99 Angular interview questions to hire top developers

Questions

1. What is Angular and why do we use it?
2. Explain the difference between Angular and AngularJS.
3. What are components in Angular? Can you give a simple example?
4. What are directives in Angular? Name a few built-in directives.
5. What is data binding in Angular? Explain different types of data binding.
6. What are Angular modules (NgModules)? Why are they important?
7. What is a template in Angular?
8. Explain the concept of routing in Angular. How do you navigate between different views?
9. What is dependency injection (DI) in Angular?
10. What are services in Angular? How do you create and use a service?
11. What is the purpose of the `*ngIf` directive?
12. How do you handle user input in Angular forms?
13. What are Angular pipes? Give an example.
14. Explain the difference between interpolation and property binding.
15. What is the purpose of the `HttpClient` module in Angular?
16. How do you handle events in Angular, such as a button click?
17. What is the role of the `app.module.ts` file?
18. What is the purpose of Angular CLI (Command Line Interface)?
19. Explain what Observables are in Angular. Where have you used them?
20. What are lifecycle hooks in Angular components? Name a few.
21. What is the difference between `constructor` and `ngOnInit`?
22. How would you pass data from a parent component to a child component?
23. How would you make an HTTP GET request in Angular, and display the data?
24. What are the advantages of using Angular compared to traditional JavaScript?
25. What is Angular, in simple terms?
26. Can you name a few advantages of using Angular?
27. What is a component in Angular? Imagine you're building with LEGOs.
28. What is a module in Angular? How does it help organize your app?
29. What is data binding in Angular? Can you give a simple example?
30. What are directives in Angular? Give an example of one you might use.
31. Explain what dependency injection is, using a simple analogy.
32. What is a service in Angular? When would you use one?
33. What is routing in Angular? Why is it useful?
34. What's the difference between interpolation and property binding?
35. What is an Angular template?
36. What is the purpose of the `ngIf` directive?
37. What is the purpose of `ngFor` directive?
38. What is two-way data binding? How does it work?
39. What is the Angular CLI? How can it help you?
40. What are Angular pipes? Give a simple example.
41. What are input and output properties in Angular components?
42. How do you pass data from a parent component to a child component?
43. How do you pass data from a child component to a parent component?
44. What is an observable in Angular? Why are they important?
45. What is a promise? How is it different from an observable?
46. What is HTTP Client in Angular? What is it used for?
47. How do you handle errors in Angular?
48. What are lifecycle hooks in Angular? Name a few.
49. Can you describe a simple Angular project you have worked on, even if it was small?
50. Explain the concept of Angular pipes and their use cases. Can you create a custom pipe?
51. What are Angular directives? Differentiate between structural and attribute directives with examples.
52. Describe Angular lifecycle hooks. When would you use `ngOnChanges` versus `ngOnInit`?
53. How does Angular's dependency injection work? Explain providers, injectors, and tokens.
54. What are Angular modules (NgModule)? How do they help in organizing an Angular application?
55. Explain Angular services. How do you create and inject a service into a component?
56. What is RxJS in the context of Angular? Describe common RxJS operators like `map`, `filter`, and `subscribe`.
57. How does Angular handle form validation? Explain template-driven vs. reactive forms.
58. Explain Angular routing. How do you configure routes and navigate between components?
59. What are Angular interceptors? How can you use them to modify HTTP requests and responses?
60. Describe Angular's change detection mechanism. How can you optimize change detection performance?
61. What are Angular resolvers? How do they help in pre-fetching data before a route is activated?
62. How can you implement lazy loading in an Angular application? What are the benefits?
63. Explain the concept of state management in Angular using NgRx or Akita. What problems do they solve?
64. How do you handle error handling in Angular applications? Explain different strategies.
65. What are Angular animations? How do you create and apply animations to components?
66. Describe Angular Universal. What are the benefits of server-side rendering?
67. How do you test Angular components? Explain unit testing vs. end-to-end testing.
68. What are web workers and how can you leverage them in Angular to improve performance?
69. Explain the concept of content projection in Angular. When would you use it?
70. How do you secure an Angular application? What are common security best practices?
71. What are Angular schematics? How can you use them to automate repetitive tasks?
72. How do you implement internationalization (i18n) in an Angular application?
73. Explain the concept of ahead-of-time (AOT) compilation in Angular. What are the benefits over just-in-time (JIT) compilation?
74. What are the different ways to handle environment-specific configurations in an Angular application?
75. Describe how you would optimize an Angular application for performance. Consider aspects like bundle size, change detection, and lazy loading.
76. Explain how to use Angular CLI to generate components, services, and modules.
77. How would you optimize an Angular application for better performance, focusing on change detection strategies?
78. Explain the differences between various RxJS operators like `switchMap`, `mergeMap`, and `concatMap`, and when would you use each?
79. Describe a situation where you would implement a custom Angular CLI builder.
80. What are the advantages and disadvantages of using Angular Universal for server-side rendering?
81. How would you implement a micro frontend architecture using Angular, and what are the challenges involved?
82. Explain how to implement custom form validation that interacts with a backend service asynchronously.
83. Describe your experience with NgRx or other state management libraries in Angular. What problems did it solve, and what were the trade-offs?
84. How do you handle authentication and authorization in an Angular application, particularly with regards to securing API calls?
85. Explain the concept of 'tree shaking' in Angular and how it contributes to reducing bundle size.
86. Describe your strategy for testing Angular applications, including unit, integration, and end-to-end tests.
87. How would you approach debugging performance issues in a large Angular application?
88. Explain how to create and use dynamic components in Angular.
89. What are Angular schematics and how can they be used to automate repetitive tasks?
90. Describe your experience with accessibility (a11y) in Angular applications and what tools/techniques you use.
91. How do you handle internationalization (i18n) and localization (l10n) in an Angular application?
92. Explain the concept of content projection in Angular and its use cases.
93. How would you implement a custom Angular decorator?
94. Describe the process of creating a custom Angular pipe, including pure and impure pipes.
95. How do you manage different environments (e.g., development, staging, production) in an Angular project?
96. Explain how you would implement a progressive web app (PWA) using Angular.
97. Describe your experience with integrating Angular with other frontend frameworks or libraries.