SpringBoot

1. **What is Spring Boot?**

   - **Answer:** Spring Boot is a framework designed to simplify the development of new Spring applications. It provides defaults for code and annotation configuration to start new Spring projects with minimal setup and effort.

2. **What are the advantages of using Spring Boot?**

   - **Answer:** Advantages include rapid application development, reduced boilerplate code, embedded servers, opinionated defaults, and easy integration with Spring ecosystem components.

3. **What are the main features of Spring Boot?**

   - **Answer:** Main features include autoconfiguration, standalone application, opinionated defaults, Spring Boot CLI, embedded server support, and production-ready features like metrics and health checks.

4. **How do you create a Spring Boot application?**

   - **Answer:** You can create a Spring Boot application using Spring Initializr (https://start.spring.io/), Spring Boot CLI, or by manually configuring a Maven/Gradle project.

5. **What is Spring Initializr?**

   - **Answer:** Spring Initializr is a web-based tool provided by Spring to generate Spring Boot project structure with the necessary dependencies and configuration.

6. **What is the Spring Boot Starter?**

   - **Answer:** Spring Boot Starters are a set of convenient dependency descriptors you can include in your application. They provide a single dependency to pull in a set of libraries typically used together.

7. **What is the difference between `@SpringBootApplication` and `@EnableAutoConfiguration`?**

   - **Answer:** `@SpringBootApplication` is a combination of `@Configuration`, `@EnableAutoConfiguration`, and `@ComponentScan`. It sets up default configurations, scans for components, and enables auto-configuration.

8. **What is the role of the `@SpringBootApplication` annotation?**

   - **Answer:** The `@SpringBootApplication` annotation marks the main class of a Spring Boot application and triggers auto-configuration, component scanning, and allows to define extra configuration on application context.

9. **How does Spring Boot autoconfiguration work?**

   - **Answer:** Spring Boot autoconfiguration automatically configures your Spring application based on the dependencies present in the classpath. It uses `@EnableAutoConfiguration` to automatically configure beans.

10. **What is the `application.properties` file in Spring Boot?**

    - **Answer:** The `application.properties` file is used to configure your Spring Boot application. It allows you to set application-specific properties such as server port, database configuration, and more.

11-20

11. **What is the `application.yml` file in Spring Boot?**

    - **Answer:** The `application.yml` file is an alternative to `application.properties` for configuring your Spring Boot application using YAML format, which supports hierarchical configuration.

12. **How do you configure the embedded Tomcat server in Spring Boot?**

    - **Answer:** You can configure the embedded Tomcat server using properties in `application.properties` or `application.yml` such as `server.port`, `server.servlet.context-path`, and `server.tomcat.max-threads`.

13. **How can you change the default port of the embedded server in Spring Boot?**

    - **Answer:** You can change the default port by setting the `server.port` property in `application.properties` or `application.yml`.

14. **What is Spring Boot DevTools?**

    - **Answer:** Spring Boot DevTools is a module that provides features to help with the development of Spring Boot applications, including automatic restart, live reload, and configurations for faster feedback during development.

15. **What is Spring Boot Actuator?**

- **Answer:** Spring Boot Actuator provides production-ready features to help monitor and manage your application, such as health checks, metrics, info, environment, and more through various endpoints.

16. **How do you enable Spring Boot Actuator?**

- **Answer:** You can enable Spring Boot Actuator by adding the `spring-boot-starter-actuator` dependency to your project and configuring endpoints in the `application.properties` or `application.yml`.

17. **What is a Spring Boot Starter Web?**

- **Answer:** `spring-boot-starter-web` is a starter dependency that simplifies setting up a Spring MVC application with embedded Tomcat and provides RESTful web services support.

18. **What is the use of `@RestController` annotation in Spring Boot?**

- **Answer:** The `@RestController` annotation is a specialized version of `@Controller` that combines `@Controller` and `@ResponseBody`. It is used to create RESTful web services and automatically serializes return objects to JSON/XML.

19. **How do you handle exceptions in Spring Boot?**

- **Answer:** You can handle exceptions using `@ControllerAdvice` and `@ExceptionHandler` annotations to define global exception handling logic and custom error responses.

20. **What is the purpose of `@RequestMapping` annotation?**

- **Answer:** The `@RequestMapping` annotation is used to map HTTP requests to handler methods in controller classes. It can map URLs, HTTP methods, request parameters, headers, and more.

21-30

21. **What is Spring Data JPA?**

- **Answer:** Spring Data JPA is part of the Spring Data family that makes it easy to implement JPA-based repositories. It provides a simplified data access layer and automates common persistence tasks.

22. **How do you enable Spring Data JPA repositories?**

- **Answer:** You enable Spring Data JPA repositories by using the `@EnableJpaRepositories` annotation in your configuration class and extending `JpaRepository` in your repository interfaces.

23. **What is `spring-boot-starter-data-jpa`?**

- **Answer:** `spring-boot-starter-data-jpa` is a starter dependency that includes Spring Data JPA, Hibernate, and other libraries required for building JPA-based data access layers in Spring Boot applications.

24. **What is `@Entity` annotation in JPA?**

- **Answer:** The `@Entity` annotation marks a class as a JPA entity and indicates that it should be mapped to a database table.

25. **What is the difference between `@Component`, `@Service`, `@Repository`, and `@Controller` annotations?**

- **Answer:**

  - **@Component:** A generic stereotype for any Spring-managed component.

  - **@Service:** Indicates a service layer component.

  - **@Repository:** Indicates a data access layer component and provides additional database exception translation.

  - **@Controller:** Indicates a web controller component.

26. **What is Spring Boot CLI?**

- **Answer:** Spring Boot CLI is a command-line tool that helps you quickly build Spring Boot applications using Groovy scripts. It provides commands for running, testing, and packaging applications.

27. **What is `@SpringBootTest` annotation?**

- **Answer:** The `@SpringBootTest` annotation is used for integration testing in Spring Boot applications. It loads the complete application context and allows you to write tests that require a fully initialized Spring environment.

28. **How do you profile-specific properties in Spring Boot?**

- **Answer:** Profile-specific properties can be defined using files named `application-{profile}.properties` or `application-{profile}.yml`. You can activate profiles using the `spring.profiles.active` property.

29. **What is the purpose of `@Configuration` annotation in Spring Boot?**

- **Answer:** The `@Configuration` annotation indicates that a class declares one or more `@Bean` methods and can be processed by the Spring container to generate bean definitions and service requests.

30. **What is the use of `@ConditionalOnProperty` annotation?**

- **Answer:** The `@ConditionalOnProperty` annotation is used to conditionally enable or disable bean definitions based on the presence or value of a property.

31-40

31. **What is Spring Boot Security?**

- **Answer:** Spring Boot Security simplifies the integration of Spring Security with Spring Boot applications. It provides security features such as authentication, authorization, and protection against common vulnerabilities.

32. **How do you secure a Spring Boot application?**

- **Answer:** You can secure a Spring Boot application by adding `spring-boot-starter-security` dependency and configuring security settings using `WebSecurityConfigurerAdapter`.

33. **What is OAuth2?**

- **Answer:** OAuth2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, or Google.

34. **How do you implement OAuth2 in a Spring Boot application?**

- **Answer:** You can implement OAuth2 in a Spring Boot application by using Spring Security OAuth2 and configuring authorization and resource servers using annotations and properties.

35. **What is Spring Boot Admin?**

- **Answer:** Spring Boot Admin is a community project used to manage and monitor Spring Boot applications. It provides a web UI for viewing application metrics, health checks, and other management information.

36. **What is the `@SpringBootApplication` equivalent in XML configuration?**

- **Answer:** The equivalent of `@SpringBootApplication` in XML configuration is combining `<context:component-scan>`, `<context:annotation-config>`, and `<mvc:annotation-driven>` tags.

37. **What is a Spring Boot actuator endpoint?**

- **Answer:** An actuator endpoint is a URL that provides operational information about your application, such as health, metrics, environment, and more. Actuator endpoints are enabled by including the Spring Boot Actuator dependency.

Of course, let's continue from where we left off and provide more questions and answers to reach the top 50 Spring Boot interview questions.

39-50

39. **What is `@Value` annotation in Spring Boot?**

- **Answer:** The `@Value` annotation is used to inject values into fields in a Spring-managed bean. These values can come from properties files, system properties, or other sources.

40. **What is Spring Boot DevTools and how does it help in development?**

- **Answer:** Spring Boot DevTools provides a set of tools that enhance the development experience. It includes features like automatic restart, live reload, and additional configurations for faster development and testing.

41. **How do you configure a datasource in Spring Boot?**

- **Answer:** You can configure a datasource in Spring Boot by setting properties in `application.properties` or `application.yml`, such as `spring.datasource.url`, `spring.datasource.username`, and `spring.datasource.password`.

42. **What is `SpringApplication.run()` method?**

- **Answer:** The `SpringApplication.run()` method is used to launch a Spring Boot application. It sets up the default configuration, starts the Spring application context, and performs a classpath scan.

43. **How do you create a custom starter in Spring Boot?**

- **Answer:** To create a custom starter, you need to define a new Maven or Gradle module, include the necessary dependencies, provide auto-configuration classes, and ensure they are registered in `META-INF/spring.factories`.

44. **What is the use of `@EnableAutoConfiguration` annotation?**

- **Answer:** The `@EnableAutoConfiguration` annotation enables Spring Boot’s auto-configuration mechanism, which attempts to automatically configure your Spring application based on the jar dependencies you have added.

45. **How does Spring Boot manage dependencies?**

- **Answer:** Spring Boot manages dependencies through a curated list of dependencies known as "starters." It provides a BOM (Bill Of Materials) to manage dependency versions, ensuring compatibility between dependencies.

46. **What is Spring Cloud?**

- **Answer:** Spring Cloud is a suite of tools designed to help developers build robust, resilient, and scalable distributed systems. It provides features such as service discovery, configuration management, and circuit breakers.

47. **What is Spring Boot's externalized configuration?**

- **Answer:** Spring Boot's externalized configuration allows you to define properties and configurations outside of your application code, using property files, YAML files, environment variables, and command-line arguments.

48. **What is `@ConfigurationProperties` in Spring Boot?**

- **Answer:** The `@ConfigurationProperties` annotation is used to bind external configuration properties to Java objects, making it easy to manage application configuration in a type-safe manner.

49. **What are embedded containers in Spring Boot?**

- **Answer:** Embedded containers are server instances that are packaged within the application. Spring Boot supports embedded servers like Tomcat, Jetty, and Undertow, allowing you to run web applications without external servers.

50. **How do you deploy a Spring Boot application?**

- **Answer:** You can deploy a Spring Boot application by packaging it as a JAR or WAR file and deploying it to a server, cloud platform, or container orchestration system like Kubernetes. For JAR files, you can run them directly using `java -jar`.