TCS T4 Spring Boot

Spring Boot Basics (10 Questions)

1. **What command is used to create a new Spring Boot project from the command line?**
   - A. spring init
   - B. mvn spring-boot:run
   - C. gradle spring-boot
   - D. spring create

Answer: A. spring init

2. **The default embedded server in Spring Boot applications is:**
   - A. Jetty
   - B. Undertow
   - C. WebLogic
   - D. Tomcat

Answer: D. Tomcat

3. **How can you disable the default web environment in a Spring Boot application?**
   - A. server.disable=true
   - B. spring.main.web-application-type=none
   - C. spring.server.disable=true
   - D. server.stop=true

Answer: B. spring.main.web-application-type=none

4. **Which starter would you use for Spring Boot and JPA integration?**
   - A. spring-boot-starter-data-jpa
   - B. spring-boot-starter-jpa
   - C. spring-boot-data-jpa
   - D. spring-boot-starter-sql

Answer: A. spring-boot-starter-data-jpa

5. **What annotation is used to run unit tests with the full Spring ApplicationContext?**
   - A. @RunWith(SpringJUnit4ClassRunner.class)
   - B. @Test
   - C. @SpringBootTest
   - D. @Autowired

Answer: C. @SpringBootTest

6. **Which file is used to customize application configuration in Spring Boot?**

   - A. application.properties or application.yml
   - B. config.properties
   - C. bootstrap.properties
   - D. settings.xml

Answer: A. application.properties or application.yml

7. **How do you specify a different application profile in Spring Boot?**

   - A. spring.config=dev
   - B. spring.profiles.active=dev
   - C. server.profile=dev
   - D. application.profile=dev

Answer: B. spring.profiles.active=dev

8. **To make a Spring Boot application exit with a specific code, which interface should be implemented?**

   - A. CommandLineRunner
   - B. ExitCodeGenerator
   - C. ExitCodeApplication
   - D. ExitApplication

Answer: B. ExitCodeGenerator

9. **What is the purpose of the @EnableAutoConfiguration annotation?**

   - A. To automatically configure dependencies and beans based on classpath
   - B. To scan for components in a specific package
   - C. To start the application server
   - D. To configure logging properties

Answer: A. To automatically configure dependencies and beans based on classpath

10. **In Spring Boot, which property is used to set the default server port to 8081?**

    - A. server.port=8081
    - B. port=8081
    - C. application.port=8081
    - D. spring.port=8081

Answer: A. server.port=8081

Spring Data JPA (10 Questions)

11. **Which annotation is used to define a primary key in a JPA entity?**

- A. @PrimaryKey
- B. @Id
- C. @Key
- D. @GeneratedValue

Answer: B. @Id

12. **What is the purpose of the @Query annotation in Spring Data JPA?**
    - A. To define custom SQL or JPQL queries
    - B. To establish foreign key relationships
    - C. To specify primary keys
    - D. To create views in the database

Answer: A. To define custom SQL or JPQL queries

11. **Which file is typically used to configure application-level properties in Spring Boot?**
    - A. application.yml
    - B. application.conf
    - C. config.xml
    - D. settings.json

Answer: A. application.yml

12. **What is the purpose of the @ComponentScan annotation in a Spring Boot application?**
    - A. To enable component scanning for Spring-managed beans
    - B. To exclude certain components from being loaded
    - C. To register components manually
    - D. To specify the main entry point of the application

Answer: A. To enable component scanning for Spring-managed beans

13. **How can a Spring Boot application be packaged as an executable JAR file?**
    - A. Use mvn compile
    - B. Configure spring-boot-maven-plugin and run mvn package
    - C. Use gradle bootJar
    - D. Both B and C

Answer: D. Both B and C

14. **Which of the following is NOT an advantage of using Spring Boot?**
    - A. Reduced boilerplate code
    - B. Simplified dependency management
    - C. Automatic JSON handling

- D. Embedded servers are not supported

Answer: D. Embedded servers are not supported

15. **In a Spring Boot application, where are custom configurations for development and production environments typically defined?**

    - A. application.properties
    - B. application-dev.properties and application-prod.properties
    - C. application-custom.xml
    - D. application-db.yml

Answer: B. application-dev.properties and application-prod.properties

16. **Which annotation in Spring Boot is used to mark a method as one that should execute once at application startup?**

    - A. @PostConstruct
    - B. @RunOnStartup
    - C. @ApplicationStartup
    - D. @EnableStartup

Answer: A. @PostConstruct

17. **Which of the following commands is used to run a Spring Boot application using Maven?**

    - A. mvn spring:start
    - B. mvn spring-boot:run
    - C. mvn run:app
    - D. mvn spring:bootRun

Answer: B. mvn spring-boot:run

18. **Which annotation is used to inject values from application.properties or application.yml files in Spring Boot?**

    - A. @ConfigValue
    - B. @Value
    - C. @Inject
    - D. @InjectValue

Answer: B. @Value

19. **How does Spring Boot support configuration properties inheritance in YAML files?**

    - A. Through the extends property
    - B. By using profiles and inheritance within profiles
    - C. By creating different yaml files for each class
    - D. By using the inherits keyword

Answer: B. By using profiles and inheritance within profiles

20. **Which annotation in Spring Boot applications denotes a command-line runner that executes upon application startup?**
    - A. @ApplicationRunner
    - B. @CommandLineRunner
    - C. @StartupRunner
    - D. @Runner

Answer: B. @CommandLineRunner

RESTful Services - 10 Questions

21. **Which annotation would you use to define a path variable in Spring Boot?**
    - A. @RequestParam
    - B. @RequestPath
    - C. @PathVariable
    - D. @Variable

Answer: C. @PathVariable

22. **How do you define a response status in a Spring Boot REST controller method?**
    - A. @Status
    - B. @ResponseCode
    - C. @HttpStatus
    - D. @ResponseStatus

Answer: D. @ResponseStatus

23. **Which HTTP status code is usually returned when no content is available to send for a request?**
    - A. 200 OK
    - B. 201 Created
    - C. 404 Not Found
    - D. 204 No Content

Answer: D. 204 No Content

24. **What is the purpose of the @RequestBody annotation in Spring Boot?**
    - A. To pass a request parameter
    - B. To map the request body to a Java object
    - C. To map the response to a JSON object
    - D. To specify HTTP headers

Answer: B. To map the request body to a Java object

25. **Which method is used to update a resource partially in a REST API?**
    - A. GET
    - B. POST
    - C. PUT
    - D. PATCH

Answer: D. PATCH

...

Microservices - 10 Questions

31. **What is the purpose of Spring Cloud Eureka in a microservices architecture?**
    - A. Service registration and discovery
    - B. Centralized configuration management
    - C. Rate limiting
    - D. Inter-service authentication

Answer: A. Service registration and discovery

32. **In Spring Boot microservices, what is the main role of an API Gateway?**
    - A. To serve static content
    - B. To route and load balance requests
    - C. To provide caching
    - D. To handle database connections

Answer: B. To route and load balance requests

33. **Which pattern in microservices helps prevent cascading failures by providing a fallback method?**
    - A. Retry
    - B. Circuit Breaker
    - C. API Gateway
    - D. Load Balancer

Answer: B. Circuit Breaker

...

Testing in Spring Boot - 10 Questions

41. **Which of the following annotations is used to initialize a Spring Boot test with the full context?**
    - A. @TestContext
    - B. @SpringBootTest
    - C. @ContextTest

- D. @ServiceTest

Answer: B. @SpringBootTest

42. **What is the purpose of @MockBean in Spring Boot testing?**
    - A. To mock database connections
    - B. To replace beans in the application context with mocks
    - C. To initialize real beans
    - D. To mock web requests

Answer: B. To replace beans in the application context with mocks

43. **In Spring Boot testing, which class is typically used to test controllers by simulating HTTP requests?**
    - A. TestHttpClient
    - B. MockMvc
    - C. HttpSimulator
    - D. HttpRequestTest

Answer: B. MockMvc

...

Database Interactions (Spring Data JPA) - 10 Questions

51. **What is the purpose of @Repository in Spring Data JPA?**
    - A. To define a REST endpoint
    - B. To provide a mechanism for CRUD operations on a data source
    - C. To store controllers
    - D. To define entity classes

Answer: B. To provide a mechanism for CRUD operations on a data source

52. **How does Spring Data JPA automatically implement common database operations like findAll and save?**
    - A. Through @Query annotations
    - B. By dynamically implementing repository interfaces
    - C. By using XML configuration
    - D. Through a configuration file

Answer: B. By dynamically implementing repository interfaces

Spring Boot Advanced Concepts (10 Questions)

1. **How can you expose actuator endpoints in a Spring Boot application?**
    - A. By setting management.endpoints.web.exposure.include in application.properties
    - B. By adding @ActuatorExpose

- C. By enabling expose: true in YAML
- D. Actuator endpoints are exposed by default

Answer: A. By setting management.endpoints.web.exposure.include in application.properties

2. **In Spring Boot, how do you customize the default error page?**
    - A. Implement ErrorPageConfigurer
    - B. Create a custom ErrorController class
    - C. Add a custom error page in application.yml
    - D. Set the error page path in application.properties

Answer: B. Create a custom ErrorController class

3. **What annotation is used in Spring Boot to enable asynchronous processing?**
    - A. @EnableAsync
    - B. @Async
    - C. @EnableAsynchronous
    - D. @AsyncProcessing

Answer: A. @EnableAsync

4. **What is Spring Boot DevTools primarily used for?**
    - A. Production performance monitoring
    - B. Remote debugging
    - C. Development-time features like automatic restarts and live reload
    - D. Testing data integrity

Answer: C. Development-time features like automatic restarts and live reload

5. **What is the default port for Spring Boot Actuator endpoints?**
    - A. 8081
    - B. 9090
    - C. 8000
    - D. 8080 (same as main server)

Answer: D. 8080 (same as main server)

6. **In Spring Boot, which starter dependency is added to include logging with SLF4J and Logback?**
    - A. spring-boot-starter-logging
    - B. spring-boot-starter-logback
    - C. spring-boot-starter-slf4j
    - D. spring-boot-logger

Answer: A. spring-boot-starter-logging

7. **To automatically restart a Spring Boot application on file changes, which module should you include?**
    - o A. spring-boot-starter-devtools
    - o B. spring-boot-starter-reload
    - o C. spring-boot-restart
    - o D. spring-boot-filewatcher

Answer: A. spring-boot-starter-devtools

8. **How can you enable SSL for a Spring Boot application?**
    - o A. ssl.enable=true in application.properties
    - o B. server.ssl.enabled=true and configuring keystore properties
    - o C. @EnableSSL on the main class
    - o D. By enabling https in Spring Boot

Answer: B. server.ssl.enabled=true and configuring keystore properties

9. **Which annotation is used to create a custom Spring Boot configuration class?**
    - o A. @Configurable
    - o B. @Configuration
    - o C. @Config
    - o D. @SpringBootConfig

Answer: B. @Configuration

10. **How does Spring Boot manage dependency injection by default?**
    - o A. XML configuration
    - o B. Automatic injection by scanning components with @Component
    - o C. JNDI lookup
    - o D. Directly referencing dependencies in application code

Answer: B. Automatic injection by scanning components with @Component

---

Microservices Advanced Concepts (10 Questions)

11. **What does Spring Cloud Zuul provide in a microservices architecture?**
    - o A. Distributed logging
    - o B. API Gateway
    - o C. Service registry
    - o D. Security features

Answer: B. API Gateway

12. **Which Spring Cloud component is used for centralized configuration management in microservices?**
    - A. Eureka
    - B. Ribbon
    - C. Config Server
    - D. Circuit Breaker

Answer: C. Config Server

13. **What pattern helps microservices tolerate network issues by retrying failed requests?**
    - A. Rate limiting
    - B. Circuit Breaker
    - C. Bulkhead
    - D. Retry pattern

Answer: D. Retry pattern

14. **What is the main purpose of Hystrix in Spring Cloud?**
    - A. Service registration and discovery
    - B. Load balancing
    - C. Circuit breaker and fault tolerance
    - D. Distributed transactions

Answer: C. Circuit breaker and fault tolerance

15. **Which property enables client-side load balancing with Spring Cloud Ribbon?**
    - A. ribbon.loadBalancer.enabled=true
    - B. ribbon.client.enabled=true
    - C. client.loadbalancer.active=true
    - D. load.balancer.enabled=true

Answer: A. ribbon.loadBalancer.enabled=true

...

Security (10 Questions)

21. **Which Spring Security annotation enables method-level security?**
    - A. @Secure
    - B. @EnableMethodSecurity
    - C. @EnableSecurity
    - D. @Secured

Answer: B. @EnableMethodSecurity

22. **What is the purpose of the @PreAuthorize annotation in Spring Security?**
    - A. To define method-level access control
    - B. To restrict access to HTTP endpoints
    - C. To configure user roles
    - D. To handle login and logout

Answer: A. To define method-level access control

23. **Which component in Spring Security is responsible for handling login attempts?**
    - A. UserDetailsService
    - B. AuthenticationProvider
    - C. LoginFilter
    - D. AuthenticationManager

Answer: D. AuthenticationManager

24. **In Spring Security, which class is used to customize user authentication?**
    - A. UserAuthService
    - B. CustomUserDetailsService
    - C. CustomAuthenticationManager
    - D. AuthenticationService

Answer: B. CustomUserDetailsService

25. **What is the purpose of the BCryptPasswordEncoder in Spring Security?**
    - A. To decode passwords
    - B. To encrypt passwords using the SHA algorithm
    - C. To hash passwords with bcrypt
    - D. To manage password storage

Answer: C. To hash passwords with bcrypt

...

Testing in Spring Boot (10 Questions)

31. **Which annotation is used in Spring Boot to mock beans within the application context?**
    - A. @Mock
    - B. @AutowiredMock
    - C. @MockBean
    - D. @MockedComponent

Answer: C. @MockBean

32. **Which Spring Boot test utility is specifically used to test HTTP endpoints?**

    - A. WebClientTest
    - B. MockHttpServletRequest
    - C. MockHttpTest
    - D. MockMvc

Answer: D. MockMvc

33. **What does the @DataJpaTest annotation configure for tests?**

    - A. Full application context with embedded server
    - B. In-memory database and JPA components for testing
    - C. Testing of JPA repositories without requiring a database connection
    - D. Testing of REST endpoints only

Answer: B. In-memory database and JPA components for testing

34. **Which method in MockMvc is used to simulate HTTP GET requests?**

    - A. getRequest()
    - B. performGet()
    - C. sendGet()
    - D. perform(MockMvcRequestBuilders.get())

Answer: D. perform(MockMvcRequestBuilders.get())

35. **What annotation enables the creation of a temporary database during tests?**

    - A. @TemporaryDatabase
    - B. @H2Database
    - C. @AutoConfigureTestDatabase
    - D. @AutoDatabase

Answer: C. @AutoConfigureTestDatabase

...

Spring Data JPA Advanced (10 Questions)

41. **What does the @Modifying annotation indicate in Spring Data JPA?**

    - A. A method updates or deletes records
    - B. A method reads only records
    - C. A method is paginated
    - D. A method is transactional

Answer: A. A method updates or deletes records

42. **How do you perform pagination in Spring Data JPA?**

- A. Use PageRequest in query methods
- B. Use Pagination interface
- C. Add LIMIT in SQL
- D. Set paging property in application.yml

Answer: A. Use PageRequest in query methods

43. **Which annotation specifies a native SQL query in Spring Data JPA?**
    - A. @SQL
    - B. @NativeQuery
    - C. @NativeQuerySQL
    - D. @Query

Answer: D. @Query

1. **Which annotation is used to specify the main configuration class in a Spring Boot application?**
    - A. @SpringBootApp
    - B. @SpringApplication
    - C. @SpringBootConfiguration
    - D. @SpringBootApplication

Answer: D. @SpringBootApplication

2. **What does the @EnableAutoConfiguration annotation do in a Spring Boot application?**
    - A. Disables auto-configuration
    - B. Automatically scans the classpath for components
    - C. Enables Spring Boot's auto-configuration based on the classpath
    - D. Only enables security configuration

Answer: C. Enables Spring Boot's auto-configuration based on the classpath

3. **What is the purpose of the @ConfigurationProperties annotation in Spring Boot?**
    - A. To bind external configuration properties to Java classes
    - B. To initialize beans in the application
    - C. To configure security settings
    - D. To start the application context

Answer: A. To bind external configuration properties to Java classes

4. **Which annotation enables component scanning in Spring Boot?**
    - A. @ComponentScan
    - B. @ScanComponents
    - C. @EnableComponents

- D. @Component

Answer: A. @ComponentScan

5. **In Spring Boot, which property is used to set the default server port in application.properties?**
    - A. spring.port
    - B. server.port
    - C. spring.boot.port
    - D. server.defaultPort

Answer: B. server.port

6. **Which annotation is used to indicate a repository in Spring Boot?**
    - A. @DAO
    - B. @Repository
    - C. @Service
    - D. @DataRepository

Answer: B. @Repository

7. **How can you run a Spring Boot application on a custom profile?**
    - A. spring-boot:run -Dspring.profiles.active=customProfile
    - B. java -jar app.jar --spring.profiles.active=customProfile
    - C. run spring -profile=customProfile
    - D. boot.run --profile=customProfile

Answer: B. java -jar app.jar --spring.profiles.active=customProfile

8. **Which annotation is used to handle exceptions at the controller level in Spring Boot?**
    - A. @HandleException
    - B. @ControllerAdvice
    - C. @ExceptionController
    - D. @ExceptionHandler

Answer: D. @ExceptionHandler

9. **How do you exclude a class from being considered in component scanning?**
    - A. @ExcludeComponent
    - B. @ComponentIgnore
    - C. @ComponentScan(excludeFilters)
    - D. @ExcludeFilter

Answer: C. @ComponentScan(excludeFilters)

10. **Which annotation enables scheduling tasks in Spring Boot?**

- A. @Schedule
- B. @EnableScheduling
- C. @Scheduled
- D. @TaskScheduler

Answer: B. @EnableScheduling

Microservices and Spring Cloud (10 Questions)

11. **What is the role of Eureka Server in a microservices architecture?**
    - A. Database management
    - B. Load balancing
    - C. Service registry
    - D. Security management

Answer: C. Service registry

12. **Which property is used to specify the Eureka server URL in the client configuration?**
    - A. eureka.server.url
    - B. eureka.instance.url
    - C. eureka.client.service-url.defaultZone
    - D. eureka.client.service.defaultZone

Answer: C. eureka.client.service-url.defaultZone

13. **Which Spring Cloud component provides client-side load balancing?**
    - A. Feign
    - B. Ribbon
    - C. Config Server
    - D. Zuul

Answer: B. Ribbon

14. **What is the main purpose of Spring Cloud Config Server?**
    - A. Service registration
    - B. Load balancing
    - C. Centralized external configuration management
    - D. API Gateway

Answer: C. Centralized external configuration management

15. **How can you enable the circuit breaker in Spring Cloud?**
    - A. @EnableCircuitBreaker

- B. @CircuitBreaker
- C. @EnableHystrix
- D. @EnableResilience

Answer: A. @EnableCircuitBreaker

16. **What is the purpose of Spring Cloud Feign?**
    - A. Load balancing
    - B. Service discovery
    - C. HTTP client for service-to-service calls
    - D. Distributed logging

Answer: C. HTTP client for service-to-service calls

17. **Which annotation is used to mark a Spring Cloud Feign client?**
    - A. @Feign
    - B. @FeignClient
    - C. @FeignService
    - D. @FeignController

Answer: B. @FeignClient

18. **In a Spring Cloud Config Server, which configuration repository can you use?**
    - A. Git repository
    - B. Local file system
    - C. SVN repository
    - D. All of the above

Answer: D. All of the above

19. **Which component in Spring Cloud provides distributed tracing?**
    - A. Zipkin
    - B. Eureka
    - C. Hystrix
    - D. Config Server

Answer: A. Zipkin

20. **Which library provides service discovery for Spring Cloud applications?**
    - A. Hystrix
    - B. Ribbon
    - C. Eureka
    - D. Zuul

Answer: C. Eureka

Spring Data JPA (10 Questions)

21. **What is the purpose of the @Id annotation in a Spring Data JPA entity?**
    - A. To mark a field as ignored
    - B. To mark a field as a primary key
    - C. To mark a field for validation
    - D. To mark a field as a foreign key

Answer: B. To mark a field as a primary key

22. **How can you enable caching in Spring Data JPA?**
    - A. @EnableCache
    - B. @EnableCaching
    - C. @Cache
    - D. @Cacheable

Answer: B. @EnableCaching

23. **Which annotation is used to specify a custom query in Spring Data JPA?**
    - A. @SQL
    - B. @JpaQuery
    - C. @CustomQuery
    - D. @Query

Answer: D. @Query

24. **What does the @OneToMany annotation specify in Spring Data JPA?**
    - A. One-to-one relationship
    - B. One-to-many relationship
    - C. Many-to-many relationship
    - D. Many-to-one relationship

Answer: B. One-to-many relationship

25. **How can you define a primary key strategy in Spring Data JPA?**
    - A. @PrimaryKey
    - B. @GeneratedValue(strategy=…)
    - C. @PrimaryKeyGenerated
    - D. @KeyStrategy

Answer: B. @GeneratedValue(strategy=…)

26. **Which of the following is NOT a valid cascade type in JPA?**
    - A. MERGE
    - B. UPDATE
    - C. PERSIST
    - D. DETACH

Answer: B. UPDATE

27. **In JPA, what is the purpose of the @Embeddable annotation?**
    - A. To mark a class as having a composite primary key
    - B. To allow a class to be embedded within another entity
    - C. To mark a class as an embedded database
    - D. To enable caching of entities

Answer: B. To allow a class to be embedded within another entity

28. **What type of relationship is represented by @ManyToOne?**
    - A. Many-to-many
    - B. Many-to-one
    - C. One-to-one
    - D. One-to-many

Answer: B. Many-to-one

29. **Which annotation is used to specify the name of a database table in an entity class?**
    - A. @TableName
    - B. @EntityTable
    - C. @Table
    - D. @DatabaseTable

Answer: C. @Table

30. **In JPA, which annotation is used to define a unique constraint?**
    - A. @Unique
    - B. @UniqueConstraint
    - C. @Column(unique=true)
    - D. @Table(uniqueConstraints=…)

Answer: D. @Table(uniqueConstraints=…)

1. **In Spring Boot, which of these configurations is typically used for unit tests?**
    - A. @RunWith
    - B. @TestConfiguration

- C. @MockBean
- D. @TestProperty

Answer: B. @TestConfiguration

2. **What is the primary purpose of @SpringBootTest?**
   - A. To run integration tests with a real server
   - B. To enable in-memory databases
   - C. To configure data source beans only
   - D. To mock service beans

Answer: A. To run integration tests with a real server

3. **How does Spring Boot manage logging by default?**
   - A. Logback
   - B. Log4j
   - C. JUL
   - D. SLF4J

Answer: A. Logback

4. **What is the function of the @Retryable annotation in Spring Cloud?**
   - A. Retries failed requests automatically
   - B. Caches data for repeated requests
   - C. Enables load balancing
   - D. Enables monitoring

Answer: A. Retries failed requests automatically

5. **Which HTTP status code generally represents a "Conflict" error?**
   - A. 400
   - B. 401
   - C. 409
   - D. 500

Answer: C. 409

6. **Which annotation is used to specify a controller advice for handling exceptions globally?**
   - A. @GlobalExceptionHandler
   - B. @ControllerAdvice
   - C. @RestControllerAdvice
   - D. @ExceptionHandler

Answer: B. @ControllerAdvice

7. **What is the default thread pool size for asynchronous tasks in Spring Boot?**
   - A. 4
   - B. 5
   - C. 8
   - D. 10

Answer: D. 10

8. **Which property is used to enable async support in Spring Boot?**
   - A. spring.async.enabled
   - B. spring.task.execution.pool.enabled
   - C. task.async.enable
   - D. @EnableAsync

Answer: D. @EnableAsync

9. **In Spring Boot, which annotation do you use for scheduling a method to run periodically?**
   - A. @Scheduled
   - B. @Cron
   - C. @EnableScheduling
   - D. @EnableTaskScheduler

Answer: A. @Scheduled

10. **Which property controls the logging level in Spring Boot?**
    - A. logging.level
    - B. log.level
    - C. logger.level
    - D. logging.debug.level

Answer: A. logging.level

Spring Data JPA and Hibernate (10 Questions)

11. **What is the default fetch type for @OneToMany relationships in JPA?**
    - A. EAGER
    - B. LAZY
    - C. IMMEDIATE
    - D. DYNAMIC

Answer: B. LAZY

12. **What is the function of @GeneratedValue with strategy = GenerationType.IDENTITY?**

- A. Uses database table for ID generation
- B. Relies on database identity column for primary key
- C. Uses UUID for ID generation
- D. Generates custom ID

Answer: B. Relies on database identity column for primary key

13. **Which annotation allows combining multiple filter conditions in JPA?**
    - A. @Filter
    - B. @FilterChain
    - C. @FilterDef
    - D. @Filters

Answer: D. @Filters

14. **In Spring Data JPA, what method can you use to save a list of entities?**
    - A. saveBatch
    - B. saveAll
    - C. insertList
    - D. persistAll

Answer: B. saveAll

15. **What is the function of @Modifying in JPA?**
    - A. Marks a query as modifiable
    - B. Executes a write operation
    - C. Allows batch updates
    - D. Allows for nested transactions

Answer: B. Executes a write operation

16. **What does @Inheritance(strategy = InheritanceType.JOINED) specify in JPA?**
    - A. Single table inheritance
    - B. Joined table inheritance
    - C. Table per class inheritance
    - D. Table union inheritance

Answer: B. Joined table inheritance

17. **In a JPA repository, which method retrieves the first entity matching a condition?**
    - A. findOne
    - B. findTopBy
    - C. findFirstBy

- D. findUniqueBy

Answer: C. findFirstBy

18. **Which annotation in JPA represents a composite key class?**

    - A. @CompositeKey
    - B. @EmbeddedId
    - C. @PrimaryKeyClass
    - D. @MultiKey

Answer: B. @EmbeddedId

19. **Which property enables show SQL in application.properties?**

    - A. spring.sql.show
    - B. spring.jpa.show-sql
    - C. spring.jpa.sql-enabled
    - D. sql.display

Answer: B. spring.jpa.show-sql

20. **In Spring Data JPA, what method deletes all records of a particular entity?**

    - A. deleteAll
    - B. removeAll
    - C. deleteRecords
    - D. removeEntities

Answer: A. deleteAll

Spring Security (10 Questions)

21. **Which filter is applied first in the Spring Security filter chain by default?**

    - A. AuthenticationFilter
    - B. UsernamePasswordAuthenticationFilter
    - C. SecurityContextPersistenceFilter
    - D. LogoutFilter

Answer: C. SecurityContextPersistenceFilter

22. **How can you specify which endpoints to secure in Spring Security?**

    - A. @ControllerSecurity
    - B. @SecurityEnabled
    - C. HttpSecurity
    - D. EndpointSecurity

Answer: C. HttpSecurity

23. **Which annotation is used to disable security for specific methods in Spring?**
    - A. @AllowAll
    - B. @Unsecured
    - C. @PermitAll
    - D. @AllowUnsecured

Answer: C. @PermitAll

24. **Which annotation enables method-level security?**
    - A. @EnableWebSecurity
    - B. @EnableGlobalMethodSecurity
    - C. @EnableSecurity
    - D. @SecurityConfig

Answer: B. @EnableGlobalMethodSecurity

25. **Which annotation is used to secure a method with roles?**
    - A. @AllowRoles
    - B. @Secured
    - C. @RoleAllowed
    - D. @PermitRoles

Answer: B. @Secured

26. **In Spring Security, which method sets user roles?**
    - A. setAuthorities
    - B. grantRoles
    - C. setRoles
    - D. authorizeRoles

Answer: A. setAuthorities

27. **Which annotation is used to secure REST APIs using Spring Security?**
    - A. @SecuredREST
    - B. @RestSecure
    - C. @RestController
    - D. @PreAuthorize

Answer: D. @PreAuthorize

28. **How do you set a custom login page in Spring Security?**
    - A. http.setCustomLogin

- B. http.formLogin().loginPage()
- C. http.loginCustom()
- D. http.customLoginPage()

Answer: B. http.formLogin().loginPage()

29. **Which of the following is not a valid HttpMethod in Spring Security?**
    - A. GET
    - B. PATCH
    - C. TRACE
    - D. CONNECT

Answer: D. CONNECT

30. **What is the purpose of @WithMockUser in testing?**
    - A. To mock a user session
    - B. To set default user credentials for tests
    - C. To simulate multiple user sessions
    - D. To add roles to a test user

Answer: B. To set default user credentials for tests

Spring Boot Testing (10 Questions)

31. **What does @WebMvcTest test in Spring Boot?**
    - A. Full application context
    - B. Only MVC components
    - C. Database layer
    - D. Security configurations

Answer: B. Only MVC components

32. **Which annotation is used for parameterized tests in Spring Boot?**
    - A. @TestParams
    - B. @ParameterizedTest
    - C. @TestParameterized
    - D. @ParamsTest

Answer: B. @ParameterizedTest

33. **Which tool is commonly used for Spring Boot integration testing?**
    - A. JUnit
    - B. Mockito

- C. RestAssured
- D. DBUnit

Answer: C. RestAssured

Dependency Injection (50 Questions)

1. **What is Dependency Injection in Spring?**
   - A. Process of creating beans
   - B. Process of providing the dependent objects of a class
   - C. Process of calling methods automatically
   - D. Process of managing memory

Answer: B. Process of providing the dependent objects of a class

2. **Which of the following is a type of Dependency Injection in Spring?**
   - A. Constructor Injection
   - B. Method Injection
   - C. Field Injection
   - D. All of the above

Answer: D. All of the above

3. **Which annotation is used to inject a bean in a Spring class using constructor-based dependency injection?**
   - A. @Inject
   - B. @Autowired
   - C. @ConstructorInject
   - D. @Autowired(required=false)

Answer: B. @Autowired

4. **Which of the following is used for defining a bean in Spring?**
   - A. @Bean
   - B. @Component
   - C. @Service
   - D. All of the above

Answer: D. All of the above

5. **In Spring, what happens if there is more than one bean of the same type and you use @Autowired?**
   - A. It will throw an exception
   - B. It will inject the first bean found
   - C. It will prompt the user to select a bean

- D. It will inject the bean with the highest priority

Answer: A. It will throw an exception

6. **Which annotation is used for marking a class as a Spring bean?**
   - A. @Entity
   - B. @Controller
   - C. @Component
   - D. @Service

Answer: C. @Component

7. **What is the scope of a Spring bean by default?**
   - A. Singleton
   - B. Prototype
   - C. Request
   - D. Session

Answer: A. Singleton

8. **Which of the following is used to inject a bean into a field?**
   - A. @Autowired
   - B. @Bean
   - C. @Component
   - D. @Inject

Answer: A. @Autowired

9. **What is the advantage of Dependency Injection?**
   - A. It reduces the coupling between objects
   - B. It allows multiple objects to be instantiated
   - C. It eliminates the need for object creation
   - D. It reduces memory consumption

Answer: A. It reduces the coupling between objects

10. **Which annotation is used to specify a bean with multiple constructors?**
    - A. @Autowired
    - B. @Primary
    - C. @Qualifier
    - D. @Bean

Answer: C. @Qualifier

Spring Task Scheduler and Cron Jobs (50 Questions)

11. **Which annotation is used to schedule a method in Spring?**
    - A. @TaskScheduler
    - B. @Scheduled
    - C. @EnableScheduling
    - D. @Cron

Answer: B. @Scheduled

12. **How can you specify the cron expression for scheduling tasks in Spring?**
    - A. @Scheduled(cron = "0 0 12 * * ?")
    - B. @Scheduled(cronExpression = "0 0 12 * * ?")
    - C. @Scheduled(cronTime = "0 0 12 * * ?")
    - D. @Scheduled(cron="*/10 * * * * ?")

Answer: A. @Scheduled(cron = "0 0 12 * * ?")

13. **Which of the following annotations enables task scheduling in Spring Boot?**
    - A. @Scheduled
    - B. @EnableScheduling
    - C. @EnableTaskScheduling
    - D. @EnableCron

Answer: B. @EnableScheduling

14. **What is the default fixed rate for scheduling tasks in Spring Boot?**
    - A. 1 second
    - B. 5 seconds
    - C. 10 seconds
    - D. 60 seconds

Answer: A. 1 second

15. **What is the purpose of @EnableScheduling annotation in Spring Boot?**
    - A. To schedule tasks at fixed intervals
    - B. To enable scheduling in the Spring application
    - C. To enable task execution on specified intervals
    - D. To manage thread pools for scheduled tasks

Answer: B. To enable scheduling in the Spring application

16. **What is the cron expression used to run a task every day at 2 AM?**
    - A. 0 2 * * *

- B. 0 2 1 * * ?
- C. 0 0 2 * * ?
- D. 0 0 1 2 * ?

Answer: C. 0 0 2 * * ?

17. **Which of the following is a valid cron expression?**
    - A. */5 * * * *
    - B. 5 5 5 5 5
    - C. */2 * * *
    - D. * * * * * *

Answer: A. */5 * * * *

18. **Which of the following is true about the @Scheduled(fixedDelay = 1000) annotation?**
    - A. The task runs every 1 second regardless of the completion time of the previous task
    - B. The task runs 1 second after the completion of the previous task
    - C. The task runs every 1 second but does not wait for the previous task to finish
    - D. The task runs every 1 second and blocks the next execution

Answer: B. The task runs 1 second after the completion of the previous task

19. **What is the role of @TaskExecutor in Spring?**
    - A. It defines the task scheduling configurations
    - B. It executes tasks concurrently
    - C. It defines cron expression
    - D. It specifies when to execute scheduled tasks

Answer: B. It executes tasks concurrently

20. **Which of the following is a valid cron expression format in Spring?**
    - A. seconds minutes hours day-of-month month day-of-week
    - B. minutes seconds hours day-of-week month day-of-month
    - C. hour minute second month day-of-week
    - D. minute second hour day-of-week month

Answer: A. seconds minutes hours day-of-month month day-of-week

---

Authorization and Authentication (50 Questions)

21. **Which of the following is used to configure Spring Security for form-based login?**
    - A. http.formLogin().loginPage("/login")
    - B. http.authenticationForm().loginPage("/login")

- C. http.enableLogin().loginPage("/login")
- D. http.formLogin().loginUrl("/login")

Answer: A. http.formLogin().loginPage("/login")

22. **Which of the following is the correct order for Spring Security filters?**
    - A. SecurityContextPersistenceFilter -> UsernamePasswordAuthenticationFilter -> ExceptionTranslationFilter
    - B. SecurityContextPersistenceFilter -> ExceptionTranslationFilter -> UsernamePasswordAuthenticationFilter
    - C. ExceptionTranslationFilter -> SecurityContextPersistenceFilter -> UsernamePasswordAuthenticationFilter
    - D. UsernamePasswordAuthenticationFilter -> SecurityContextPersistenceFilter -> ExceptionTranslationFilter

Answer: A. SecurityContextPersistenceFilter -> UsernamePasswordAuthenticationFilter -> ExceptionTranslationFilter

23. **What does the @PreAuthorize annotation in Spring Security do?**
    - A. Grants access based on method return value
    - B. Defines roles required to access a method
    - C. Enables logging for all security events
    - D. Secures HTTP requests at runtime

Answer: B. Defines roles required to access a method

24. **Which annotation is used for role-based method security in Spring?**
    - A. @RolesAllowed
    - B. @Secured
    - C. @PreAuthorize
    - D. @AllowRoles

Answer: B. @Secured

25. **In Spring Security, what is the default role prefix?**
    - A. ROLE_
    - B. USER_
    - C. ADMIN_
    - D. SECURITY_

Answer: A. ROLE_

26. **What is the primary purpose of SecurityContextHolder in Spring Security?**
    - A. To hold session data
    - B. To store authentication data

- C. To store user roles
- D. To store encryption keys

Answer: B. To store authentication data

27. **Which of the following is true about OAuth2 authentication?**
    - A. OAuth2 only supports username-password authentication
    - B. OAuth2 is used for authorizing third-party applications
    - C. OAuth2 does not require any client credentials
    - D. OAuth2 is used for session management in Spring Security

Answer: B. OAuth2 is used for authorizing third-party applications

28. **Which of the following is used for JWT (JSON Web Token) authentication in Spring Security?**
    - A. JwtAuthenticationFilter
    - B. JWTAuthorizationFilter
    - C. OAuth2AuthenticationFilter
    - D. JwtTokenProvider

Answer: A. JwtAuthenticationFilter

29. **Which annotation is used for securing HTTP requests in Spring Security?**
    - A. @Secured
    - B. @PreAuthorize
    - C. @EnableWebSecurity
    - D. @SecurityConfig

Answer: C. @EnableWebSecurity

30. **Which of the following is true about @Secured annotation in Spring Security?**
    - A. It specifies the user roles required to access a method
    - B. It authenticates the user before method execution
    - C. It is used to configure the login form
    - D. It is used to configure basic authentication

Answer: A. It specifies the user roles required to access a method

---

Spring Events and Event Listeners (50 Questions)

31. **Which annotation is used to mark a method as an event listener in Spring?**
    - A. @EventHandler
    - B. @Subscribe
    - C. @EventListener

- D. @Listener

Answer: C. @EventListener

32. **What is the purpose of ApplicationEventPublisher in Spring?**
    - A. To listen to events
    - B. To publish events to listeners
    - C. To manage events within the application context
    - D. To configure event-driven architecture

Answer: B. To publish events to listeners

33. **Which of the following is used to publish an event in Spring?**
    - A. event.publish()
    - B. applicationContext.publishEvent()
    - C. applicationContext.getEventPublisher()
    - D. applicationEventListener.publish()

Answer: B. applicationContext.publishEvent()

34. **What is the default event type for Spring's application events?**
    - A. ApplicationStartedEvent
    - B. ApplicationContextEvent
    - C. ContextRefreshedEvent
    - D. ApplicationEvent

Answer: D. ApplicationEvent

35. **How do you specify the type of event to listen for in a Spring event listener?**
    - A. By providing the event class as a parameter to the @EventListener method
    - B. By annotating the listener class with the event type
    - C. By configuring the event type in application.properties
    - D. By extending ApplicationListener

Answer: A. By providing the event class as a parameter to the @EventListener method

Dependency Injection

1. What is the core principle of Dependency Injection?
    - A. Tight coupling
    - B. Loose coupling
    - C. Inversion of Control
    - D. Dependency inversion
    - **Answer: C**

2. Which of the following is NOT a type of dependency injection?
   - A. Constructor injection
   - B. Setter injection
   - C. Field injection
   - D. Method injection
   - **Answer: D**
3. What is the role of the @Autowired annotation in Spring?
   - A. Marks a method as a bean factory method
   - B. Marks a field or method parameter for automatic dependency injection
   - C. Marks a class as a configuration class
   - D. Marks a method as an interceptor
   - **Answer: B**

Spring Task Scheduler and Cron Jobs

4. What is the primary purpose of the @Scheduled annotation?
   - A. To mark a method as a transaction
   - B. To mark a method as an aspect
   - C. To mark a method to be executed periodically
   - D. To mark a method as a controller
   - **Answer: C**
5. What is the correct cron expression to run a task every Monday at 9 AM?
   - A. 0 0 9 * * MON
   - B. 0 9 * * MON
   - C. 9 0 * * MON
   - D. MON 9 0 * *
   - **Answer: A**

Authorization and Authentication

6. What is the primary difference between authentication and authorization?
   - A. Authentication verifies user identity, authorization grants permissions.
   - B. Authentication grants permissions, authorization verifies user identity.
   - C. Both are the same.
   - D. Neither is related to security.
   - **Answer: A**
7. Which Spring Security annotation is used to protect a controller method?

- A. @Secured
- B. @PreAuthorize
- C. @PostAuthorize
- D. All of the above
- **Answer: D**

Spring Events and Event Listeners

8. What is the purpose of the @EventListener annotation?
   - A. To mark a method as a controller
   - B. To mark a method as a transaction
   - C. To mark a method as an event listener
   - D. To mark a method as a scheduled task
   - **Answer: C**
9. How are Spring events typically propagated?
   - A. Using a message queue
   - B. Using a publish-subscribe mechanism
   - C. Using a direct method invocation
   - D. Using a remote procedure call
   - **Answer: B**

Spring Messaging

10. What is the primary purpose of Spring Integration?
    - A. To provide a framework for building enterprise integration solutions
    - B. To provide a framework for building RESTful web services
    - C. To provide a framework for building reactive applications
    - D. To provide a framework for building microservices
    - **Answer: A**
11. Which of the following is NOT a core concept in Spring Integration?
    - A. Message Channel
    - B. Message Transformer
    - C. Message Filter
    - D. Message Queue
    - **Answer: D**

Spring Boot Starters

12. What is the primary purpose of Spring Boot starters?

- A. To simplify the configuration of Spring applications
  - B. To provide a framework for building microservices
  - C. To provide a framework for building reactive applications
  - D. To provide a framework for building enterprise integration solutions
  - **Answer: A**

Spring MVC

13. What is the role of the @RequestMapping annotation in Spring MVC?
    - A. To map a method to a specific HTTP request
    - B. To mark a class as a controller
    - C. To mark a method as a transaction
    - D. To mark a method as an event listener
    - **Answer: A**
14. Which of the following is NOT a common HTTP method supported by Spring MVC?
    - A. GET
    - B. POST
    - C. PUT
    - D. DELETE
    - **Answer: D**

Spring Data

15. What is the primary purpose of Spring Data JPA?

- A. To simplify database interactions
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building microservices
- **Answer: A**

16. Which annotation is used to mark a class as a JPA repository?

- A. @Repository
- B. @JpaRepository
- C. @JpaEntity
- D. @Transactional
- **Answer: B**

Spring Security

17. What is the primary purpose of Spring Security?

- A. To provide security features like authentication and authorization
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building microservices
- **Answer: A**

18. Which authentication mechanism is commonly used with Spring Security?

- A. Basic Authentication
- B. Form-Based Authentication
- C. OAuth2
- D. All of the above
- **Answer: D**

19. What is the role of the @PreAuthorize annotation in Spring Security?

- A. To authorize access to a method before execution
- B. To authorize access to a method after execution
- C. To log all method invocations
- D. To cache method invocations
- **Answer: A**

Spring Boot

20. What is the primary purpose of Spring Boot?

- A. To simplify the development of Spring applications
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building microservices
- **Answer: A**

21. Which Spring Boot starter POM is used for web applications?

- A. spring-boot-starter-web
- B. spring-boot-starter-data-jpa
- C. spring-boot-starter-security
- D. spring-boot-starter-test
- **Answer: A**

Spring AOP

22. What is the core concept of Aspect-Oriented Programming (AOP)?

- A. Separating concerns across multiple modules
- B. Modularizing cross-cutting concerns
- C. Encapsulating business logic
- D. All of the above
- **Answer: B**

23. Which annotation is used to define an aspect in Spring AOP?

- A. @Aspect
- B. @Component
- C. @Service
- D. @Controller
- **Answer: A**

Spring Cloud

24. What is the primary purpose of Spring Cloud?

- A. To build distributed systems and microservices
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building enterprise integration solutions
- **Answer: A**

25. Which Spring Cloud component is used for service discovery?

- A. Spring Cloud Config
- B. Spring Cloud Netflix Eureka
- C. Spring Cloud Gateway
- D. Spring Cloud Bus
- **Answer: B**

Spring Batch

26. What is the primary purpose of Spring Batch?

- A. To build batch processing applications
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building microservices
- **Answer: A**

27. Which of the following is a core component of a Spring Batch job?

- A. Job Repository
- B. Job Launcher
- C. Step
- D. All of the above
- **Answer: D**

Spring JMS

28. What is the primary purpose of Spring JMS?

- A. To provide a framework for building messaging applications
- B. To provide a framework for building RESTful web services
- C. To provide a framework for building reactive applications
- D. To provide a framework for building microservices
- **Answer: A**

29. Which of the following is a common JMS provider?

- A

generate more

Show drafts

Here are 25 more MCQs with answers on Spring Framework:

Spring Data

30. What is the purpose of the @Query annotation in Spring Data JPA?

- A. To define custom JPQL or SQL queries
- B. To specify the query method name
- C. To enable pagination
- D. To enable sorting
- **Answer: A**

31. What is the purpose of the @Transactional annotation in Spring Data JPA?

- A. To mark a method as a transaction boundary
- B. To specify the query method name
- C. To enable pagination

- D. To enable sorting
- **Answer: A**

Spring Security

32. What is the role of the UserDetailsService interface in Spring Security?

- A. To load user information from a data source
- B. To authenticate users
- C. To authorize user access
- D. To encrypt passwords
- **Answer: A**

33. What is the purpose of the @EnableWebSecurity annotation in Spring Security?

- A. To enable Spring Security in a web application
- B. To define custom security filters
- C. To configure authentication and authorization
- D. To encrypt passwords
- **Answer: A**

Spring Boot

34. What is the purpose of the @SpringBootApplication annotation?

- A. To mark a class as a Spring Boot application
- B. To enable Spring Boot auto-configuration
- C. To define the main class of a Spring Boot application
- D. All of the above
- **Answer: D**

35. What is the purpose of the application.properties file in Spring Boot?

- A. To configure application properties
- B. To define beans
- C. To define controllers
- D. To define services
- **Answer: A**

Spring AOP

36. What is the role of the Pointcut in Spring AOP?

- A. To define the join points where advice should be applied
- B. To provide the implementation of the advice

- C. To weave the advice into the target object
- D. To define the order of execution of advice
- **Answer: A**

37. What is the role of the Advice in Spring AOP?

- A. To define the join points where advice should be applied
- B. To provide the implementation of the advice
- C. To weave the advice into the target object
- D. To define the order of execution of advice
- **Answer: B**

Spring Cloud

38. What is the purpose of Spring Cloud Config?

- A. To centralize configuration management
- B. To provide service discovery
- C. To implement circuit breakers
- D. To provide API gateway functionality
- **Answer: A**

39. What is the purpose of Spring Cloud Gateway?

- A. To centralize configuration management
- B. To provide service discovery
- C. To implement circuit breakers
- D. To provide API gateway functionality
- **Answer: D**

Spring Batch

40. What is the purpose of the Job interface in Spring Batch?

- A. To define a unit of work in a batch job
- B. To launch a batch job
- C. To monitor the execution of a batch job
- D. To restart a failed batch job
- **Answer: A**

41. What is the purpose of the Step interface in Spring Batch?

- A. To define a unit of work in a batch job
- B. To launch a batch job

- C. To monitor the execution of a batch job
- D. To restart a failed batch job
- **Answer: A**

Spring JMS

42. What is the purpose of the Message interface in Spring JMS?

- A. To represent a message in a JMS message system
- B. To send a message to a JMS destination
- C. To receive a message from a JMS destination
- D. To create a JMS connection
- **Answer: A**

43. What is the purpose of a Message-Driven POJO (MDP) in Spring JMS?

- A. To receive messages asynchronously
- B. To send messages synchronously
- C. To create JMS connections
- D. To configure JMS destinations

44. What is the purpose of the @EnableJpaRepositories annotation in Spring Data JPA?

- A. To enable JPA repositories in a Spring Boot application
- B. To specify the base package for JPA repositories
- C. To configure the JPA entity manager
- D. All of the above
- **Answer: D**

45. What is the purpose of the @Transactional annotation in Spring Data JPA repositories?

- A. To mark a method as a transaction boundary
- B. To specify the query method name
- C. To enable pagination
- D. To enable sorting
- **Answer: A**

Spring Security

46. What is the purpose of the HttpSecurity class in Spring Security?

- A. To configure HTTP security for a web application
- B. To authenticate users
- C. To authorize user access

- D. To encrypt passwords
- **Answer: A**

47. What is the purpose of the UserDetails interface in Spring Security?

- A. To represent user information
- B. To authenticate users
- C. To authorize user access
- D. To encrypt passwords
- **Answer: A**

Spring Boot

48. What is the purpose of the @RestController annotation in Spring Boot?

- A. To mark a class as a REST controller
- B. To enable Spring Boot auto-configuration
- C. To define the main class of a Spring Boot application
- D. All of the above
- **Answer: A**

49. What is the purpose of the @PathVariable annotation in Spring MVC?

- A. To bind a URI template variable to a method parameter
- B. To bind a request parameter to a method parameter
- C. To bind a request body to a method parameter
- D. To bind a request header to a method parameter
- **Answer: A**

Spring AOP

50. What is the purpose of the @Around advice in Spring AOP?

- A. To execute before a method
- B. To execute after a method
- C. To execute both before and after a method
- D. To execute on exceptions thrown by a method
- **Answer: C**

Spring Messaging (50 Questions)

36. **Which of the following is the primary purpose of Spring Messaging?**
    - A. To handle HTTP requests in Spring MVC
    - B. To send and receive messages in a messaging queue

- C. To handle database transactions
- D. To define Spring beans

Answer: B. To send and receive messages in a messaging queue

37. **Which protocol is used by Spring Messaging for messaging?**
    - A. HTTP
    - B. AMQP
    - C. WebSockets
    - D. Both B and C

Answer: D. Both B and C

38. **Which Spring module is used for working with messaging and queues?**
    - A. Spring JMS
    - B. Spring Data
    - C. Spring Web
    - D. Spring Integration

Answer: A. Spring JMS

39. **What is the default message broker in Spring WebSocket?**
    - A. RabbitMQ
    - B. ActiveMQ
    - C. SimpleMessageBroker
    - D. JmsTemplate

Answer: C. SimpleMessageBroker

40. **What does the @MessageMapping annotation do in Spring?**
    - A. Maps HTTP requests to a method
    - B. Maps WebSocket messages to a method
    - C. Maps JMS messages to a method
    - D. Maps Spring Security rules to a method

Answer: B. Maps WebSocket messages to a method

41. **Which of the following is used to send a message to a destination in Spring Messaging?**
    - A. @SendTo
    - B. @MessageSend
    - C. @MessageDestination
    - D. @SendMessage

Answer: A. @SendTo

42. **Which interface in Spring Messaging provides the functionality for sending and receiving messages?**

    - A. MessageChannel
    - B. MessageQueue
    - C. JmsTemplate
    - D. WebSocketHandler

Answer: A. MessageChannel

43. **What is the default configuration for a message broker in Spring WebSocket?**

    - A. Stomp
    - B. RabbitMQ
    - C. JMS
    - D. AMQP

Answer: A. Stomp

44. **In Spring Messaging, which annotation is used to receive messages?**

    - A. @MessageListener
    - B. @ReceiveMessage
    - C. @MessageMapping
    - D. @Listen

Answer: C. @MessageMapping

45. **Which of the following is true about the Spring Integration module?**

    - A. It is used for messaging and integration with various protocols
    - B. It only supports JMS
    - C. It is used for defining REST endpoints
    - D. It is used only for WebSocket communication

Answer: A. It is used for messaging and integration with various protocols

---

Spring Boot Starters (50 Questions)

46. **What is the purpose of Spring Boot starters?**

    - A. They are used to add dependencies for specific functionalities to your project
    - B. They configure Spring beans automatically
    - C. They provide security for Spring Boot applications
    - D. They are used to run Spring Boot applications

Answer: A. They are used to add dependencies for specific functionalities to your project

47. **Which of the following is a commonly used Spring Boot starter for web applications?**

- A. spring-boot-starter-web
- B. spring-boot-starter-messaging
- C. spring-boot-starter-data-jpa
- D. spring-boot-starter-thymeleaf

Answer: A. spring-boot-starter-web

48. **Which Spring Boot starter is used to include support for embedded Tomcat server?**
    - A. spring-boot-starter-logging
    - B. spring-boot-starter-tomcat
    - C. spring-boot-starter-web
    - D. spring-boot-starter-jetty

Answer: B. spring-boot-starter-tomcat

49. **Which Spring Boot starter is used to add support for database connectivity?**
    - A. spring-boot-starter-data-jpa
    - B. spring-boot-starter-web
    - C. spring-boot-starter-thymeleaf
    - D. spring-boot-starter-logging

Answer: A. spring-boot-starter-data-jpa

50. **Which of the following starters is used for adding support for Spring Security?**
    - A. spring-boot-starter-security
    - B. spring-boot-starter-data
    - C. spring-boot-starter-actuator
    - D. spring-boot-starter-web

Answer: A. spring-boot-starter-security

51. **What is the role of spring-boot-starter-thymeleaf starter in a Spring Boot project?**
    - A. It adds support for HTML templating with Thymeleaf
    - B. It enables messaging features
    - C. It provides JPA integration
    - D. It is used for adding web sockets support

Answer: A. It adds support for HTML templating with Thymeleaf

52. **Which Spring Boot starter is used for building RESTful web services?**
    - A. spring-boot-starter-web
    - B. spring-boot-starter-rest
    - C. spring-boot-starter-jms

- D. spring-boot-starter-messaging

Answer: A. spring-boot-starter-web

53. **What does the spring-boot-starter-logging starter include?**
    - A. Logging configurations for applications
    - B. Database connection configurations
    - C. WebSocket configurations
    - D. Actuator configurations for monitoring

Answer: A. Logging configurations for applications

54. **Which Spring Boot starter would you use for integrating Spring Boot with Elasticsearch?**
    - A. spring-boot-starter-data-elasticsearch
    - B. spring-boot-starter-web
    - C. spring-boot-starter-data-mongodb
    - D. spring-boot-starter-thymeleaf

Answer: A. spring-boot-starter-data-elasticsearch

55. **Which starter is required to enable Spring Boot Actuator?**
    - A. spring-boot-starter-actuator
    - B. spring-boot-starter-web
    - C. spring-boot-starter-security
    - D. spring-boot-starter-data-jpa

Answer: A. spring-boot-starter-actuator

Spring MVC (50 Questions)

56. **What is the purpose of Spring MVC in the Spring Framework?**
    - A. To create database connections
    - B. To handle HTTP requests and responses
    - C. To manage application configurations
    - D. To implement messaging queues

Answer: B. To handle HTTP requests and responses

57. **Which of the following annotations is used to mark a class as a Spring MVC controller?**
    - A. @Controller
    - B. @Service
    - C. @RestController
    - D. @Bean

Answer: A. @Controller

58. **Which of the following annotations is used to define a RESTful controller in Spring MVC?**
    - A. @RestController
    - B. @Controller
    - C. @Service
    - D. @Component

Answer: A. @RestController

59. **What is the role of the @RequestMapping annotation in Spring MVC?**
    - A. To map HTTP requests to controller methods
    - B. To specify the HTTP method (GET, POST, etc.)
    - C. To define response formats (JSON, XML)
    - D. All of the above

Answer: D. All of the above

60. **Which of the following methods in a Spring controller is used to handle HTTP GET requests?**
    - A. @RequestMapping(method = RequestMethod.GET)
    - B. @PostMapping
    - C. @GetMapping
    - D. @PutMapping

Answer: C. @GetMapping

61. **What is the purpose of @ModelAttribute in Spring MVC?**
    - A. To bind request parameters to model attributes
    - B. To handle HTTP POST requests
    - C. To define a view for a controller
    - D. To handle form submissions

Answer: A. To bind request parameters to model attributes

62. **In Spring MVC, which annotation is used to specify a path variable in a URL?**
    - A. @RequestParam
    - B. @PathVariable
    - C. @RequestMapping
    - D. @RequestBody

Answer: B. @PathVariable

63. **What is the default view resolver in Spring MVC?**
    - A. InternalResourceViewResolver

- B. ThymeleafViewResolver
- C. JSPViewResolver
- D. FreeMarkerViewResolver

Answer: A. InternalResourceViewResolver

64. **Which annotation is used to handle form submissions in Spring MVC?**
    - A. @RequestMapping
    - B. @PostMapping
    - C. @GetMapping
    - D. @SubmitMapping

Answer: B. @PostMapping

Here are additional MCQs based on your specified topics:

Dependency Injection (50 Questions)

65. **Which of the following is the default scope for Spring beans?**

- A. Singleton
- B. Prototype
- C. Request
- D. Session

Answer: A. Singleton

66. **What is the main purpose of Dependency Injection (DI) in Spring?**

- A. To manage database connections
- B. To manage application configurations
- C. To decouple class dependencies and manage object creation
- D. To handle HTTP requests

Answer: C. To decouple class dependencies and manage object creation

67. **Which of the following annotations is used to declare a Spring bean in a class?**

- A. @Bean
- B. @Component
- C. @Controller
- D. @Service

Answer: B. @Component

68. **Which annotation in Spring is used to inject dependencies into a class?**

- A. @Inject

- B. @Autowired
- C. @InjectDependencies
- D. @AutowiredBean

Answer: B. @Autowired

69. **What does the @Qualifier annotation do in Spring?**

- A. It injects dependencies into a class
- B. It specifies which bean to inject when multiple beans are available
- C. It defines the scope of a bean
- D. It provides transactional support for beans

Answer: B. It specifies which bean to inject when multiple beans are available

70. **Which of the following is a type of dependency injection in Spring?**

- A. Constructor Injection
- B. Setter Injection
- C. Field Injection
- D. All of the above

Answer: D. All of the above

71. **Which annotation is used for autowiring a constructor in Spring?**

- A. @Autowired
- B. @Inject
- C. @Bean
- D. @AutowiredConstructor

Answer: A. @Autowired

72. **In Spring, what does the @Primary annotation do?**

- A. Marks the bean as the default bean when multiple candidates exist
- B. Defines the primary database connection
- C. Marks the primary configuration class
- D. Marks the primary service class

Answer: A. Marks the bean as the default bean when multiple candidates exist

73. **What is the main difference between @Autowired and @Inject in Spring?**

- A. @Autowired is specific to Spring, while @Inject is a part of Java EE
- B. @Autowired can inject fields, while @Inject cannot
- C. @Autowired is used for constructor injection, while @Inject is used for setter injection

- D. @Autowired supports optional injection, while @Inject does not

Answer: A. @Autowired is specific to Spring, while @Inject is a part of Java EE

74. **Which of the following is NOT a valid type of Spring bean scope?**

- A. Singleton
- B. Prototype
- C. Application
- D. Request

Answer: C. Application

75. **In Spring, what happens if no qualifying bean is found for autowiring?**

- A. An exception is thrown
- B. A default bean is created
- C. The bean will be null
- D. The application stops

Answer: A. An exception is thrown

Spring Task Scheduler and Cron Jobs (50 Questions)

76. **Which annotation is used to enable task scheduling in a Spring application?**

- A. @EnableScheduling
- B. @EnableTasks
- C. @ScheduledTasks
- D. @TaskScheduling

Answer: A. @EnableScheduling

77. **What is the purpose of the @Scheduled annotation in Spring?**

- A. To schedule tasks to run periodically
- B. To handle cron jobs
- C. To configure data source
- D. To configure a message broker

Answer: A. To schedule tasks to run periodically

78. **Which of the following cron expression represents the task to run every day at midnight in Spring?**

- A. 0 0 12 * * ?
- B. 0 0 0 * * ?
- C. 0 0 23 * * ?

- D. 0 0 0 1 * ?

Answer: B. 0 0 0 * * ?

79. **Which method in TaskScheduler is used to schedule a task?**

- A. schedule()
- B. runTask()
- C. scheduleAtFixedRate()
- D. scheduleTask()

Answer: A. schedule()

80. **Which of the following is a valid cron expression format used in Spring's @Scheduled annotation?**

- A. * * * * *
- B. Seconds Minutes Hours Day-of-month Month Day-of-week Year
- C. 0-59 0-59 0-23 1-31 1-12 0-6
- D. Seconds Minutes Hours Day-of-month Month

Answer: B. Seconds Minutes Hours Day-of-month Month Day-of-week Year

81. **What does the fixedRate attribute in the @Scheduled annotation do?**

- A. Specifies the frequency of task execution in milliseconds
- B. Executes the task after a fixed interval from the last task's start time
- C. Specifies a fixed delay between the completion of one execution and the start of the next
- D. None of the above

Answer: B. Executes the task after a fixed interval from the last task's start time

82. **What is the purpose of the fixedDelay attribute in the @Scheduled annotation?**

- A. It specifies the delay between the start of one execution and the start of the next
- B. It defines the interval after which a task is executed
- C. It defines how often a task is executed in seconds
- D. None of the above

Answer: A. It specifies the delay between the start of one execution and the start of the next

83. **Which of the following is a valid cron expression to run a task every hour?**

- A. 0 0 * * * ?
- B. 0 * * * * ?
- C. * 0 * * * ?
- D. 0 0 0 * * ?

Answer: A. 0 0 * * * ?

84. **Which interface does Spring use for task scheduling?**

- A. TaskExecutor
- B. TaskScheduler
- C. Runnable
- D. ScheduledExecutorService

Answer: B. TaskScheduler

85. **Which of the following annotations is used to enable the Spring scheduling functionality for cron jobs?**

- A. @EnableScheduling
- B. @ScheduledTasks
- C. @CronEnabled
- D. @TaskScheduling

Answer: A. @EnableScheduling

Authorization and Authentication (50 Questions)

86. **Which Spring Security annotation is used to specify a method should be accessible only by authenticated users?**

- A. @Secured
- B. @PreAuthorize
- C. @PostAuthorize
- D. @Authenticate

Answer: A. @Secured

87. **In Spring Security, what is the default role for authenticated users?**

- A. ROLE_USER
- B. ROLE_ADMIN
- C. ROLE_GUEST
- D. ROLE_AUTHENTICATED

Answer: A. ROLE_USER

88. **Which Spring Security class is responsible for managing authentication?**

- A. AuthenticationManager
- B. SecurityContextHolder
- C. AuthenticationProvider
- D. SecurityManager

Answer: A. AuthenticationManager

89. **Which annotation in Spring Security is used to restrict access to a method to a specific role?**

- A. @PreAuthorize
- B. @Secured
- C. @RolesAllowed
- D. @PermitAll

Answer: B. @Secured

90. **Which interface is used by Spring Security to retrieve user details for authentication?**

- A. UserDetailsService
- B. AuthenticationProvider
- C. UserService
- D. UserRepository

Answer: A. UserDetailsService

91. **What is the purpose of the SecurityContextHolder in Spring Security?**

- A. To hold the security context of the current request
- B. To handle user registration
- C. To store user credentials
- D. To manage roles and permissions

Answer: A. To hold the security context of the current request

92. **Which Spring Security feature allows the user to authenticate using JWT (JSON Web Token)?**

- A. HttpSecurity
- B. JwtAuthenticationFilter
- C. JWTTokenFilter
- D. OAuth2Login

Answer: B. JwtAuthenticationFilter

93. **What does the @PreAuthorize annotation in Spring Security allow?**

- A. It allows the user to authorize method execution based on roles
- B. It prevents unauthorized users from accessing a method
- C. It checks the user’s permissions dynamically
- D. All of the above

Answer: D. All of the above

94. **Which class in Spring Security manages the user session for an HTTP request?**

- A. SecurityContextHolder
- B. HttpSessionSecurityContextRepository
- C. AuthenticationManager
- D. SessionManager

Answer: B. HttpSessionSecurityContextRepository

Spring Data

51. What is the purpose of the @EnableJpaRepositories annotation?

- A. To enable JPA repositories in a Spring Boot application
- B. To specify the base package for JPA repositories
- C. To configure the JPA entity manager
- D. All of the above
- **Answer: D**

Spring Security

52. What is the purpose of the @PreAuthorize annotation?

- A. To authorize access to a method before execution
- B. To authorize access to a method after execution
- C. To log all method invocations
- D. To cache method invocations
- **Answer: A**

Spring Boot

53. What is the purpose of the @SpringBootApplication annotation?

- A. To mark a class as a Spring Boot application
- B. To enable Spring Boot auto-configuration
- C. To define the main class of a Spring Boot application
- D. All of the above
- **Answer: D**

Spring AOP

54. What is the purpose of the @AfterThrowing advice?

- A. To execute before a method
- B. To execute after a method
- C. To execute both before and after a method
- D. To execute on exceptions thrown by a method

- **Answer: D**

Spring Cloud

55. What is the purpose of Spring Cloud Config Server?

- A. To centralize configuration management
- B. To provide service discovery
- C. To implement circuit breakers
- D. To provide API gateway functionality
- **Answer: A**

Dependency Injection (Coding MCQs)

1. **What is the correct way to inject a dependency in Spring using annotation-based configuration?**
   - A. @Inject
   - B. @Autowired
   - C. @InjectBean
   - D. @Dependency

Answer: B. @Autowired

2. **Which annotation can be used to create a Spring bean that will be automatically injected into other classes?**
   - A. @Component
   - B. @Bean
   - C. @Service
   - D. @Repository

Answer: A. @Component

3. **In Spring, which of the following is used to configure a bean explicitly?**
   - A. @ComponentScan
   - B. @Configuration
   - C. @Bean
   - D. @Repository

Answer: C. @Bean

4. **Which of the following methods is used for dependency injection via constructor in Spring?**
   - A. Using @Autowired on a constructor
   - B. Using @Inject on a constructor
   - C. Both A and B
   - D. Neither A nor B

Answer: C. Both A and B

Spring Task Scheduler and Cron Jobs (Coding MCQs)

5. **Which annotation is used to schedule tasks in Spring Boot?**
   - A. @Scheduled
   - B. @TaskScheduled
   - C. @CronTask
   - D. @Task

Answer: A. @Scheduled

6. **Which of the following cron expressions would run a task every day at midnight?**
   - A. 0 0 12 * * *
   - B. 0 0 0 * * *
   - C. 0 0 0 1 * *
   - D. 0 0 12 * * *

Answer: B. 0 0 0 * * *

7. **What is the default cron expression for @Scheduled(cron = "") if not specified?**
   - A. It runs the task immediately
   - B. It runs every 10 seconds
   - C. It runs every minute
   - D. It runs according to the system's time

Answer: D. It runs according to the system's time

8. **Which of the following annotations enables Spring's scheduling support?**
   - A. @EnableScheduling
   - B. @TaskScheduler
   - C. @ScheduledTask
   - D. @ScheduledTasks

Answer: A. @EnableScheduling

Authorization and Authentication (Coding MCQs)

9. **Which interface is used for user authentication in Spring Security?**
   - A. AuthenticationManager
   - B. UserDetailsService
   - C. SecurityContextHolder

  - D. AuthorizationManager

Answer: B. UserDetailsService

10. **Which annotation is used to secure methods in Spring Security?**
    - A. @Secure
    - B. @PreAuthorize
    - C. @Authorize
    - D. @AccessControl

Answer: B. @PreAuthorize

11. **Which method in Spring Security's HttpSecurity class is used to configure form-based login?**
    - A. http.formLogin()
    - B. http.loginForm()
    - C. http.formAuthentication()
    - D. http.setLoginForm()

Answer: A. http.formLogin()

12. **In Spring Security, which class is used to authenticate the current user via JWT token?**
    - A. JwtAuthenticationFilter
    - B. UsernamePasswordAuthenticationFilter
    - C. SecurityContextFilter
    - D. JwtAuthenticationManager

Answer: A. JwtAuthenticationFilter

Spring Events and Event Listeners (Coding MCQs)

13. **Which annotation is used to publish an event in Spring?**
    - A. @EventPublisher
    - B. @PublishEvent
    - C. @EventListener
    - D. @TransactionalEventListener

Answer: A. @EventPublisher

14. **What method is used to listen for an event in a Spring application?**
    - A. @HandleEvent
    - B. @EventListener
    - C. @ListenToEvent
    - D. @EventHandling

Answer: B. @EventListener

15. **Which of the following annotations can be used to handle Spring application events asynchronously?**
    - A. @Async
    - B. @EventListener(value = ApplicationEvent.class, async = true)
    - C. @AsyncEventListener
    - D. @AsyncHandler

Answer: B. @EventListener(value = ApplicationEvent.class, async = true)

16. **In Spring, how can you publish a custom event?**
    - A. Using ApplicationContext.publishEvent()
    - B. Using EventPublisher.publish()
    - C. Using EventBus.publish()
    - D. Using @EventPublisher.publish()

Answer: A. Using ApplicationContext.publishEvent()

Spring Messaging (Coding MCQs)

17. **Which of the following is used for messaging in Spring?**
    - A. Spring JMS
    - B. Spring Integration
    - C. Spring AMQP
    - D. All of the above

Answer: D. All of the above

18. **What annotation is used to listen to a JMS queue in Spring?**
    - A. @JmsListener
    - B. @QueueListener
    - C. @MessageListener
    - D. @Listener

Answer: A. @JmsListener

19. **Which Spring framework component provides message-driven POJOs?**
    - A. @Service
    - B. @MessageListener
    - C. @JmsListener
    - D. @Component

Answer: C. @JmsListener

20. **Which annotation is used to configure the sending of messages to a JMS destination in Spring?**
    - A. @SendTo
    - B. @MessageMapping
    - C. @JmsTemplate
    - D. @RabbitTemplate

Answer: A. @SendTo

Spring Boot Starters (Coding MCQs)

21. **Which of the following is a common Spring Boot starter used for building web applications?**
    - A. spring-boot-starter-web
    - B. spring-boot-starter-data-jpa
    - C. spring-boot-starter-security
    - D. spring-boot-starter-mail

Answer: A. spring-boot-starter-web

22. **Which Spring Boot starter is used for working with databases via JPA?**
    - A. spring-boot-starter-jdbc
    - B. spring-boot-starter-data-jpa
    - C. spring-boot-starter-data-mongodb
    - D. spring-boot-starter-h2

Answer: B. spring-boot-starter-data-jpa

23. **Which Spring Boot starter is used for sending emails?**
    - A. spring-boot-starter-email
    - B. spring-boot-starter-mail
    - C. spring-boot-starter-smtp
    - D. spring-boot-starter-gmail

Answer: B. spring-boot-starter-mail

24. **Which Spring Boot starter is used to enable Spring Security in a project?**
    - A. spring-boot-starter-security
    - B. spring-boot-starter-auth
    - C. spring-boot-starter-login
    - D. spring-boot-starter-authorization

Answer: A. spring-boot-starter-security

Spring MVC (Coding MCQs)

25. **Which annotation is used to handle HTTP GET requests in Spring MVC?**
    - A. @RequestMapping
    - B. @GetMapping
    - C. @RequestParam
    - D. @PostMapping

Answer: B. @GetMapping

26. **Which of the following is used to bind a form submission to a model object in Spring MVC?**
    - A. @ModelAttribute
    - B. @RequestBody
    - C. @RequestParam
    - D. @ModelForm

Answer: A. @ModelAttribute

27. **Which annotation is used in Spring MVC to map a POST request to a method?**
    - A. @RequestMapping
    - B. @PostMapping
    - C. @RequestBody
    - D. @RequestParam

Answer: B. @PostMapping

28. **Which annotation in Spring MVC is used to handle HTTP requests based on method type and URI pattern?**
    - A. @RequestMapping
    - B. @RequestMethod
    - C. @GetMapping
    - D. @UrlMapping

Answer: A. @RequestMapping

SBQ questions

1. Dependency Injection (DI)

1. **What is Dependency Injection in Spring?**
   - *Answer*: Dependency Injection is a design pattern where an object's dependencies (i.e., the objects it needs) are provided to it rather than the object creating them. In Spring, DI is used to reduce tight coupling between components.
2. **What are the different types of Dependency Injection in Spring?**
   - *Answer*: The three types of DI in Spring are:
     - **Constructor-based**: Dependencies are provided through the constructor.
     - **Setter-based**: Dependencies are provided via setter methods.
     - **Field-based**: Dependencies are injected directly into fields using @Autowired annotation.
3. **What is the role of @Autowired annotation in Spring?**
   - *Answer*: The @Autowired annotation is used to automatically inject dependencies into a Spring bean. It can be used on fields, constructors, or setter methods.
4. **Explain the difference between Singleton and Prototype bean scopes in Spring.**
   - *Answer*:
     - **Singleton**: A single instance of the bean is created and shared across the entire Spring container.
     - **Prototype**: A new instance of the bean is created every time it is requested from the container.

2. Spring Task Scheduler and Cron Jobs

5. **How does Spring Task Scheduler work?**
   - *Answer*: Spring Task Scheduler allows you to schedule tasks to run at specified intervals or at specific times. It uses annotations like @Scheduled to configure tasks.
6. **What is the role of the @Scheduled annotation in Spring?**
   - *Answer*: The @Scheduled annotation is used to schedule tasks to run at fixed intervals, after a fixed delay, or at specific times using cron expressions.
7. **What is a cron expression, and how is it used in Spring?**
   - *Answer*: A cron expression is a string representing a schedule for running a task. It is used in Spring to define the time intervals for scheduled tasks in the @Scheduled annotation.
8. **How can you schedule a task to run every Monday at 8 AM in Spring?**
   - *Answer*: You can use the @Scheduled annotation with a cron expression like this:

java

Copy code

```
@Scheduled(cron = "0 0 8 * * MON")
public void task() {
   // Task logic
}
```

3. Authorization and Authentication

9. **What is the difference between Authentication and Authorization?**
   - *Answer*:
     - **Authentication**: The process of verifying the identity of a user or system.
     - **Authorization**: The process of granting or denying access to resources based on the authenticated identity.
10. **How can you implement authentication in Spring Boot?**
    - *Answer*: Authentication in Spring Boot is typically implemented using Spring Security, which can be configured with in-memory authentication, database-backed authentication, or third-party authentication providers like OAuth2 or JWT.
11. **What is the purpose of the @PreAuthorize annotation in Spring Security?**
    - *Answer*: The @PreAuthorize annotation is used to define method-level security in Spring, allowing access to methods based on the user's roles or permissions.
12. **How can you configure role-based access control in Spring Security?**
    - *Answer*: Role-based access control can be configured in Spring Security using the @Secured or @PreAuthorize annotations, along with appropriate role checks. Roles can be granted to users during authentication.

4. Spring Events and Event Listeners

13. **What is the purpose of Spring Events?**
    - *Answer*: Spring Events allow communication between components in a decoupled manner. Event publishers send events, and event listeners handle them asynchronously.
14. **What is the role of the @EventListener annotation in Spring?**
    - *Answer*: The @EventListener annotation is used to mark methods as listeners for specific Spring application events, allowing them to react to events when they occur.
15. **How do you publish an event in Spring?**
    - *Answer*: Events are published in Spring using the ApplicationEventPublisher interface, which can be autowired into a component and used to publish events.

16. **What is the difference between @EventListener and ApplicationListener in Spring?**
    - *Answer*:
        - **@EventListener**: An annotation-based approach to handle events.
        - **ApplicationListener**: An interface-based approach to handle events, where the class implements the onApplicationEvent method.

5. Spring Messaging

17. **What is Spring Messaging?**
    - *Answer*: Spring Messaging provides support for messaging protocols like JMS, AMQP, and WebSocket, allowing Spring applications to send and receive messages.
18. **What is the use of @MessageMapping annotation in Spring WebSocket?**
    - *Answer*: The @MessageMapping annotation is used to map WebSocket messages to handler methods in a Spring controller, similar to how @RequestMapping maps HTTP requests.
19. **How can you configure message listeners in Spring JMS?**
    - *Answer*: Message listeners in Spring JMS can be configured using the @JmsListener annotation or by implementing the MessageListener interface and configuring it in the listener container.
20. **What is the role of the JmsTemplate in Spring?**
    - *Answer*: JmsTemplate is a helper class in Spring that simplifies sending and receiving JMS messages. It provides a higher-level abstraction over the JMS API.

6. Spring Boot Starters

21. **What are Spring Boot starters?**
    - *Answer*: Spring Boot starters are pre-configured dependencies provided by Spring Boot to simplify the setup of various features, such as web development, data access, and messaging.
22. **Name some commonly used Spring Boot starters.**
    - *Answer*: Some commonly used Spring Boot starters include:
        - spring-boot-starter-web (for building web applications)
        - spring-boot-starter-data-jpa (for JPA and Hibernate integration)
        - spring-boot-starter-thymeleaf (for Thymeleaf templating engine)
        - spring-boot-starter-security (for security features)
23. **What is the purpose of spring-boot-starter-test?**
    - *Answer*: The spring-boot-starter-test provides libraries and dependencies for testing Spring Boot applications, including JUnit, Mockito, Hamcrest, and Spring Test.

24. **How can you exclude a starter dependency in Spring Boot?**

    - *Answer*: You can exclude a starter dependency by using the exclude attribute in the @SpringBootApplication annotation or in the pom.xml file.

7. Spring MVC

25. **What is Spring MVC?**

    - *Answer*: Spring MVC (Model-View-Controller) is a web framework in Spring that provides a structure for building web applications by separating the business logic (Model), user interface (View), and user input (Controller).

26. **How do you map a URL to a controller method in Spring MVC?**

    - *Answer*: In Spring MVC, you use the @RequestMapping annotation on a controller method to map a URL to the method. You can also use specialized annotations like @GetMapping, @PostMapping, etc.

27. **What is the difference between @Controller and @RestController in Spring MVC?**

    - *Answer*:

        - @Controller: Used to define a controller class that handles web requests and returns view names (JSP, Thymeleaf, etc.).

        - @RestController: A combination of @Controller and @ResponseBody, used for building RESTful web services that return data in formats like JSON or XML.

28. **How can you pass data from a controller to a view in Spring MVC?**

    - *Answer*: Data can be passed to a view in Spring MVC using Model or ModelAndView objects, which are added in the controller method using the addAttribute method.

1. Dependency Injection (DI)

29. **What is the purpose of the @Qualifier annotation in Spring?**

- *Answer*: The @Qualifier annotation is used to specify which bean to inject when there are multiple candidates for autowiring, helping Spring to resolve ambiguity.

30. **What is a "bean" in Spring Framework?**

- *Answer*: A "bean" is an object that is managed by the Spring container. Beans are created, configured, and wired by Spring IoC container.

31. **What are the advantages of Dependency Injection?**

- *Answer*: Dependency Injection promotes loose coupling between components, improves testability, and makes the application more maintainable and flexible.

32. **What is the difference between @Autowired and @Inject annotations in Spring?**

- *Answer*: Both annotations are used for dependency injection, but @Autowired is Spring-specific, while @Inject is part of the Java standard (javax.inject) and has similar functionality.

33. **How does Spring handle circular dependencies?**

- *Answer*: Spring handles circular dependencies by using setter injection or proxy-based approaches. Constructor injection is not feasible for circular dependencies in Spring.

2. Spring Task Scheduler and Cron Jobs

34. **How do you configure a fixed delay schedule for a task in Spring?**

- *Answer*: You can use the @Scheduled annotation with the fixedDelay attribute, which specifies the delay between the completion of the last execution and the start of the next execution.

java

Copy code

```
@Scheduled(fixedDelay = 5000) // Delay 5 seconds
public void task() {
  // Task logic
}
```

35. **What is the difference between fixedRate and fixedDelay in Spring scheduling?**

- *Answer*:
    - **fixedRate**: Executes the task at a fixed rate, irrespective of when the last task finished.
    - **fixedDelay**: Executes the task after a specified delay from the end of the previous task.

36. **Can you schedule a task in Spring Boot based on cron expressions?**

- *Answer*: Yes, Spring Boot allows scheduling tasks using cron expressions with the @Scheduled(cron = "expression") annotation. Cron expressions allow fine-grained control over task scheduling.

37. **What is the use of the TaskScheduler interface in Spring?**

- *Answer*: TaskScheduler is an interface in Spring that provides methods for scheduling tasks with flexible scheduling options. It is an abstraction over different scheduling frameworks like ScheduledExecutorService.

38. **What is the default time unit for fixedDelay and fixedRate in Spring Task Scheduling?**

- *Answer*: The default time unit for both fixedDelay and fixedRate is milliseconds.

3. Authorization and Authentication

39. **How does Spring Security handle password hashing?**

- *Answer*: Spring Security supports password hashing through the PasswordEncoder interface, which provides implementations like BCryptPasswordEncoder, NoOpPasswordEncoder, etc.

40. **What is the role of JWT (JSON Web Token) in Spring Security?**

- *Answer*: JWT is used for stateless authentication, where a token is generated after a user logs in and is included in subsequent requests to authenticate the user.

41. **What is the difference between @Secured and @PreAuthorize in Spring Security?**

- *Answer*:
    - **@Secured**: Specifies which roles can access a method.
    - **@PreAuthorize**: Provides more flexibility, allowing you to use SpEL (Spring Expression Language) to define authorization conditions.

42. **What is OAuth2, and how is it used in Spring Security?**

- *Answer*: OAuth2 is an authorization framework that allows third-party applications to access a user's data without exposing their credentials. Spring Security OAuth2 supports client-side and server-side OAuth2 authentication.

43. **How can you restrict access to a URL in Spring Security based on roles?**

- *Answer*: Access to a URL can be restricted using http.authorizeRequests().antMatchers("/restricted").hasRole("ADMIN") to allow access only for users with the ADMIN role.

4. Spring Events and Event Listeners

44. **What is the ApplicationEventPublisher interface used for in Spring?**

- *Answer*: ApplicationEventPublisher is used to publish events to the Spring application context, which can then be handled by any listeners registered for the event.

45. **What is the use of @Async with event listeners in Spring?**

- *Answer*: @Async is used to run event listeners asynchronously, ensuring that the event processing does not block the main thread.

46. **How can you listen to an event in Spring without using @EventListener?**

- *Answer*: You can implement the ApplicationListener interface and override the onApplicationEvent method to listen to events.

47. **What are some common use cases for Spring events?**

- *Answer*: Common use cases include handling post-save actions (e.g., sending notifications after data is saved), triggering background processes, and reacting to changes in the application state.

48. **How can you use custom events in Spring?**

- *Answer*: Custom events can be created by extending the ApplicationEvent class and then publishing them using the ApplicationEventPublisher.

5. Spring Messaging

49. **What is Spring AMQP?**

- *Answer*: Spring AMQP (Advanced Message Queuing Protocol) is a messaging framework that provides support for message-driven POJOs with RabbitMQ as the underlying messaging platform.

50. **What is the role of @MessageMapping in Spring WebSocket?**

- *Answer*: @MessageMapping is used in Spring WebSocket to map incoming WebSocket messages to methods, similar to how @RequestMapping maps HTTP requests to controller methods.

51. **What is the purpose of @SendTo in Spring WebSocket?**

- *Answer*: @SendTo is used in Spring WebSocket to define the destination to which the response should be sent after processing a message. It works in combination with @MessageMapping.

52. **How does Spring JMS integrate with ActiveMQ?**

- *Answer*: Spring JMS provides integration with ActiveMQ through the JmsTemplate and @JmsListener annotation to send and receive messages in a Java Message Service (JMS) environment.

53. **What is the role of RabbitTemplate in Spring AMQP?**

- *Answer*: RabbitTemplate is a helper class in Spring AMQP for sending and receiving messages to and from RabbitMQ queues.

6. Spring Boot Starters

54. **What is spring-boot-starter-web used for?**

- *Answer*: spring-boot-starter-web is used to develop web applications in Spring Boot. It includes dependencies for Spring MVC, embedded Tomcat server, and Jackson for JSON processing.

55. **What is spring-boot-starter-data-jpa used for?**

- *Answer*: spring-boot-starter-data-jpa provides the required dependencies to integrate Spring Data JPA with Hibernate for database interaction.

56. **How does spring-boot-starter-logging help in Spring Boot applications?**

- *Answer*: spring-boot-starter-logging automatically configures a logging framework (such as Logback) for logging purposes in Spring Boot applications.

57. **What is spring-boot-starter-thymeleaf used for?**

- *Answer*: spring-boot-starter-thymeleaf provides the necessary dependencies for using Thymeleaf as a templating engine in Spring Boot applications.

58. **What is the purpose of spring-boot-starter-actuator?**

- *Answer*: spring-boot-starter-actuator provides production-ready features such as health checks, metrics, and application monitoring in Spring Boot applications.

7. Spring MVC

59. **What is the role of @RequestMapping in Spring MVC?**

- *Answer*: @RequestMapping is used to map web requests to specific handler methods in a Spring controller.

60. **What is the difference between @GetMapping and @PostMapping in Spring MVC?**

- *Answer*:
  - @GetMapping is used to handle HTTP GET requests.
  - @PostMapping is used to handle HTTP POST requests.

61. **What is the role of ModelAndView in Spring MVC?**

- *Answer*: ModelAndView is used in Spring MVC to return both the model (data) and the view (view name or template) from a controller method.

62. **What is the purpose of @PathVariable in Spring MVC?**

- *Answer*: @PathVariable is used to bind a method parameter to a URI template variable in Spring MVC.

63. **How can you perform form handling in Spring MVC?**

- *Answer*: Form handling in Spring MVC can be performed by using the @ModelAttribute annotation to bind form data to an object and handling form submissions in controller methods with @PostMapping.

1. Dependency Injection (DI)

64. **What is the difference between constructor injection and setter injection in Spring?**

- *Answer*:
  - **Constructor Injection**: Dependencies are provided through the constructor of the bean, ensuring immutability and making dependencies mandatory.
  - **Setter Injection**: Dependencies are provided through setter methods, which allows optional dependencies and can be changed after object creation.

65. **How does Spring manage the lifecycle of beans?**

- *Answer*: Spring manages the lifecycle of beans by calling lifecycle methods like @PostConstruct and @PreDestroy after initializing and before destroying the bean, respectively.

66. **What is the @Primary annotation used for in Spring?**

- *Answer*: The @Primary annotation is used to indicate that a particular bean should be used when multiple candidates are available for autowiring.

67. **What is @Value used for in Spring?**

- *Answer*: The @Value annotation is used to inject values from property files, environment variables, or expression language (SpEL) into Spring beans.

68. **What is the scope of a singleton bean in Spring?**

- *Answer*: A singleton bean in Spring is created only once per Spring IoC container. It is shared across the entire application context.

2. Spring Task Scheduler and Cron Jobs

69. **How can you run a task in Spring at fixed intervals using the @Scheduled annotation?**

- *Answer*: You can specify intervals using the fixedRate or fixedDelay attributes of the @Scheduled annotation, or use a cron expression for more advanced scheduling.

70. **What is a cron expression, and how is it used in Spring?**

- *Answer*: A cron expression is a string that defines a schedule for running tasks. It consists of six fields: second, minute, hour, day of the month, month, and day of the week. In Spring, it is used with the @Scheduled(cron = "expression") annotation to schedule tasks.

71. **How do you ensure that a task is executed periodically using Spring Task Scheduler?**

- *Answer*: You can use the @Scheduled annotation with the fixedRate or cron attribute, or use TaskScheduler to schedule tasks programmatically.

72. **How would you schedule a task to run every day at midnight using Spring?**

- *Answer*: You can use the following cron expression in @Scheduled:

java

Copy code

```
@Scheduled(cron = "0 0 0 * * ?")
public void task() {
   // Task logic
}
```

73. **What is the purpose of the @EnableScheduling annotation in Spring?**

- *Answer*: The @EnableScheduling annotation enables the Spring scheduler and allows you to run tasks asynchronously with the help of the @Scheduled annotation.

3. Authorization and Authentication

74. **What is the role of the AuthenticationManager in Spring Security?**

- *Answer*: The AuthenticationManager is responsible for authenticating a user’s credentials. It works by delegating to various authentication providers.

75. **What is the role of GrantedAuthority in Spring Security?**

- *Answer*: GrantedAuthority represents an authority granted to an Authentication object. It is used to define the roles or permissions assigned to the authenticated user.

76. **How can you configure HTTP Basic authentication in Spring Security?**

- *Answer*: You can configure HTTP Basic authentication by using http.httpBasic() in the Spring Security configuration class.

77. **What is the purpose of @PreAuthorize annotation in Spring Security?**

- *Answer*: @PreAuthorize is used to define pre-method authorization, allowing you to secure method invocations using Spring Expression Language (SpEL).

78. **What is the difference between Authentication and Authorization in Spring Security?**

- *Answer*:
    - **Authentication** is the process of verifying the identity of a user (e.g., username and password).
    - **Authorization** is the process of determining what actions a user is allowed to perform based on their roles or permissions.

4. Spring Events and Event Listeners

79. **What is the ApplicationEvent class used for in Spring?**

- *Answer*: ApplicationEvent is the base class for all events in Spring. You can extend it to create custom events that are published and listened to by Spring's event listeners.

80. **How can you publish an event in Spring?**

- *Answer*: You can publish an event using ApplicationEventPublisher, typically by calling publishEvent() within the application context.

java

Copy code

```
@Autowired
private ApplicationEventPublisher publisher;


public void publishCustomEvent() {
   publisher.publishEvent(new CustomEvent(this));
}
```

81. **What is the purpose of @EventListener in Spring?**

- *Answer*: @EventListener is used to define methods that listen for specific events within the Spring application context. The method will be invoked when the event is published.

82. **What is the ApplicationListener interface used for in Spring?**

- *Answer*: ApplicationListener is used to handle events in Spring by implementing the onApplicationEvent method, allowing the custom logic for event handling.

83. **How do you ensure that an event listener runs asynchronously in Spring?**

- *Answer*: You can use the @Async annotation to make an event listener asynchronous, allowing the event to be handled in a separate thread.

5. Spring Messaging

84. **What is the role of the @MessageMapping annotation in Spring WebSocket?**

- *Answer*: @MessageMapping is used to map incoming WebSocket messages to methods. It functions similarly to @RequestMapping for HTTP requests but for WebSocket messages.

85. **What is the difference between @SendTo and @MessageMapping in Spring WebSocket?**

- *Answer*:
    - **@MessageMapping**: Maps a WebSocket message to a controller method.
    - **@SendTo**: Specifies the destination where the response of a WebSocket message should be sent.

86. **What is Spring Integration, and how does it differ from Spring Messaging?**

- *Answer*: Spring Integration is a framework for building enterprise integration patterns, while Spring Messaging focuses on providing messaging support using protocols like WebSocket and JMS. Spring Integration can use Spring Messaging as part of its implementation.

87. **What is the purpose of JmsListener in Spring JMS?**

- *Answer*: @JmsListener is used to mark methods that will listen for JMS messages on a specific destination. The method is invoked automatically when a message is received.

88. **What is the purpose of RabbitTemplate in Spring AMQP?**

- *Answer*: RabbitTemplate is used to send and receive messages to and from RabbitMQ queues. It is the primary interface for interacting with RabbitMQ in Spring AMQP.

6. Spring Boot Starters

89. **What is spring-boot-starter-parent?**

- *Answer*: spring-boot-starter-parent is a special starter in Spring Boot that acts as a parent for all Spring Boot projects. It provides default configurations, dependency management, and plugin management.

90. **What dependencies are included in spring-boot-starter-web?**

- *Answer*: spring-boot-starter-web includes dependencies for Spring MVC, Tomcat (embedded server), Jackson (JSON processing), and validation libraries.

91. **What is the purpose of spring-boot-starter-validation?**

- *Answer*: spring-boot-starter-validation provides Hibernate Validator, which allows validation of beans in Spring Boot applications using annotations like @NotNull, @Size, etc.

92. **What is spring-boot-starter-mail used for?**

- *Answer*: spring-boot-starter-mail provides the required dependencies to send emails in Spring Boot applications via JavaMailSender.

93. **How does Spring Boot's spring-boot-starter-data-rest help in building REST APIs?**

- *Answer*: spring-boot-starter-data-rest automatically creates RESTful APIs for Spring Data repositories, enabling CRUD operations on entities without writing controller code.

7. Spring MVC

94. **What is the purpose of @ModelAttribute in Spring MVC?**

- *Answer*: @ModelAttribute is used to bind request parameters to a method argument or to add attributes to the model, which can be accessed by the view.

95. **How can you return a JSON response in Spring MVC?**

- *Answer*: You can return a JSON response by using the @RestController annotation or by using @ResponseBody on a controller method to indicate that the response body should be serialized as JSON.

96. **What is the role of @RequestBody in Spring MVC?**

- *Answer*: @RequestBody is used to bind the incoming HTTP request body to a method parameter, typically used for reading JSON or XML data in RESTful web services.

97. **How do you handle validation errors in Spring MVC?**

- *Answer*: You can use @Valid or @Validated annotations to validate incoming requests, and BindingResult can be used to capture validation errors for handling them.

98. **What is the purpose of @ResponseStatus in Spring MVC?**

- *Answer*: @ResponseStatus is used to specify the HTTP status code that should be returned for a specific method or exception handler in Spring MVC.

1. Dependency Injection (DI)

99. **What are the advantages of using Dependency Injection in Spring?**

- *Answer*: Dependency Injection provides loose coupling between classes, improves code maintainability, testability, and flexibility by allowing easy swapping of components.

100. **What is @Autowired annotation used for in Spring?**

- *Answer*: The @Autowired annotation is used to automatically wire a bean by type. Spring will automatically inject the required dependencies into the fields, constructors, or methods.

101. **How does Spring handle multiple beans of the same type for autowiring?**

- *Answer*: If multiple beans of the same type are found, Spring will throw an exception unless one of the beans is marked with @Primary or @Qualifier is used to specify which bean should be injected.

102. **Can you use @Autowired to inject primitive types or simple values?**

- *Answer*: Yes, @Autowired can inject primitive types, but it requires that the bean be a suitable type or the value be configured properly (via @Value for simple values).

103. **What is the role of ApplicationContext in Dependency Injection?**

- *Answer*: The ApplicationContext is responsible for managing the lifecycle of beans, resolving dependencies, and providing the beans to the application when needed.

2. Spring Task Scheduler and Cron Jobs

104. **How can you execute a task periodically in Spring using the @Scheduled annotation with a fixed delay?**

- *Answer*: You can use @Scheduled(fixedDelay = 5000) to execute a task with a fixed delay of 5000 milliseconds (5 seconds) after the previous execution finishes.

105. **What is the significance of the @EnableScheduling annotation in Spring?**

- *Answer*: The @EnableScheduling annotation enables the scheduling of tasks in a Spring application, allowing the use of the @Scheduled annotation to schedule tasks.

106. **How can you schedule a task to run at a fixed rate using Spring?**

- *Answer*: You can use the @Scheduled(fixedRate = 10000) annotation to run a task at a fixed rate of 10 seconds, regardless of the task's execution time.

107. **What is the use of @Scheduled(cron = "0 0 12 * * ?") in Spring?**

- *Answer*: The cron expression 0 0 12 * * ? schedules a task to run at 12:00 PM (noon) every day.

108. **How can you disable a scheduled task in Spring?**

- *Answer*: You can disable a scheduled task by commenting out or removing the @Scheduled annotation from the method or by using conditional checks within the method.

3. Authorization and Authentication

109. **What is the difference between stateless and stateful authentication?**

- *Answer*: Stateless authentication does not store any session or state between requests (e.g., JWT), while stateful authentication requires storing session information on the server side (e.g., HTTP sessions).

110. **How would you implement custom authentication in Spring Security?**

- *Answer*: You can implement custom authentication by creating a custom AuthenticationProvider and overriding the authenticate() method to perform the authentication logic.

111. **What is the role of SecurityContextHolder in Spring Security?**

- *Answer*: SecurityContextHolder holds the security context, including the authentication information of the currently authenticated user, which can be accessed throughout the application.

112. **What is remember-me authentication in Spring Security?**

- *Answer*: remember-me authentication allows users to stay logged in across sessions by storing a cookie on the client-side, which is used for re-authentication.

113. **How would you secure a REST API with JWT in Spring Security?**

- *Answer*: You can secure a REST API with JWT by creating a custom filter that intercepts incoming requests, validates the JWT token, and sets the authentication context.

4. Spring Events and Event Listeners

114. **How do you publish a custom event in Spring?**

- *Answer*: You can publish a custom event by using the ApplicationEventPublisher:

java

Copy code

```
@Autowired
private ApplicationEventPublisher publisher;


public void publishEvent() {
   publisher.publishEvent(new CustomEvent(this));
}
```

115. **How do you listen to events asynchronously in Spring?**

- *Answer*: You can listen to events asynchronously by annotating the listener method with @Async or by configuring an AsyncTaskExecutor.

116. **How can you filter the events in Spring?**

- *Answer*: You can filter events by using the ApplicationListener interface with custom logic or by specifying @EventListener with a condition.

117. **What is the role of @Async in event listeners?**

- *Answer*: The @Async annotation allows an event listener method to be executed asynchronously, freeing up the main thread to handle other requests.

118. **Can Spring events be used for cross-cutting concerns like logging?**

- *Answer*: Yes, Spring events can be used to handle cross-cutting concerns such as logging or auditing by publishing and listening for specific events across the application.

5. Spring Messaging

119. **What is the purpose of Spring AMQP (Advanced Message Queuing Protocol)?**

- *Answer*: Spring AMQP provides support for sending and receiving messages using AMQP-compliant brokers like RabbitMQ. It allows for message-driven application architecture.

120. **How would you configure a Spring application to use JMS (Java Message Service)?**

- *Answer*: To use JMS in Spring, you need to configure a JmsTemplate, define a ConnectionFactory, and set up destination queues or topics.

121. **What is the role of @JmsListener in Spring JMS?**

- *Answer*: @JmsListener is used to annotate methods that will listen for messages on a JMS destination, such as a queue or topic.

122. **How can you use Spring WebSocket for messaging in a Spring application?**

- *Answer*: Spring WebSocket allows for bidirectional communication between the client and server over a single, long-lived connection, enabling real-time messaging in web applications.

123. **What is the difference between @MessageMapping and @RequestMapping in Spring WebSocket?**

- *Answer*: @MessageMapping is used for mapping WebSocket messages to handler methods, while @RequestMapping is used for mapping HTTP requests to controller methods.

6. Spring Boot Starters

124. **What is the purpose of spring-boot-starter-data-jpa?**

- *Answer*: spring-boot-starter-data-jpa is used to set up the Spring Data JPA configuration, which simplifies the integration of JPA-based repositories in Spring Boot applications.

125. **What does spring-boot-starter-web include?**

- *Answer*: spring-boot-starter-web includes dependencies for building web applications with Spring MVC, embedded Tomcat, and support for RESTful services.

126. **What is spring-boot-starter-actuator used for?**

- *Answer*: spring-boot-starter-actuator adds production-ready features to Spring Boot applications, such as metrics, health checks, and application monitoring.

127. **How does spring-boot-starter-thymeleaf help in developing a web application?**

- *Answer*: spring-boot-starter-thymeleaf integrates Thymeleaf as the templating engine, allowing the development of dynamic web pages in Spring Boot applications.

128. **What is the spring-boot-starter-test used for?**

- *Answer*: spring-boot-starter-test includes libraries for testing Spring Boot applications, including JUnit, Mockito, and Spring TestContext Framework.

7. Spring MVC

129. **What is the difference between @RequestMapping and @GetMapping in Spring MVC?**

- *Answer*: @RequestMapping can handle all HTTP methods, while @GetMapping is a specialized version of @RequestMapping for HTTP GET requests.

130. **What is the purpose of @PathVariable in Spring MVC?**

- *Answer*: @PathVariable is used to extract values from URI path segments and bind them to method parameters in Spring MVC controller methods.

131. **How would you configure Spring MVC to return a view instead of JSON?**

- *Answer*: You can return a view (e.g., JSP or Thymeleaf template) by using the @Controller annotation and returning the view name from a controller method.

132. **What is @ResponseBody used for in Spring MVC?**

- *Answer*: @ResponseBody is used to indicate that the return value of a method should be written directly to the HTTP response body, typically as JSON or XML.

133. **What is the purpose of @ExceptionHandler in Spring MVC?**

- *Answer*: @ExceptionHandler is used to define methods that handle specific exceptions thrown by controller methods, allowing custom error handling in Spring MVC.