

65 Spring Framework interview questions to ask developers

Questions

1. Can you explain the core principles of Dependency Injection in Spring?
2. How does Spring Boot differ from the traditional Spring framework?
3. What is the Spring MVC framework and how does it work?
4. Explain the concept of AOP (Aspect-Oriented Programming) in Spring.
5. How does Spring handle database transactions?
6. What is Spring Security and how would you implement basic authentication in a Spring application?
7. How does Spring support testing, and what are some best practices for testing Spring applications?
8. Can you explain the concept of profiles in Spring and how they are used?
9. What is Inversion of Control in Spring and how does it work?
10. Explain the difference between @Component, @Service, and @Repository annotations.
11. How do you configure a data source in Spring?
12. What is the purpose of the @Autowired annotation?
13. Describe the lifecycle of a Spring Bean.
14. How do you handle exceptions in Spring MVC?
15. What is the role of DispatcherServlet in Spring MVC?
16. Explain the concept of Spring Bean Scopes.
17. How do you implement caching in a Spring application?
18. What is the difference between @RequestParam and @PathVariable annotations?
19. How do you schedule tasks in Spring?
20. Explain the purpose of the @Configuration annotation.
21. What is the role of ViewResolver in Spring MVC?
22. How do you implement validation in Spring?
23. What is the difference between constructor injection and setter injection?
24. How do you handle cross-origin requests in Spring?
25. Explain the concept of Spring Boot Actuator.
26. What is the purpose of the @Transactional annotation?
27. How do you implement logging in a Spring application?
28. Describe the steps to create a simple RESTful web service using Spring Boot.
29. How do you use the @Qualifier annotation in Spring?
30. What is the role of PropertyPlaceholderConfigurer in Spring?
31. How does Spring support internationalization (i18n)?
32. What is the difference between @Controller and @RestController in Spring MVC?
33. How do you manage application properties in Spring Boot?
34. Explain the concept of Spring Data JPA.
35. What are the benefits of using Spring's @Scheduled annotation?
36. How does Spring Boot simplify dependency management?
37. What is a Spring Boot Starter and why is it useful?
38. How do you handle application events in Spring?
39. Can you explain the concept of Inversion of Control (IoC) container in Spring?
40. What is the difference between BeanFactory and ApplicationContext?
41. How does Spring's IoC container manage bean lifecycles?
42. Explain the difference between singleton and prototype bean scopes.
43. What is the purpose of the @Autowired annotation and how does it work?
44. How does Spring handle circular dependencies between beans?
45. What is the role of BeanPostProcessor in Spring?
46. Explain the difference between constructor injection and setter injection in Spring.
47. What is the purpose of the @Value annotation in Spring?
48. How does Spring support internationalization?
49. What is the difference between @Component and @Bean annotations?
50. Explain the concept of method injection in Spring and when you might use it.
51. What is Dependency Injection and why is it important in Spring?
52. Can you explain the difference between field injection and constructor injection?
53. How do you define a dependency injection configuration in Spring?
54. What is the purpose of the @Autowired annotation in Spring?
55. What are some common issues you might encounter with dependency injection in Spring, and how do you resolve them?
56. How does Spring handle dependency injection for prototype-scoped beans?
57. What is a bean post-processor, and how does it relate to dependency injection?
58. Can you explain how to manage dependencies between beans in a Spring application?
59. How would you handle a situation where a Spring application starts to slow down due to an increase in traffic?
60. If you needed to integrate a third-party API into a Spring application, how would you approach this task?
61. How would you manage application properties for different environments (e.g., development, testing, production) in a Spring Boot application?
62. How would you approach debugging a Spring application that is throwing a runtime exception but the logs are not providing enough information?
63. Describe a situation where you had to implement a new feature in a legacy Spring application. How did you ensure compatibility and minimal disruption?
64. How would you handle a situation where a Spring application's database connection pool is exhausted, leading to failed requests?
65. How would you approach monitoring the health and performance of a Spring Boot application in production?