As an AI language model, I can provide you with a list of common interview questions for an 8-year experienced .NET developer. The questions cover various aspects of .NET development, including C#, ASP.NET, SQL, and general software development principles. Remember that interview questions can vary depending on the specific company and role, so it's essential to prepare for a wide range of topics. Here's a list of 100 interview questions:

1. Explain the difference between .NET Framework and .NET Core.

2. What are the new features introduced in C# 7.0, 8.0, and 9.0?

3. How does garbage collection work in .NET?

4. What are the different types of collections in C#?

5. What is the purpose of the 'using' statement in C#?

6. Describe the SOLID principles in object-oriented design.

7. How do you handle exceptions in C#?

8. Explain the differences between abstract classes and interfaces.

9. What is the difference between 'var' and 'dynamic' in C#?

10. Describe the asynchronous programming model in C# using 'async/await'.

11. How do you implement multithreading in .NET?

12. What is dependency injection, and how is it achieved in .NET?

13. Explain the role of delegates and events in C#.

14. Describe the concept of Boxing and Unboxing in C#.

15. How do you manage the state in ASP.NET applications?

16. What is ASP.NET Core, and what are its benefits over ASP.NET Framework?

17. What are the different types of authentication mechanisms in ASP.NET Core?

18. Explain the differences between Razor Pages and MVC in ASP.NET Core.

19. How do you enable Cross-Origin Resource Sharing (CORS) in an ASP.NET Web API?

20. What is Entity Framework, and how does it relate to databases in .NET applications?

21. How do you optimize database queries in Entity Framework?

22. Describe the advantages of using a stored procedure over inline SQL queries.

23. What are the different types of joins in SQL, and how are they used?

24. Explain the concepts of normalization and denormalization in databases.

25. How do you handle concurrency issues in a multi-user database environment?

26. Describe the role of Indexes in databases and their impact on query performance.

27. How do you improve the performance of an ASP.NET application?

28. What are NuGet packages, and how are they used in .NET projects?

29. Explain the purpose of the Global.asax file in ASP.NET.

30. How do you configure routing in ASP.NET MVC?

31. What is the use of TempData in ASP.NET MVC?

32. How do you secure sensitive data in a .NET application?

33. Explain the process of publishing an ASP.NET Core application to Azure.

34. What are Docker containers, and how can they be used in .NET development?

35. Describe the role of the 'appsettings.json' file in ASP.NET Core configuration.

36. How do you implement caching in .NET applications?

37. Explain the concept of inversion of control (IoC) and its benefits.

38. How do you monitor and log errors in a .NET application?

39. Describe the role of a web server and application server in ASP.NET hosting.

40. What is the purpose of the 'Web.config' file in ASP.NET?

41. How do you handle cross-site scripting (XSS) attacks in ASP.NET applications?

42. Explain the concept of Cross-Site Request Forgery (CSRF) and its prevention methods.

43. What is the role of Global Filters in ASP.NET MVC?

44. How do you implement user authentication and authorization in an ASP.NET Core application?

45. Describe the role of middleware in ASP.NET Core pipeline processing.

46. How do you enable session state in an ASP.NET Core application?

47. Explain the benefits of using Git for version control in .NET projects.

48. Describe the concept of Continuous Integration (CI) and Continuous Deployment (CD).

49. How do you implement a custom authentication provider in ASP.NET Core?

50. What are the different types of design patterns used in .NET development?

51. Explain the use of the 'yield' keyword in C# and its benefits.

52. How do you implement custom error handling in ASP.NET Core?

53. Describe the difference between 'readonly' and 'const' in C#.

54. What is the purpose of the 'ViewBag' in ASP.NET MVC?

55. Explain the role of the 'ModelState' in ASP.NET MVC validation.

56. How do you implement data validation in ASP.NET Core?

57. Describe the benefits of using Dependency Injection (DI) in .NET applications.

58. What is the purpose of the 'ActionResult' class in ASP.NET Core?

59. Explain the role of the 'IIS Express' web server in ASP.NET development.

60. How do you enable SSL/TLS in an ASP.NET application?

61. Describe the process of securing Web API endpoints using tokens.

62. How do you implement custom authentication schemes in ASP.NET Core?

63. Explain the differences between Inversion of Control (IoC) and Dependency Injection (DI).

64. What are the advantages of using an ORM like Entity Framework over traditional ADO.NET?

65. Describe the role of the 'web.config' file in ASP.NET configuration.

66. How do you implement file uploading in an ASP.NET Core application?

67. Explain the concept of Code-First and Database-First approaches in Entity Framework.

68. What is CORS, and how do you enable it in an ASP.NET Core Web API?

69. How do you implement data pagination in ASP.NET Core Web API?

70. Describe the benefits of using an ASP.NET Core middleware for logging.

71. How do you secure API endpoints in an ASP.NET Core application?

72. Explain the benefits of using a distributed caching system in ASP.NET Core.

73. How do you implement data seeding in Entity Framework Code-First?

74. Describe the role of the 'ViewModel' in ASP.NET MVC.

75. How do you implement localization in an ASP.NET Core application?

76. What is the purpose of the 'using' statement in C# when dealing with IDisposable objects?

77. Describe the process of optimizing database performance in SQL Server.

78. How do you implement unit testing in .NET projects using MSTest or NUnit?

79. Explain the difference between 'String' and 'StringBuilder' in C#.

80. Describe the benefits of using the 'async' and 'await' keywords in C#.

81. How do you manage application settings in an ASP.NET Core application?

82. Explain the differences between Razor Pages and Web Forms in ASP.NET Core.

83. What are the advantages of using AutoMapper in .NET projects?

84. Describe the role of CORS policy in ASP.NET Core Web API.

85. How do you implement caching in ASP.NET MVC?

86. Explain the process of deploying an ASP.NET Core application to Azure App Service.

87. What is the purpose of the 'Entity State' in Entity Framework?

88. Describe the concept of middleware in ASP.NET Core.

89. How do you handle concurrency issues in an Entity Framework context?

90. Explain the concept of Inversion of Control (IoC) containers in .NET.

91. What are the benefits of using SQL Server stored procedures?

92. How do you implement security best practices in ASP.NET applications?

93. Describe the role of the 'web.config' transformations in ASP.NET deployment.

94. How do you handle cross-site request forgery (CSRF) attacks in ASP.NET Core?

95. Explain the process of securing an ASP.NET Web API using JSON Web Tokens (JWT).

96. How do you optimize the performance of an ASP.NET Core application?

97. Describe the advantages of using asynchronous programming in .NET applications.

98. How do you manage session state in an ASP.NET application?

99. What are the benefits of using Blazor for building web applications?

100. Describe the process of debugging and troubleshooting in .NET projects.

Remember to study each topic thoroughly and practice coding examples to reinforce your knowledge before the interview. Good luck!