

93 ASP.NET MVC interview questions to hire top developers

Questions

1. What's the big deal about MVC? Why not just one big code file?
2. Imagine you're showing MVC to a friend. How would you describe each part: Model, View, and Controller?
3. Okay, so a user clicks a button. What part of MVC handles that first?
4. What is Razor view engine? Why is it used in ASP.NET MVC?
5. What's the deal with 'DataAnnotations'? What problems do they solve?
6. What's the purpose of a 'ViewModel' in MVC? Can you give an example?
7. How does routing work in MVC? Can you give a simple example?
8. What are HTML helpers in MVC? Can you give an example of a built-in helper and a custom helper?
9. What is the use of TempData, ViewData, and ViewBag? What are the differences between them?
10. How do you handle errors in an MVC application? What are some best practices?
11. What's the role of the _Layout.cshtml file? Why is it useful?
12. What are Action Filters in MVC? Give a simple example.
13. What is model binding in MVC? How does it simplify your work?
14. How do you validate user input in an MVC application?
15. What are Partial Views? Why would you use them?
16. How can you include JavaScript and CSS files in your MVC application?
17. Explain the difference between strongly typed views and loosely typed views.
18. What is the use of Areas in MVC? When would you use them?
19. Explain what bundling and minification are and how they improve performance.
20. What is the role of the Global.asax file in an MVC application?
21. How do you handle sessions in MVC?
22. How do you implement security in an MVC application? What are some common security threats?
23. What is Dependency Injection (DI) and how is it used in MVC?
24. What are some advantages of using ASP.NET MVC over traditional ASP.NET Web Forms?
25. How would you test an MVC controller?
26. What is ASP.NET MVC, in simple terms, and why do we use it?
27. Can you explain the roles of Model, View, and Controller in MVC?
28. What is the purpose of a View in ASP.NET MVC?
29. How does data flow between the Model, View, and Controller in a typical MVC application?
30. What is the purpose of a Controller in ASP.NET MVC?
31. What is the purpose of a Model in ASP.NET MVC?
32. What is routing in ASP.NET MVC, and why is it important?
33. What is an Action Method in a Controller?
34. Can you describe a scenario where you might use the ViewBag or ViewData to pass data to a View?
35. What are HTML helpers in ASP.NET MVC, and can you give an example?
36. What is the difference between Html.TextBox and Html.TextBoxFor?
37. How do you handle form submissions in ASP.NET MVC?
38. What is client-side validation, and why is it important alongside server-side validation?
39. What are some ways to handle errors or exceptions in an ASP.NET MVC application?
40. Explain the purpose of using layouts (master pages) in ASP.NET MVC.
41. What are partial views, and when would you use them?
42. How would you include CSS and JavaScript files in your ASP.NET MVC project?
43. What is the role of the web.config file in an ASP.NET MVC application?
44. Can you explain what a NuGet package is and why we use it in ASP.NET MVC?
45. If you had to explain the benefits of using ASP.NET MVC to someone who has never heard of it, what would you say?
46. How would you implement custom model validation in ASP.NET MVC, and why is it important?
47. Explain the role of Action Filters in ASP.NET MVC and provide an example of when you might use a custom Action Filter.
48. Describe how you would handle exceptions globally in an ASP.NET MVC application.
49. What are the benefits of using asynchronous controllers in ASP.NET MVC, and how do they work?
50. Explain the difference between TempData, ViewData, and ViewBag in ASP.NET MVC.
51. How can you implement security in an ASP.NET MVC application to protect against common web vulnerabilities like XSS and CSRF?
52. Discuss different approaches to state management in ASP.NET MVC, including cookies, session state, and hidden fields.
53. How do you implement routing in ASP.NET MVC, including custom routes and attribute routing?
54. Explain the purpose of the AntiForgeryToken helper in ASP.NET MVC and how it prevents CSRF attacks.
55. Describe how you would unit test an ASP.NET MVC controller action that interacts with a database.
56. How can you optimize the performance of an ASP.NET MVC application, including caching strategies and bundling/minification?
57. Explain the role of the HTML helpers in ASP.NET MVC and provide examples of when you might use them.
58. Discuss different approaches to dependency injection in ASP.NET MVC, including constructor injection and property injection.
59. How can you implement localization and globalization in an ASP.NET MVC application to support multiple languages and cultures?
60. Explain the difference between authorization and authentication in ASP.NET MVC and how you would implement each.
61. Describe how you would handle file uploads in an ASP.NET MVC application, including validation and storage.
62. How do you use areas in ASP.NET MVC to organize a large application into smaller, more manageable modules?
63. Explain the role of model binders in ASP.NET MVC and how you can create custom model binders.
64. Discuss different approaches to data access in ASP.NET MVC, including Entity Framework and ADO.NET.
65. How can you implement a custom route constraint in ASP.NET MVC to restrict the values that a route parameter can accept?
66. Explain how to implement output caching in ASP.NET MVC and what are the different caching options available?
67. Describe how you would implement a custom membership provider in ASP.NET MVC.
68. How do you use the AuthorizeAttribute in ASP.NET MVC, and can you extend it for complex scenarios?
69. What is the role of the ValidateAntiForgeryToken attribute, and how does it prevent cross-site request forgery (CSRF) attacks?
70. How would you implement a custom filter to log all requests and responses in an ASP.NET MVC application?
71. Explain how you would integrate ASP.NET Web API with an ASP.NET MVC application, and what are the benefits of doing so?
72. How would you implement custom model binding in ASP.NET MVC, and what are the benefits of doing so?
73. Explain the role of action filters in ASP.NET MVC and provide examples of how you would implement custom action filters for specific scenarios (e.g., logging, authorization).
74. Describe your experience with implementing security measures in ASP.NET MVC applications, such as authentication and authorization, and discuss common security vulnerabilities and how to prevent them.
75. How do you handle exceptions and errors in ASP.NET MVC applications, and what strategies do you use for logging and monitoring errors in production environments?
76. Explain your approach to optimizing the performance of ASP.NET MVC applications, including techniques for caching, bundling and minification, and database optimization.
77. Describe your experience with implementing RESTful APIs using ASP.NET MVC Web API and discuss considerations for designing and building scalable and maintainable APIs.
78. How would you implement a custom route constraint in ASP.NET MVC to handle specific routing scenarios?
79. Explain how you would use Dependency Injection (DI) in an ASP.NET MVC application, and discuss the benefits of using DI frameworks like Autofac or Ninject.
80. Describe your experience with implementing unit testing and integration testing in ASP.NET MVC projects, and discuss the tools and frameworks you have used for testing.
81. How would you implement localization and globalization in an ASP.NET MVC application to support multiple languages and cultures?
82. Explain your approach to managing and deploying ASP.NET MVC applications to different environments (e.g., development, staging, production) using tools like Azure DevOps or Jenkins.
83. Describe your experience with using asynchronous programming techniques (e.g., async/await) in ASP.NET MVC applications to improve performance and responsiveness.
84. How would you implement a custom HTML helper in ASP.NET MVC to simplify the creation of reusable UI components?
85. Explain how you would use SignalR in an ASP.NET MVC application to implement real-time communication features like chat or live updates.
86. Describe your experience with implementing Single Page Applications (SPAs) using frameworks like Angular, React, or Vue.js in conjunction with ASP.NET MVC Web API.
87. How would you handle cross-site scripting (XSS) and cross-site request forgery (CSRF) vulnerabilities in ASP.NET MVC applications?
88. Explain how you would implement caching strategies in ASP.NET MVC to improve the performance of your application?
89. Can you describe a complex problem you solved using ASP.NET MVC, and what was your approach?
90. How do you keep up-to-date with the latest trends and best practices in ASP.NET MVC development?
91. Explain your understanding of the OWASP top 10 vulnerabilities and how they relate to ASP.NET MVC applications.
92. Describe a scenario where you had to debug a performance issue in an ASP.NET MVC application. What tools and techniques did you use?