of data binding in Angular?**
    The different types of data binding in Angular are:

11. Interpolation ({{}})

12. Property binding ([...])

a component from an external source. It helps in creating loosely coupled and testable code.

16. **Explain the Angular module and its purpose.**

An Angular module is a mechanism to organize the application into cohesive blocks of functionality. It encapsulates components, services, directives, and other features required for a specific part of the application.

17. **What is the purpose of NgModule in Angular?**

@NgModule is a decorator that defines a class as an Angular module. It specifies the metadata required to configure the module and its dependencies.

18. **What is a service in Angular?**

A service in Angular is a reusable singleton object that encapsulates a specific functionality and can be injected into components or other services.

19. **What is the difference between a component and a directive in Angular?**

A component is a type of directive with a template, while a directive doesn't necessarily have a template and can be used to extend the behavior of existing elements.

20. **What is Angular CLI?**

Angular CLI (Command Line Interface) is a powerful command-line tool that helps in initializing, developing, and maintaining Angular applications. It provides various commands to generate components, services, modules, etc.

21. **What is the purpose of the "ngFor" directive in Angular?**

The "ngFor" directive is used to iterate over a collection of items in Angular and generate the corresponding HTML elements for each item.

22. **What is the purpose of the "ngIf" directive in Angular?**

The "ngIf" directive is used to conditionally display or hide elements based on a given expression. It adds or removes elements from the DOM based on the truthiness of the expression.

23. **What is Angular Router?**

Angular Router is a built-in module in Angular that provides navigation and routing functionality. It allows developers to create single-page applications with multiple views and handle navigation between them.

24. **What is lazy loading in Angular?**

Lazy loading is a technique in Angular where modules are loaded on-demand when the user navigates to a specific route. It helps in improving the initial loading time of the application by loading only the necessary modules.

25. **What is Angular Forms?**

Angular Forms is a set of features and techniques used to handle and validate user input in forms. It provides both template-driven forms and reactive forms approaches.

26. **What is the difference between template-driven forms and reactive forms?**

Template-driven forms rely on directives in the template to create and handle forms, while reactive forms use explicit form controls and reactive programming to manage form data and validation.

27. **What is Angular HTTP Client?**

Angular HTTP Client is a module that provides an API to make HTTP requests to a server. It simplifies the process of communicating with a backend and handling responses.

28. **What is Angular Interceptor?**
   Angular Interceptor is a feature that allows you to intercept HTTP requests or responses before they are sent or received. It is used to modify or handle requests globally in an application.

29. **What are Angular pipes?**
   Angular pipes are a feature that transforms data before displaying it in the template. They are used for tasks such as formatting dates, numbers, and strings, as well as creating custom transformations.

30. **What is Angular testing and what are its different levels?**
   Angular testing involves writing tests to verify the behavior and functionality of Angular components, services, and other parts of the application. It includes unit testing, integration testing, and end-to-end testing.

31. **What is TestBed in Angular testing?**
   TestBed is an Angular utility that creates a testing module for configuring and initializing the environment for unit testing Angular components.

32. **What is Angular CLI command to generate a new component?**
   The Angular CLI command to generate a new component is:

33. Copy code

34. ng generate component component-name

35. **What is Angular CLI command to generate a new service?**
   The Angular CLI command to generate a new service is:

36. Copy code

37. ng generate service service-name

38. **What is Angular CLI command to generate a new module?**
   The Angular CLI command to generate a new module is:

39. arduinoCopy code

40. ng generate   module   module   -name

41. **What is Angular CLI command to start the development server?**
   The Angular CLI command to start the development server is:

42. Copy code

43. ng serve

44. **What is Angular CLI command to build the application for production?**
   The Angular CLI command to build the application for production is:

45. cssCopy code

46. ng build   --prod

47. **What is Angular CLI command to run unit tests?**
   The Angular CLI command to run unit tests is:

48. bashCopy code

49. ng   test

50. **What is Angular CLI command to run end-to-end tests?**
   The Angular CLI command to run end-to-end tests is:

51. Copy code

52. ng e2e

53. **What is Angular Material?**
   Angular Material is a UI component library for Angular applications. It provides a set of pre-built and customizable UI components following the Material Design principles.

54. **What is Angular Universal?**
   Angular Universal is a server-side rendering (SSR) solution for Angular applications. It

allows the application to be rendered on the server and sent as fully rendered HTML to the client, improving performance and SEO.

55. **What is Angular Ivy?**

Angular Ivy is the next-generation rendering engine and compiler introduced in Angular version 9. It offers improved performance, smaller bundle sizes, and better debugging and build times.

56. **What is Angular change detection?**

Angular change detection is a mechanism that detects and propagates changes made to data in an application. It automatically updates the view to reflect the updated data.

57. **What are Angular lifecycle hooks?**

Angular lifecycle hooks are methods provided by Angular that allow you to tap into different stages of a component's lifecycle, such as initialization, change detection, and destruction. Examples include ngOnInit, ngOnChanges, and ngOnDestroy.

58. **What is AOT (Ahead-of-Time) compilation in Angular?**

AOT compilation is a process in Angular where the application's templates and components are compiled during the build phase rather than at runtime. It results in faster rendering and smaller bundle sizes.

59. **What is JIT (Just-in-Time) compilation in Angular?**

JIT compilation is the default compilation mode in Angular, where the templates and components are compiled at runtime in the browser. It offers a faster development cycle but slightly slower performance compared to AOT.

60. **What is the purpose of the "async" pipe in Angular?**

The "async" pipe in Angular is used to subscribe to an Observable or Promise in the template and automatically update the view with the emitted values.

61. **What is ngZone in Angular?**

ngZone is a service in Angular that provides a way to execute code outside or inside the Angular zone. It helps in managing change detection and optimizing performance.

62. **What is a decorator in Angular?**

A decorator in Angular is a function that modifies a class, property, method, or parameter by extending or adding behavior to it. Decorators are used extensively in Angular to configure and enhance various parts of the application.

63. **What is tree shaking in Angular?**

Tree shaking is a technique used in Angular (enabled by the underlying TypeScript and webpack) to eliminate unused code from the final bundle during the build process. It helps in reducing the bundle size and improving performance.

64. **What are Angular schematics?**

Angular schematics are command-line tools provided by the Angular CLI that automate the process of generating and modifying code in an Angular application. They can be used to generate components, modules, services, and more.

65. **What is ng-container in Angular?**

ng-container is a special element in Angular that provides a way to group multiple elements without adding an extra element to the DOM. It is often used in conjunction with structural directives.

66. **What is ng-content in Angular?**

ng-content is a placeholder element in Angular that allows the insertion of content

from the parent component into a child component. It is used to create reusable components with customizable content.

67. **What is Angular Renderer?**
Angular Renderer is an API that provides a way to manipulate the DOM directly in Angular. It is used when there is a need to modify the DOM that is not covered by Angular's declarative templates.

68. **What is ViewEncapsulation in Angular?**
ViewEncapsulation is a feature in Angular that encapsulates the styles of a component to prevent them from leaking and affecting other components. It provides different encapsulation modes like Emulated, Native, and None.

69. **What is zone.js in Angular?**
zone.js is a library used by Angular to provide a zone-based execution context. It helps in tracking asynchronous operations and triggering change detection when needed.

70. **What is the purpose of the "trackBy" function in Angular ngFor?**
The "trackBy" function is used in conjunction with the ngFor directive in Angular to improve the performance of rendering lists. It provides away to uniquely identify and track items in the collection, allowing Angular to optimize the rendering process by reusing existing DOM elements instead of recreating them.

71. **What are Angular decorators like @ViewChild and @ContentChild used for?**
Decorators like @ViewChild and @ContentChild are used to access child elements or components in Angular. @ViewChild is used to access a single child component or element, while @ContentChild is used to access projected content within a component.

72. **What is ng-template in Angular?**
ng-template is a directive in Angular that defines a template block that can be conditionally rendered or used as a template for other structural directives like ngIf and ngFor. It allows for greater flexibility in rendering dynamic content.

73. **What is the purpose of the "HostListener" decorator in Angular?**
The "HostListener" decorator is used to add event listeners to the host element of a component in Angular. It allows the component to listen to and react to events raised on the host element.

74. **What is the purpose of the "HostBinding" decorator in Angular?**
The "HostBinding" decorator is used to bind a property of a component to a property of its host element. It allows the component to modify or reflect the state of the host element.

75. **What is the purpose of the "@Injectable" decorator in Angular?**
The "@Injectable" decorator is used to annotate a service class in Angular. It allows the class to be injected with dependencies and enables the Angular dependency injection system to create and provide instances of the service.

76. **What is the purpose of the "ngStyle" directive in Angular?**
The "ngStyle" directive is used to dynamically apply styles to an element based on the values of expressions in the component. It allows for dynamic styling without directly manipulating the CSS classes.

77. **What is the purpose of the "ngClass" directive in Angular?**
The "ngClass" directive is used to conditionally apply CSS classes to an element based on the values of expressions in the component. It allows for dynamic class binding.

78. **What is the purpose of the "ngModel" directive in Angular?**
The "ngModel" directive is used for two-way data binding between a form input element and a component property. It allows the component to get and set the value of the input element.

79. **What is the purpose of the "ngSwitch" directive in Angular?**
The "ngSwitch" directive is used to conditionally render content based on the value of an expression. It allows for multiple cases and provides an alternative to nested ngIf statements.

80. **What is the purpose of the "ng-container" directive in Angular?**
The "ng-container" directive is a structural directive that acts as a grouping element without rendering any additional element to the DOM. It is often used to apply structural directives to multiple elements.

81. **What is the purpose of the "ngZone" service in Angular?**
The "ngZone" service provides a way to execute code within or outside the Angular zone. It is used to handle change detection and trigger Angular's rendering cycle.

82. **What is the purpose of the "async" pipe in Angular?**
The "async" pipe in Angular is used to subscribe to an asynchronous data source, such as an Observable or Promise, and automatically update the view with the emitted values.

83. **What is the purpose of the "ng-content" directive in Angular?**
The "ng-content" directive is used to project content from a parent component into a child component. It allows for dynamic composition of components and flexible content insertion.

84. **What is Angular Material CDK?**
Angular Material CDK (Component Dev Kit) isa set of tools and utilities provided by Angular Material that helps in building custom, reusable UI components. It provides features like drag and drop, virtual scrolling, overlays, and more.

85. **What is Angular Ivy Renderer?**
Angular Ivy Renderer is the rendering engine introduced with Angular Ivy. It is responsible for transforming the component template into executable code, optimizing rendering performance, and enabling features like incremental DOM updates.

86. **What is the purpose of the "ngOnInit" method in Angular?**
The "ngOnInit" method is a lifecycle hook in Angular that is called once, after the component has been initialized and its inputs have been bound. It is commonly used to perform initialization tasks.

87. **What is the purpose of the "ngOnChanges" method in Angular?**
The "ngOnChanges" method is a lifecycle hook in Angular that is called when one or more input properties of a component change. It allows the component to respond to changes in input values.

88. **What is the purpose of the "ngDoCheck" method in Angular?**
The "ngDoCheck" method is a lifecycle hook in Angular that is called during every change detection cycle. It is used to implement custom change detection logic and perform manual checks for changes.

89. **What is the purpose of the "ngAfterViewInit" method in Angular?**
The "ngAfterViewInit" method is a lifecycle hook in Angular that is called after the component's view has been fully initialized. It is used to perform tasks that require access to the rendered DOM elements.

90. **What is the purpose of the "ngAfterViewChecked" method in Angular?**
The "ngAfterViewChecked" method is a lifecycle hook in Angular that is called after every check of the component's view. It is used to perform tasks that need to be executed after the view has been checked for changes.

91. **What is the purpose of the "ngOnDestroy" method in Angular?**
The "ngOnDestroy" method is a lifecycle hook in Angular that is called just before a component is destroyed and removed from the DOM. It is used to perform cleanup tasks and unsubscribe from subscriptions.

92. **What is the purpose of the "ngContent" selector in Angular?**
The "ngContent" selector is used in the template of a component to define a content projection slot. It allows the component to accept and render content provided by its parent component.

93. **What is Angular Ivy's differential loading?**
Angular Ivy's differential loading is a feature that generates different bundles for modern browsers and older browsers. Modern browsers receive a smaller bundle with ES2015+ syntax, while older browsers receive a larger bundle transpiled to ES5 syntax.

94. **What is the purpose of the "ngZone.runOutsideAngular()" method in Angular?**
The "ngZone.runOutsideAngular()" method is used to run a specific code block outside the Angular zone. It helps to prevent unnecessary change detection cycles and improves the performance of the code executed within the block.

95. **What is the purpose of the "NoopAnimationsModule" module in Angular?**
The "NoopAnimationsModule" module is a module provided by Angular that disables animations in Angular Material components. It is useful in scenarios where animations are not desired or need to be turned off for testing purposes.

96. **What is the purpose of the "@Host" decorator in Angular?**
The "@Host" decorator is used in Angular to restrict the dependency resolution to only the host component, excluding any parent components. It ensures that the dependency is obtained from the immediate host component.

97. **What is the purpose of the "preserveWhitespaces" configuration in Angular templates?**
The "preserveWhitespaces" configuration in Angular templates is used to control thehandling of whitespace characters in the template. When set to "true", it preserves all whitespace characters, including line breaks and spaces. When set to "false" or not specified, it removes unnecessary whitespace characters to minimize the size of the rendered HTML.

98. **What is Angular Material Theming?**
Angular Material Theming is a feature that allows customization of the visual appearance and style of Angular Material components. It provides a way to define custom color palettes, typography, and other style attributes to create a unique look and feel for the application.

99. **What is the purpose of the "RouterOutlet" directive in Angular?**
The "RouterOutlet" directive is used in Angular to define the location where the router should render the components associated with different routes. It acts as a placeholder for dynamically loaded components.

100. **What is the purpose of the "resolve" property in Angular route configuration?**
The "resolve" property in Angular route configuration is used to specify a set of data to be resolved before activating a route. It allows for fetching data from a server or performing other tasks asynchronously before the route is activated.

101. **What is the purpose of the "CanActivate" guard in Angular?**

The "CanActivate" guard is used in Angular to control access to a route based on certain conditions. It allows for preventing navigation to a route if specific criteria are not met, such as user authentication.

102. **What is the purpose of the "CanDeactivate" guard in Angular?**

The "CanDeactivate" guard is used in Angular to control whether a user can leave a route or component. It allows for prompting the user with a confirmation message or performing other actions before leaving the current route.

103. **What is the purpose of the "CanLoad" guard in Angular?**

The "CanLoad" guard is used in Angular to control whether a module can be loaded lazily. It allows for preventing the loading of a module based on certain conditions, such as user permissions.

104. **What is the purpose of the "ErrorHandler" interface in Angular?**

The "ErrorHandler" interface in Angular is used to define a custom error handler for handling errors that occur in the application. It allows for centralizing error handling logic and providing a consistent error-handling strategy.

105. **What is Angular Ivy's improved tree shaking?**

Angular Ivy's improved tree shaking is a feature that allows for better elimination of unused code during the build process. It analyzes the application's dependencies more accurately, resulting in smaller bundle sizes and improved performance.

106. **What is the purpose of the "trackBy" function in Angular ngFor?**

The "trackBy" function in Angular ngFor is used to optimize the rendering of lists by providing a unique identifier for each item. It helps Angular in identifying changes in the list and updating only the necessary elements in the DOM.

107. **What is the purpose of the "StrictNullChecks" compiler option in TypeScript?**

The "StrictNullChecks" compiler option in TypeScript enforces stricter null and undefined checks during type checking. It helps in catching potential null or undefined errors at compile-time and promotes safer coding practices.

108. **What is Angular's ContentChild decorator used for?**

The ContentChild decorator in Angular is used to obtain a reference to a child component, directive, or template variable projected into the component's view. It allows the component to interact with and access the properties and methods of the projected content.

109. **What is Angular's ContentChildren decorator used for?**

The ContentChildren decorator in Angular is used to obtain a reference to multiple instances of a child component, directive, or template variable projected into the component's view. It allows the component to interact with and access multiple instances of the projected content.

110. **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 inputs have been bound. It is commonly used to perform initialization tasks such as retrieving data from a server or setting up subscriptions.

111. **What is Angular's HttpClient and how is it used?**

Angular's HttpClient is a built-in module that provides an API for making HTTP requests to a server. It supports various HTTP methods such as GET, POST, PUT, DELETE, and provides features like handling request headers, request/response interception, and error handling. It is used by injecting the HttpClient service into a component or service and invoking its methods to perform HTTP operations.

112 What is the purpose of the "ngOnInit" method in Angular forms?

The "ngOnInit" method is a lifecycle hook in Angular that is used in forms to initialize and set up the form controls and validators. It is called after the component has been initialized and is a good place to set the initial values of the form or set up any form-related logic.

113 What is Angular's ActivatedRoute and how is it used?

Angular's ActivatedRoute is a service that provides information about the currently activated route. It contains route parameters, query parameters, data resolved for the route, and other route-related information. It is used by injecting the ActivatedRoute service into a component and accessing its properties and methods to retrieve information about the current route.

114 What is Angular's FormBuilder and how is it used?

Angular's FormBuilder is a utility class that provides a concise way to define and create form controls and form groups. It simplifies the process of creating and managing form controls by providing methods to define form controls with validation rules and default values. It is used by injecting the FormBuilder service into a component and invoking its methods to create form controls and groups.

115 What is Angular's ngClass directive and how is it used?

Angular's ngClass directive is used to conditionally apply CSS classes to an element based on expressions in the component. It allows dynamic class binding by evaluating the expressions and adding or removing CSS classes accordingly. It is used by adding the ngClass directive to an element and providing it with the desired class bindings.

116 What is Angular's ngStyle directive and how is it used?

Angular's ngStyle directive is used to conditionally apply inline styles to an element based on expressions in the component. It allows dynamic style binding by evaluating the expressions and applying the styles accordingly. It is used by adding the ngStyle directive to an element and providing it with the desired style bindings.

117 What is Angular's ngTemplateOutlet directive and how is it used?

Angular's ngTemplateOutlet directive is used to render a template dynamically within the current component's view. It allows the reuse of templates and the dynamic insertion of content. It is used by adding the ngTemplateOutlet directive to an element and providing it with the template reference to be rendered.

118 What is Angular's ng-container directive and how is it used?

Angular's ng-container directive is a grouping element that does not generate any additional DOM element. It is used to apply structural directives to multiple elements without the need for a wrapping element. It is often used in conjunction with ngIf, ngFor, and ngTemplateOutlet to structure the layout and logic of the template.

119 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 inputs have been bound. It is commonly used to perform initialization tasks such as retrieving data from a server or setting up subscriptions.

120. What is Angular's HttpClient and how is it used?

Angular's HttpClient is a built-in module that provides an API for making HTTP requests to a server. It supports various HTTP methods such as GET, POST, PUT, DELETE, and provides features like handling request headers, request/response interception, and error handling. It is used by injecting the HttpClient service into a component or service and invoking its methods to perform HTTP operations.

121. **What is the purpose of the "ngOnInit" method in Angular forms?**

The "ngOnInit" method is a lifecycle hook in Angular that is used in forms to initialize and set up the form controls and validators. It is called after the component has been initialized and is a good place to set the initial values of the form or set up any form-related logic.

122. **What is Angular's ActivatedRoute and how is it used?**

Angular's ActivatedRoute is a service that provides information about the currently activated route. It contains route parameters, query parameters, data resolved for the route, and other route-related information. It is used by injecting the ActivatedRoute service into a component and accessing its properties and methods to retrieve information about the current route.

123. **What is Angular's FormBuilder and how is it used?**

Angular's FormBuilder is a utility class that provides a concise way to define and create form controls and form groups. It simplifies the process of creating and managing form controls by providing methods to define form controls with validation rules and default values. It is used by injecting the FormBuilder service into a component and invoking its methods to create form controls and groups.

124. **What is Angular's ngClass directive and how is it used?**

Angular's ngClass directive is used to conditionally apply CSS classes to an element based on expressions in the component. It allows dynamic class binding by evaluating the expressions and adding or removing CSS classes accordingly. It is used by adding the ngClass directive to an element and providing it with the desired class bindings.

125. **What is Angular's ngStyle directive and how is it used?**

Angular's ngStyle directive is used to conditionally apply inline styles to an element based on expressions in the component. It allows dynamic style binding by evaluating the expressions and applying the styles accordingly. It is used by adding the ngStyle directive to an element and providing it with the desired style bindings.

126. **What is Angular's ngTemplateOutlet directive and how is it used?**

Angular's ngTemplateOutlet directive is used to render a template dynamically within the current component's view. It allows the reuse of templates and the dynamic insertion of content. It is used by adding the ngTemplateOutlet directive to an element and providing it with the template reference to be rendered.

127. **What is Angular's ng-container directive and how is it used?**

Angular's ng-container directive is a grouping element that does not generate any additional DOM element. It is used to apply structural directives to multiple elements without the need for a wrapping element. It is often used in conjunction with ngIf, ngFor, and ngTemplateOutlet to structure the layout and logic of the template.

128. **What is Angular's ViewChild decorator and how is it used?**

Angular's ViewChild decorator is used to access child elements or components in a parent component. It allows the parent component to obtain a reference to a child component or DOM element and interact with it. It isused by adding the ViewChild decorator to a property in the parent component and specifying the selector of the child component or element to be referenced.

129. **What is Angular's ContentChild decorator and how is it used?**

Angular's ContentChild decorator is used to access projected content within a component. It allows the component to obtain a reference to a projected component, directive, or template variable and interact with it. It is used by adding the

ContentChild decorator to a property in the component and specifying the selector of the projected content to be referenced.

130. **What is Angular's ContentChildren decorator and how is it used?**

Angular's ContentChildren decorator is used to access multiple instances of projected content within a component. It allows the component to obtain references to multiple projected components, directives, or template variables and interact with them. It is used by adding the ContentChildren decorator to a property in the component and specifying the selector of the projected content to be referenced.

131. **What is Angular's EventEmitter and how is it used?**

Angular's EventEmitter is a class that provides a way to emit custom events from a component. It allows the component to define custom events and emit them when certain actions or conditions occur. It is used by declaring an EventEmitter property in the component and emitting events using the emit() method.

132. **What is Angular's ngIf directive and how is it used?**

Angular's ngIf directive is used to conditionally render content in the template based on an expression in the component. It allows the component to control the visibility of elements based on certain conditions. It is used by adding the ngIf directive to an element and providing it with the expression to evaluate.

133. **What is Angular's ngFor directive and how is it used?**

Angular's ngFor directive is used to iterate over a collection and generate HTML elements for each item in the collection. It allows the component to dynamically render elements based on the data in the collection. It is used by adding the ngFor directive to an element and providing it with the collection to iterate over.

134. **What is Angular's ngSwitch directive and how is it used?**

Angular's ngSwitch directive is used to conditionally render content based on the value of an expression in the component. It allows the component to choose among multiple templates based on the value of an expression. It is used by adding the ngSwitch directive to a container element and providing it with the expression to evaluate.

135. **What is Angular's template-driven forms approach?**

Angular's template-driven forms approach is a way of building forms using template syntax and directives. It allows the form controls and validation rules to be defined directly in the template, with Angular automatically managing the form state and validation. It is suitable for simpler forms with less complex validation requirements.

136. **What is Angular's reactive forms approach?**

Angular's reactive forms approach is a way of building forms using reactive programming principles. It involves creating form controls and form groups programmatically in the component using the FormBuilder service. It provides more flexibility and control over form validation and handling complex form scenarios.

137. **What is Angular's FormGroup and FormControl classes used for?**

Angular's FormGroup class is used to represent a group of form controls. It is used to create a container for multiple form controls that are related to each other. Angular's FormControl class is used to represent a single form control. It is used to define and manage the value, validation rules, and state of an individual form control.

138. **What is Angular's FormBuilder and how is it used in reactive forms?**

Angular's FormBuilder is a utility class that simplifies the creation of form controls and form groups in reactive forms. It provides methods to define form controls and groups

with validation rules and default values. It is used byinjecting the FormBuilder service into a component and using its methods to create form controls and groups.

139. **What is Angular's Validators class and how is it used?**

Angular's Validators class is a utility class that provides a set of pre-built validators that can be used to validate form controls in Angular forms. It includes validators for required fields, minimum and maximum values, email addresses, and more. Validators are typically used when defining form controls in reactive forms or template-driven forms.

140. **What is Angular's ActivatedRouteSnapshot and how is it used?**

Angular's ActivatedRouteSnapshot is an interface that represents the state of a route at a particular moment in time. It contains information about the route's parameters, data, URL segments, and more. It is used to access the route snapshot in Angular's route guards or resolver services.

141 **What is Angular's ActivatedRouteData and how is it used?**

Angular's ActivatedRouteData is an interface that represents additional data associated with a route. It can be used to store custom data related to a route, such as breadcrumbs, page titles, or any other information that needs to be accessed during route navigation or rendering.

142. **What is Angular's RouterLink directive and how is it used?**

Angular's RouterLink directive is used to create links to navigate between different routes in an Angular application. It automatically updates the URL and activates the corresponding route when the link is clicked. It is used by adding the RouterLink directive to an anchor tag or any other clickable element and providing it with the route path or a link parameters array.

143. **What is Angular's ActivatedRouteLink directive and how is it used?**

Angular's ActivatedRouteLink directive is used to create links that navigate to a specific route based on the current route's context. It allows for relative navigation and generating links dynamically based on the current route's parameters and data. It is used by adding the ActivatedRouteLink directive to an anchor tag or any other clickable element and providing it with the desired route parameters.

144. **What is Angular's RouterOutlet directive and how is it used?**

Angular's RouterOutlet directive is used to define the location where the router should render the components associated with different routes. It acts as a placeholder for dynamically loaded components based on the current route. It is used by adding the RouterOutlet directive to a container element in the template where the routed components should be rendered.

145. **What is Angular's RouterStateSnapshot and how is it used?**

Angular's RouterStateSnapshot is an interface that represents the state of the router at a particular moment in time. It contains information about the current URL, the activated route hierarchy, and the state of the route guards. It is used to access the current router state in Angular's route guards or resolver services.

146. **What is Angular's CanActivate guard and how is it used?**

Angular's CanActivate guard is an interface that defines a guard that controls access to a route based on certain conditions. It allows for preventing navigation to a route if specific criteria are not met, such as user authentication or authorization. It is used by implementing the CanActivate interface in a guard class and providing the guard in the route configuration.

147. **What is Angular's CanDeactivate guard and how is it used?**

Angular's CanDeactivate guard is an interface that defines a guard that controls whether a user can leave a route or component. It allows for prompting the user with a confirmation message or performing other actions before leaving the current route. It is used by implementing the CanDeactivate interface in a guard class and providing the guard in the route configuration.

148. **What is Angular's CanLoad guard and how is it used?**

Angular's CanLoad guard is an interface that defines a guard that controls whether a module canbe loaded lazily. It allows for preventing the loading of a module based on certain conditions, such as user permissions or feature flags. It is used by implementing the CanLoad interface in a guard class and providing the guard in the route configuration.

149. **What is Angular's RouteResolver and how is it used?**

Angular's RouteResolver is an interface that defines a resolver service that retrieves data before activating a route. It allows for fetching data from a server or performing other tasks asynchronously before the route is activated. It is used by implementing the resolve() method in a resolver service class and providing the resolver in the route configuration.

150. **What is Angular's ErrorHandler interface and how is it used?**

Angular's ErrorHandler interface is used to define a custom error handler for handling errors that occur in the application. It allows for centralizing error handling logic and providing a consistent error-handling strategy. It is used by implementing the handleError() method in an error handler class and providing the error handler in the application's dependency injection system.

151. **What is Angular's NoopAnimationsModule and how is it used?**

Angular's NoopAnimationsModule is a module provided by Angular that disables animations in Angular Material components. It is useful in scenarios where animations are not desired or need to be turned off for testing purposes. It is used by importing the NoopAnimationsModule module in the application's module.

152. **What is Angular's BrowserAnimationsModule and how is it used?**

Angular's BrowserAnimationsModule is a module provided by Angular that enables animations in Angular Material components. It is used to animate the transitions and interactions of Angular Material components. It is used by importing the BrowserAnimationsModule module in the application's module.

153. **What is Angular's ViewChild and ViewChildren decorators used for?**

Angular's ViewChild and ViewChildren decorators are used to obtain references to child components or elements in a parent component. ViewChild is used to obtain a reference to a single child component or DOM element, while ViewChildren is used to obtain references to multiple instances of child components or DOM elements. They are used by adding the decorators to properties in the parent component and specifying the selector of the child components or elements to be referenced.

154. **What is Angular's ng-content directive and how is it used?**

Angular's ng-content directive is used to project content from a parent component into a child component. It allows for dynamic composition of components and flexible content insertion. It is used by adding the ng-content directive to the template of the child component and using it as a placeholder for the projected content.

155. **What is Angular's ng-container directive and how is it used?**

Angular's ng-container directive is a grouping element that does not generate any

additional DOM element. It is used to apply structural directives to multiple elements without the need for a wrapping element. It is often used in conjunction with ngIf, ngFor, and ngTemplateOutlet to structure the layout and logic of the template.

156. **What is Angular's ngTemplateOutlet directive and how is it used?**

Angular's ngTemplateOutlet directive is used to render a template dynamically within the current component's view. It allows the reuse of templates and the dynamic insertion of content. It is used by adding the ngTemplateOutlet directive to an element and providing it with the template reference to be rendered.

157. **What is Angular's ng-container directive and how is it used?**

Angular's ng-container directive is a grouping element that does not generate any additional DOM element. It is used to apply structural directives to multiple elements without the need for a wrapping element. It is often used in conjunction with ngIf, ngFor, and ngTemplateOutlet to structure the layout and logic of the template.

158. **What is Angular's ng-template directive and how is it used?**

Angular'sng-template directive is used to define a template block that can be conditionally rendered or used as a template for other structural directives like ngIf and ngFor. It allows for the creation of reusable templates that can be dynamically rendered or applied to elements. It is used by adding the ng-template directive to the template and providing it with a template block to be rendered or used as a template.

159. **What is Angular's ngZone service and how is it used?**

Angular's ngZone service provides a way to execute code within or outside the Angular zone. The Angular zone is an execution context that tracks asynchronous operations and triggers change detection when needed. The ngZone service is used to manage change detection and optimize the performance of the application. It is used by injecting the ngZone service into a component or service and using its run() method to execute code within the Angular zone.

160. **What is Angular's ElementRef and how is it used?**

Angular's ElementRef is a class that represents a reference to a DOM element. It provides access to the underlying DOM element and allows for direct manipulation of its properties and methods. It is used by injecting the ElementRef into a component or directive and accessing its nativeElement property to interact with the DOM element.

161. **What is Angular's Renderer2 and how is it used?**

Angular's Renderer2 is a class that provides a way to manipulate the DOM directly in Angular. It is used when there is a need to modify the DOM that is not covered by Angular's declarative templates. It provides methods for creating, modifying, and removing DOM elements, as well as for manipulating element properties, attributes, and styles. It is used by injecting the Renderer2 service into a component or directive and using its methods to perform DOM manipulations.

162. **What is Angular's Pipe and how is it used?**

Angular's Pipe is a class that allows for the creation of custom data transformation functions that can be applied in templates using the pipe syntax. Pipes are used to format, filter, or transform data before displaying it in the template. They are used by creating a custom pipe class with the @Pipe decorator and implementing the transform() method to define the transformation logic. The pipe can then be used in the template by adding it to the pipe syntax.

163. **What is Angular's Dependency Injection and how is it used?**

Angular's Dependency Injection (DI) is a design pattern and mechanism for providing

dependencies to components, services, and other objects in an application. It allows for the decoupling of components and the reusability of services. In Angular, DI is managed by the Angular injector, which is responsible for creating and providing instances of dependencies. DI is used by specifying dependencies in the constructor of a class and allowing Angular to resolve and inject the dependencies automatically.

164. **What is Angular's NgModule and how is it used?**

Angular's NgModule is a decorator that is used to define an Angular module. An Angular module is a container for a set of related components, directives, services, and other building blocks of an application. The NgModule decorator is used to configure the module by specifying the declarations, imports, providers, and exports of the module. It is used by adding the NgModule decorator to a class and providing the module configuration as its argument.

165. **What is Angular's Router and how is it used?**

Angular's Router is a module that provides a way to implement navigation and routing in an Angular application. It allows for defining routes, navigating between routes, and handling route parameters and query parameters. The Router module is used by importing the RouterModule in the application's module and configuring the routes using the RouterModule.forRoot() method. The Router module is then used by injecting the Router service into components and using its methods to navigate and interact with therouting system.

166. **What is Angular Universal and what is it used for?**

Angular Universal is a server-side rendering (SSR) solution for Angular applications. It allows the application to be rendered on the server and sent as fully rendered HTML to the client, improving performance and SEO. Angular Universal is used to create universal (isomorphic) applications that can be rendered both on the server and the client.

167. **What is Angular's HttpClientInterceptor and how is it used?**

Angular's HttpClientInterceptor is an interface that allows for intercepting and modifying HTTP requests and responses made with the HttpClient module. Interceptors can be used to add headers, handle authentication, modify the request or response payload, or perform other actions. Interceptors are used by implementing the HttpInterceptor interface and registering the interceptor in the application's module or using the providedIn property to automatically provide the interceptor.

168. **What is Angular's ngUpgrade module and when is it used?**

Angular's ngUpgrade module is used to facilitate the gradual migration of an AngularJS (Angular 1.x) application to Angular. It allows for running AngularJS and Angular components side by side in the same application and provides utilities for interoperability between the two frameworks. The ngUpgrade module is used when transitioning from AngularJS to Angular in an existing application. It allows for a step-by-step migration approach without the need to rewrite the entire application at once.