

Spring and Spring Boot – 30 Interview Questions and Answers

1. What is Spring Framework?

Spring is an open-source Java framework that provides infrastructure support for developing Java applications. It offers features like Dependency Injection, AOP, and transaction management.

2. What is Dependency Injection?

Dependency Injection (DI) is a design pattern where the dependencies of an object are injected by the framework rather than the object creating them.

3. What are the types of Dependency Injection in Spring?

- Constructor Injection
- Setter Injection
- Field Injection (not recommended for unit testing)

4. What is the Spring Bean lifecycle?

- Instantiate
- Populate properties
- Call `setBeanName()`
- Call `setBeanFactory()`
- Call `setApplicationContext()`
- Pre-initialization (BeanPostProcessor)
- Init method
- Post-initialization (BeanPostProcessor)
- Destruction

5. What is a Spring Bean?

A Spring Bean is an object that is instantiated, assembled, and managed by the Spring IoC container.

6. What are the scopes of Spring Bean?

- Singleton
- Prototype
- Request (Web)
- Session (Web)
- Application (Web)
- Websocket

7. Difference between ApplicationContext and BeanFactory?

- BeanFactory is the basic container providing DI.
- ApplicationContext is a more advanced container with features like AOP, internationalization, and event propagation.

8. What is @Component, @Controller, @Service, and @Repository?

They are specialized stereotypes for @Component to indicate different roles in an application.

9. What is the use of @Autowired?

Used to automatically inject dependencies.

10. What is Spring AOP?

Aspect-Oriented Programming in Spring allows separating cross-cutting concerns like logging and security.

11. What is Spring Boot?

Spring Boot simplifies Spring application development with features like auto-configuration, embedded servers, and starter dependencies.

12. What is a Spring Boot Starter?

A starter is a set of convenient dependency descriptors to include a group of libraries for a specific functionality.

13. How does Spring Boot auto-configuration work?

Spring Boot automatically configures your application based on the dependencies in the classpath using '@EnableAutoConfiguration'.

14. What is application.properties/application.yml?

Files used to configure application settings in Spring Boot.

15. How to create a Spring Boot project?

Using Spring Initializr, IDE (like IntelliJ/Eclipse), or manually by creating a Maven/Gradle project.

16. What is the default embedded server in Spring Boot?

Tomcat is the default. Others include Jetty and Undertow.

17. How do you change the server port in Spring Boot?

In 'application.properties':

```
``properties
server.port=8081
``
```

18. What is @SpringBootApplication?

It is a combination of `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan`.

19. How do you create a REST API using Spring Boot?

Use `@RestController` and `@RequestMapping` or `@GetMapping`, `@PostMapping`, etc.

20. What is the difference between @RestController and @Controller?

- `@Controller` returns views.
- `@RestController` returns data directly (e.g., JSON).

21. What is Spring Data JPA?

An abstraction over JPA to reduce boilerplate code and simplify database access.

22. What is the role of @Entity, @Id, and @GeneratedValue?

They map Java classes to database tables and define primary keys.

23. How do you handle exceptions in Spring Boot?

Use `@ControllerAdvice` and `@ExceptionHandler`.

24. What is Spring Security?

A framework for securing Java applications, handling authentication and authorization.

25. How to enable CORS in Spring Boot?

Use `@CrossOrigin` annotation or configure via `WebMvcConfigurer`.

26. What is Actuator in Spring Boot?

Spring Boot Actuator provides production-ready features like monitoring and metrics.

27. What are Profiles in Spring Boot?

Used to define different configurations for different environments like dev, test, and prod using `@Profile`.

28. How do you connect to a database in Spring Boot?

Add database dependency, configure properties in `application.properties`, and use Spring Data JPA or JdbcTemplate.

29. How to schedule tasks in Spring Boot?

Use `@Scheduled` annotation and enable scheduling via `@EnableScheduling`.

30. What is the use of CommandLineRunner and ApplicationRunner?

They run specific code after the Spring Boot application starts.