**Topics Covered:**

1. Spring Fundamentals – 25 MCQs
2. Spring Boot – 40 MCQs
3. Spring Security – 25 MCQs
4. Spring Micro Services – 40 MCQs

**Spring Fundamentals:** (25)

Questions covering fundamental topics in Spring, including dependency injection, bean lifecycle, annotations, Spring MVC, AOP, and more.

1. Which of the following best describes the core concept of dependency injection in Spring?

**a) Injecting dependencies into classes**

b) Injecting dependencies into interfaces

c) Injecting dependencies into controllers

d) Injecting dependencies into service classes

2. In Spring, what is the purpose of the ApplicationContext interface?

a) To provide configuration metadata to the Spring container

**b) To manage the lifecycle of beans in the Spring container**

c) To define the beans and their dependencies in the Spring container

d) To control the flow of control in the Spring application

3. What is the role of the @Autowired annotation in Spring?

a) It indicates that a bean should be autowired by name.

**b) It indicates that a bean should be autowired by type.**

c) It indicates that a bean should be autowired by constructor.

d) It indicates that a bean should be autowired by property.

4. Which of the following is not a valid scope for Spring beans?

a) Singleton

b) Prototype

c) Session

**d) Transient**

5. What is the purpose of the @Component annotation in Spring?

a) To define a bean

**b) To mark a class as a component for component scanning**

c) To define a controller

d) To mark a class as a service component

6. Which of the following files is used for configuring Spring beans using Java-based configuration?

a) applicationContext.xml

b) beans.xml

c) spring-config.xml

**d) AppConfig.java**

7. What is the role of a BeanFactory in Spring?

**a) It is responsible for instantiating and managing beans.**

b) It is responsible for resolving bean dependencies.

c) It is responsible for applying AOP to beans.

d) It is responsible for handling transactions in Spring.

8. What is the purpose of the @Value annotation in Spring?

**a) To inject values into properties or constructor arguments**

b) To inject dependencies into a bean

c) To define a bean's scope

d) To enable autowiring by name

9. Which of the following is not a valid stereotype annotation in Spring?

a) @Component

b) @Controller

c) @Service

**d) @PersistenceContext**

10. In Spring MVC, what is the role of the DispatcherServlet?

a) It is responsible for mapping requests to appropriate handler methods.

b) It is responsible for managing the lifecycle of beans.

c) It is responsible for managing database transactions.

**d) It is responsible for handling HTTP requests and responses.**

11. What does the term "Aspect" refer to in Spring AOP?

a) A class that provides additional functionalities to a bean

**b) A module that encapsulates cross-cutting concerns**

c) A configuration file for defining pointcuts

d) A bean that acts as a proxy for other beans

12. Which of the following is true about Spring Boot?

a) It is a framework for building only web applications.

b) It requires XML configuration for setup.

**c) It simplifies the process of building and deploying Spring applications.**

d) It is not suitable for microservices architecture.

13. What is the purpose of the @RequestMapping annotation in Spring MVC?

**a) To define the mapping between URL paths and controller methods**

b) To define the mapping between beans and their dependencies

c) To specify the scope of a bean

d) To enable data binding in forms

14. Which of the following is a valid stereotype annotation used in Spring MVC for defining a controller?

**a) @Controller**

b) @Repository

c) @Service

d) @Component

15. Which of the following is not a feature provided by Spring Security?

a) Authentication

b) Authorization

**c) Dependency Injection**

d) Session Management

16. What is the purpose of the @Transactional annotation in Spring?

**a) To mark a method as transactional and participate in a transaction**

**b) To define the transaction isolation level**

c) To enable lazy loading of entities

**d) To define the rollback rules for a transaction**

17. Which Spring module is used for integrating Spring with other Java EE frameworks like Hibernate, JPA, and JMS?

a) Spring Web MVC

**b) Spring Data**

c) Spring Batch

d) Spring ORM

18. What is the role of the BeanPostProcessor interface in Spring?

**a) It is used for post-processing beans before initialization.**

**b) It is used for post-processing beans after initialization.**

c) It is used for defining custom bean scopes.

d) It is used for defining custom annotations.

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

**a) To specify which bean should be autowired if multiple beans of the same type are present**

b) To define a custom bean scope

c) To enable transaction management

d) To enable AOP proxies

20. Which of the following is not a valid bean scope in Spring?

a) Request

b) Session

c) Application

**d) Transient**

21. What is the purpose of the @RestController annotation in Spring MVC?

a) To define a bean

b) To define a repository

**c) To define a controller that returns data rather than views**

d) To define a service component

22. Which Spring feature is used for handling cross-cutting concerns like logging and security?

**a) Aspect-Oriented Programming (AOP)**

b) Inversion of Control (IoC)

c) Dependency Injection (DI)

d) Application Context

23. What does ORM stand for in the context of Spring?

**a) Object-Relational Mapping**

b) Object-Resource Mapping

c) Object-Reflection Mapping

d) Object-Resource Model

24. Which of the following annotations is used to specify the primary bean when multiple beans of the same type are present?

**a) @Primary**

b) @Main

c) @Preferred

d) @Default

25. What is the purpose of the @Configuration annotation in Spring?

a) To specify the configuration for a Spring bean

b) To define the configuration for Spring MVC

c) To enable Spring Boot auto-configuration

**d) To indicate that a class declares one or more @Bean methods**

|  |
| --- |
| Correct answers for the Spring fundamentals multiple choice questions:  1. c) A lightweight Spring-based application framework  2. b) Dependency injection  3. b) It enables auto-configuration and component scanning  4. d) All of the above  5. b) Web MVC support  6. a) spring-boot-starter-data  7. c) It configures the application's properties  8. b) spring-boot-starter-jdbc  9. b) @RestController  10. a) To enable monitoring and managing of Spring Boot applications  11. a) @EnableCaching  12. b) Logback  13. a) spring-boot-starter-test  14. b) Using XML configuration  15. c) To improve the development experience by providing automatic application restarts and LiveReload  16. a) spring-boot-starter-data-mongodb  17. b) To package the Spring Boot application as an executable JAR or WAR file  18. c) @Value  19. b) To enable auto-configuration based on the dependencies in the classpath  20. d) SQLite  21. d) To bootstrap the entire application context for integration testing  22. b) It enables a property-based condition for selecting auto-configuration classes  23. c) It only works with embedded servers.  24. c) It specifies the version of Spring Boot to be used.  25. a) @SpringBootApplication |

**Spring Boot:** (40)

Questions covering Spring Boot, including its features, annotations, dependencies, testing, and configuration

1. What is Spring Boot?

a) A web server

b) A microservices framework

**c) A lightweight Spring-based application framework**

d) A tool for managing database connections

2. Which of the following is NOT a feature of Spring Boot?

a) Auto-configuration

**b) Dependency injection**

c) Embedded server support

d) Actuator endpoints

3. What is the purpose of the `@SpringBootApplication` annotation in Spring Boot?

a) It specifies the main class of the Spring Boot application

**b) It enables auto-configuration and component scanning**

c) It configures the application context

d) It sets up database connections

4. Which embedded server does Spring Boot support out of the box?

a) Apache Tomcat

b) Jetty

c) Undertow

**d) All of the above**

5. What does the `spring-boot-starter-web` dependency provide in Spring Boot?

a) JDBC support

**b) Web MVC support**

c) JPA support

d) Security support

6. Which of the following is NOT a Spring Boot starter dependency?

**a) spring-boot-starter-data**

b) spring-boot-starter-actuator

c) spring-boot-starter-logging

**d) spring-boot-starter-servlet**

7. What is the purpose of the `application.properties` (or `application.yml`) file in Spring Boot?

a) It defines the application's endpoints

b) It specifies the application's dependencies

**c) It configures the application's properties**

d) It contains the application's source code

8. Which Spring Boot starter dependency provides support for working with relational databases?

a) spring-boot-starter-data

**b) spring-boot-starter-jdbc**

c) spring-boot-starter-web

d) spring-boot-starter-actuator

9. Which annotation is used to define a REST controller in a Spring Boot application?

a) @Controller

**b) @RestController**

c) @Service

d) @Component

10. What is the purpose of Spring Boot's Actuator module?

**a) To enable monitoring and managing of Spring Boot applications**

b) To provide support for asynchronous processing

c) To handle security configurations

d) To facilitate data caching

11. Which of the following annotations is used to enable caching in a Spring Boot application?

**a) @EnableCaching**

b) @EnableCache

c) @Cacheable

d) @CacheConfig

12. What is Spring Boot's default logging framework?

a) Log4j

**b) Logback**

c) SLF4J

d) Java Util Logging

13. Which Spring Boot starter dependency provides support for testing?

**a) spring-boot-starter-test**

b) spring-boot-starter-junit

c) spring-boot-starter-mockito

d) spring-boot-starter-mockmvc

14. Which of the following is NOT a method to configure a Spring Boot application?

a) Using annotations

**b) Using XML configuration**

c) Using Java configuration

d) Using properties files

15. What is the purpose of Spring Boot DevTools?

a) To facilitate database migrations

b) To automate the deployment process

**c) To improve the development experience by providing automatic application restarts and LiveReload**

d) To provide security features for Spring Boot applications

16. Which Spring Boot starter dependency provides support for working with NoSQL databases like MongoDB?

**a) spring-boot-starter-data-mongodb**

b) spring-boot-starter-data-redis

c) spring-boot-starter-data-cassandra

d) spring-boot-starter-data-neo4j

17. What is the purpose of the `spring-boot-maven-plugin` in a Maven-based Spring Boot project?

a) To generate documentation for the Spring Boot application

**b) To package the Spring Boot application as an executable JAR or WAR file**

c) To manage the application's dependencies

d) To run tests during the Maven build process

18. Which of the following annotations is used to inject properties from `application.properties` into Spring beans?

a) @InjectProperties

b) @Autowired

**c) @Value**

d) @PropertySource

19. What is the purpose of Spring Boot's `@EnableAutoConfiguration` annotation?

a) To disable auto-configuration in the application

**b) To enable auto-configuration based on the dependencies in the classpath**

c) To enable caching in the application

d) To enable security configurations

20. Which of the following databases is NOT supported out of the box by Spring Boot's data starter dependencies?

a) MySQL

b) PostgreSQL

c) Oracle

**d) SQLite**

21. What is the purpose of the `@SpringBootTest` annotation in Spring Boot testing?

a) To specify which beans should be loaded during integration tests

b) To specify which profiles should be active during testing

c) To specify the primary test class

**d) To bootstrap the entire application context for integration testing**

22. What does Spring Boot's `@ConditionalOnProperty` annotation do?

**a) It defines a condition for creating a bean based on the value of a specific property**

b) It enables a property-based condition for selecting auto-configuration classes

c) It checks whether a property is present in the application's properties file

d) It disables property-based conditional checks for beans

23. Which of the following is NOT true about Spring Boot Actuator?

a) It provides health check endpoints for monitoring application health.

b) It allows you to remotely monitor and manage your Spring Boot application.

**c) It only works with embedded servers.**

d) It exposes endpoints for gathering metrics about your application.

24. What is the purpose of the `spring-boot-starter-parent` POM in a Spring Boot project?

a) It defines the parent project for the Spring Boot application.

**b) It provides common configurations and dependencies for Spring Boot projects.**

c) It specifies the version of Spring Boot to be used.

d) It manages the deployment process of the Spring Boot application.

25. Which annotation is used to define the main method of a Spring Boot application?

**a) @SpringBootApplication**

b) @MainMethod

c) @StartApplication

d) @MainClass

26. Which annotation is used to specify the configuration properties prefix in a Spring Boot application?

a) @PropertyPrefix

**b) @ConfigurationProperties**

c) @PropertySource

d) @PropertiesPrefix

27. What is the purpose of the `spring-boot-starter-security` dependency in Spring Boot?

**a) To enable security configurations for the application**

b) To provide support for caching

c) To enable messaging features

d) To facilitate database operations

28. Which of the following is NOT a valid profile in Spring Boot?

a) Production

b) Development

c) Staging

d) Testing

**e) None of the Above**

29. What does the `@SpringBootApplication(exclude = {...})` annotation allow you to do?

a) Exclude specific auto-configuration classes from being applied

b) Exclude specific beans from being created in the application context

c) Exclude specific profiles from being activated

d) Exclude specific packages from component scanning

30. Which annotation is used to enable asynchronous processing in a Spring Boot application?

**a) @EnableAsync**

b) @AsyncProcessing

c) @EnableProcessing

d) @AsyncEnabled

31. What is the purpose of the `spring-boot-actuator` module in Spring Boot?

a) To provide support for security features

b) To enable internationalization and localization

**c) To expose management endpoints for monitoring and managing the application**

d) To handle web requests and responses

32. Which of the following annotations is used to define custom health indicators in Spring Boot Actuator?

a) @CustomHealthIndicator

**b) @HealthIndicator**

c) @Indicator

d) @HealthCheck

33. What is the default server port for a Spring Boot application?

**a) 8080**

b) 8888

c) 9090

d) 8000

34. What does the `@ConditionalOnClass` annotation do in Spring Boot?

**a) It defines a condition for creating a bean based on the presence of a specific class in the classpath**

b) It enables class-based conditional checks for beans

c) It disables the classpath scanning for specific classes

d) It checks whether a class is present in the application's properties file

35. Which of the following annotations is used to enable scheduled tasks in a Spring Boot application?

**a) @EnableScheduling**

b) @ScheduledTask

c) @EnableTaskScheduling

**d) @Scheduled**

36. What is the purpose of the `@RestControllerAdvice` annotation in Spring Boot?

a) To define REST controllers

**b) To provide global exception handling for REST controllers**

c) To specify the advice to be applied to REST controllers

d) To enable aspect-oriented programming for REST controllers

37. Which of the following annotations is used to disable specific auto-configuration classes in Spring Boot?

a) @DisableAutoConfiguration

b) @ExcludeAutoConfiguration

c) @IgnoreAutoConfiguration

d) @SkipAutoConfiguration

**e) None of the Above**

38. What is the purpose of Spring Boot's `@EnableConfigurationProperties` annotation?

**a) To enable specific configuration properties for the application**

b) To disable auto-configuration properties

c) To enable manual configuration of properties in XML files

d) To enable automatic configuration of properties based on profiles

39. Which of the following is NOT a valid way to override default Spring Boot configuration properties?

a) Using environment variables

b) Using YAML files

**c) Using XML configuration**

d) Using command-line arguments

40. What is the purpose of the `spring-boot-actuator-autoconfigure` module in Spring Boot?

**a) To provide auto-configuration for Spring Boot Actuator**

b) To enable automatic configuration of database connections

c) To facilitate auto-configuration of security features

d) To manage the deployment process of Spring Boot applications

|  |
| --- |
| Correct answers for the Spring Boot multiple choice questions:  1. c) A lightweight Spring-based application framework  2. b) A microservices framework  3. b) It enables auto-configuration and component scanning  4. d) All of the above  5. b) Web MVC support  6. a) spring-boot-starter-data  7. c) It configures the application's properties  8. b) spring-boot-starter-jdbc  9. b) @RestController  10. a) To enable monitoring and managing of Spring Boot applications  11. a) @EnableCaching  12. b) Logback  13. a) spring-boot-starter-test  14. c) Using Java configuration  15. c) To improve the development experience by providing automatic application restarts and LiveReload  16. a) spring-boot-starter-data-mongodb  17. b) To package the Spring Boot application as an executable JAR or WAR file  18. c) @Value  19. b) To enable auto-configuration based on the dependencies in the classpath  20. d) SQLite  21. a) To specify which beans should be loaded during integration tests  22. b) @EnableAutoConfiguration enables a property-based condition for selecting auto-configuration classes  23. c) It only works with embedded servers.  24. b) It provides common configurations and dependencies for Spring Boot projects.  25. a) @SpringBootApplication  26. a) @PropertyPrefix  27. a) To enable security configurations for the application  28. d) Testing  29. a) Exclude specific auto-configuration classes from being applied  30. a) @EnableAsync  31. ?  32. a) @HealthIndicator  33. a) 8080  34. a) It defines a condition for creating a bean based on the presence of a specific class in the classpath  35. a) @EnableScheduling  36. b) To provide global exception handling for REST controller  37. b) @ExcludeAutoConfiguration  38. a) To enable specific configuration properties for the applicatio  39. c) Using XML configuration  40. c) To provide support for distributed tracing |

**Spring Security:** (25)

Questions covering Spring Security, including authentication, authorization, configuration, annotations, and modules.

1. What is Spring Security?

a) A framework for building microservices

**b) A framework for securing Java applications**

c) A lightweight web server

d) A dependency injection framework

2. Which of the following is NOT a feature of Spring Security?

a) Authentication

b) Authorization

**c) Dependency Injection**

d) Session Management

3. Which module is responsible for providing authentication and authorization support in Spring Security?

**a) spring-security-core**

b) spring-security-config

c) spring-security-web

d) spring-security-auth

4. What is the purpose of the `AuthenticationManager` interface in Spring Security?

a) To manage database connections

**b) To manage user authentication**

c) To manage session data

d) To manage authorization rules

5. Which of the following is a common authentication provider in Spring Security?

a) JDBC Authentication

b) OAuth Authentication

c) LDAP Authentication

**d) All of the above**

6. Which annotation is used to secure methods in Spring Security?

a) @Secure

b) @SecureMethod

**c) @Secured**

d) @Authorize

7. What is the purpose of the `UserDetailsService` interface in Spring Security?

a) To manage user authentication

b) To manage user authorization

**c) To load user-specific data during authentication**

d) To manage user sessions

8. Which of the following is NOT a supported authentication mechanism in Spring Security?

a) Form-based authentication

b) Basic authentication

c) OAuth authentication

d) JWT authentication

**e) None of the Above**

9. What does CSRF stand for in the context of Spring Security?

**a) Cross-Site Request Forgery**

b) Cross-Site Request Framework

c) Cross-Site Resource Forgery

d) Cross-Site Resource Framework

10. What is the purpose of the `@PreAuthorize` annotation in Spring Security?

**a) To specify access control rules before method execution**

b) To specify access control rules after method execution

c) To define security constraints in XML configuration

d) To handle authentication failures

11. Which of the following is NOT a valid HTTP security configuration in Spring Security?

a) PermitAll

b) DenyAll

c) Authenticated

**d) RoleAllowed**

12. What is the purpose of the `PasswordEncoder` interface in Spring Security?

**a) To encode user passwords for storage**

b) To decode user passwords during authentication

c) To manage user sessions

d) To manage authorization rules

13. Which Spring Security module is responsible for protecting web applications?

**a) spring-security-web**

b) spring-security-core

c) spring-security-config

d) spring-security-authentication

14. What is the purpose of the `RememberMeAuthenticationFilter` in Spring Security?

a) To enable form-based authentication

**b) To enable remember-me functionality for users**

c) To manage authentication tokens

d) To handle authorization failures

15. Which of the following is a valid HTTP security configuration in Spring Security to require HTTPS?

a) requireHttps()

b) requiresHttps()

c) requiresSSL()

d) requireSSL()

**e) requireSecure()**

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

**a) To enable web security features for a Spring application**

b) To enable OAuth authentication

c) To enable session management

d) To enable method-level security

17. Which of the following is NOT a method to configure Spring Security in a Spring Boot application?

**a) Using XML configuration**

b) Using Java configuration

c) Using properties files

d) Using annotations

18. What is the purpose of the `SecurityContextHolder` class in Spring Security?

a) To manage user authentication

b) To manage user authorization

**c) To store security context information**

d) To handle session management

19. Which Spring Security module is responsible for handling authentication mechanisms like OAuth and OpenID?

**a) spring-security-oauth**

b) spring-security-authentication

c) spring-security-core

d) spring-security-web

20. What is the purpose of the `@CrossOrigin` annotation in Spring Security?

**a) To enable cross-origin requests for specific controllers**

b) To disable cross-origin requests for specific controllers

c) To enable cross-site scripting protection

d) To handle cross-site request forgery attacks

21. Which of the following is NOT a method of authentication supported by Spring Security?

a) LDAP Authentication

**b) OAuth Authentication**

c) Token Authentication

d) JDBC Authentication

22. What is the purpose of the `SecurityConfigurerAdapter` class in Spring Security?

**a) To configure global authentication settings**

b) To configure HTTP security settings

c) To configure OAuth authentication

d) To configure session management settings

23. Which Spring Security module is responsible for providing support for OAuth 2.0 authentication?

**a) spring-security-oauth**

b) spring-security-core

c) spring-security-web

d) spring-security-config

24. What is the purpose of the `AccessDecisionManager` interface in Spring Security?

a) To manage user authentication

**b) To manage user authorization**

c) To manage user sessions

d) To manage session data

25. Which of the following annotations is used to define security constraints at the method level in Spring Security?

**a) @Secured**

b) @PreAuthorize

c) @PostAuthorize

d) @RolesAllowed

|  |
| --- |
| Correct answers corresponding to the Spring Security questions.  1. b) A framework for securing Java applications  2. c) Dependency Injection  3. a) spring-security-core  4. b) To manage user authentication  5. d) All of the above  6. c) @Secured  7. c) To load user-specific data during authentication  8. d) JWT authentication  9. a) Cross-Site Request Forgery  10. a) @PreAuthorize  11. c) Authenticated  12. a) To encode user passwords for storage  13. a) spring-security-web  14. b) To enable remember-me functionality for users  15. a) requireHttps()  16. a) @EnableWebSecurity  17. a) Using XML configuration  18. c) To store security context information  19. a) spring-security-oauth  20. b) To prevent services from communicating with each other  21. b) OAuth Authentication  22. a) To configure global authentication settings  23. d) To manage user authorization  24. b) To manage user authorization  25. a) @Secured |

**Spring Micro Services:**  (40)

Questions covering microservices architecture, including principles, patterns, challenges, and technologies commonly used in implementing microservices-based systems with Spring.

1. What is a microservice architecture?

a) A monolithic architecture

**b) A distributed architecture where applications are composed of small, independent services**

c) A centralized architecture

d) An architecture where services communicate directly with each other

2. Which of the following is a characteristic of microservices?

a) Tight coupling between services

b) Large, monolithic codebases

**c) Independent deployment and scalability of services**

d) Centralized data storage

3. What is the primary goal of using a microservices architecture?

a) To increase the complexity of the system

b) To decrease the scalability of the system

**c) To improve maintainability and scalability**

d) To introduce tight coupling between services

4. In a microservices architecture, what does each service typically represent?

**a) A single component of the application**

b) The entire application

c) A collection of components

d) A database schema

5. Which of the following is NOT a potential benefit of using microservices?

a) Improved fault isolation

b) Simplified deployment process

c) Reduced operational overhead

**d) Increased coupling between services**

6. What is Spring Cloud?

a) A framework for building monolithic applications

**b) A framework for building microservices-based applications**

c) A relational database management system

d) A messaging system

7. Which Spring project provides support for building microservices with Spring Boot?

a) Spring MVC

b) Spring Data

**c) Spring Cloud**

d) Spring Security

8. What is the purpose of service discovery in microservices architecture?

a) To hide services from each other

**b) To locate services dynamically at runtime**

c) To increase coupling between services

d) To reduce fault tolerance

9. Which of the following is a commonly used service discovery tool with Spring Cloud?

**a) Eureka**

b) RabbitMQ

c) Kafka

d) ActiveMQ

10. What is the role of a service registry in microservices architecture?

a) To store configuration properties for services

b) To manage the lifecycle of services

**c) To provide a centralized directory of available services**

d) To enforce security policies between services

11. Which Spring Cloud module provides support for centralized configuration management?

**a) Spring Cloud Config**

b) Spring Cloud Netflix

c) Spring Cloud Gateway

d) Spring Cloud Stream

12. What is the purpose of circuit breakers in microservices architecture?

a) To establish direct communication between services

b) To prevent services from communicating with each other

**c) To handle failures gracefully and prevent cascading failures**

d) To increase coupling between services

13. Which of the following is NOT a feature of Spring Cloud Netflix?

a) Service discovery

b) Circuit breakers

c) Distributed tracing

**d) Centralized logging**

14. What is the purpose of Spring Cloud Sleuth?

**a) To provide support for distributed tracing**

b) To manage service discovery

c) To implement circuit breakers

d) To handle centralized configuration

15. Which Spring Cloud module provides support for building API gateways?

**a) Spring Cloud Gateway**

b) Spring Cloud Netflix

c) Spring Cloud Config

d) Spring Cloud Sleuth

16. What is the purpose of distributed tracing in microservices architecture?

a) To debug issues in a single service

b) To monitor the performance of individual services

**c) To trace requests as they propagate through multiple services**

d) To enforce security policies between services

17. Which of the following is NOT a challenge associated with microservices architecture?

a) Increased complexity in deployment and monitoring

**b) Tight coupling between services**

c) Service discovery and load balancing

d) Data consistency and transaction management

18. What is an API gateway in the context of microservices architecture?

a) A centralized database for storing configuration properties

**b) A service that handles incoming requests and routes them to appropriate microservices**

c) A communication protocol used between microservices

d) A tool for managing service discovery

19. Which Spring Cloud module provides support for messaging-based microservices?

**a) Spring Cloud Stream**

b) Spring Cloud Gateway

c) Spring Cloud Config

d) Spring Cloud Sleuth

20. What is the role of a message broker in microservices architecture?

a) To manage service discovery

b) To handle incoming HTTP requests

**c) To facilitate communication between services using asynchronous messaging**

d) To store configuration properties for services

21. What is an event-driven architecture in microservices?

a) An architecture where services communicate using synchronous HTTP calls

**b) An architecture where services communicate using asynchronous messaging**

c) An architecture where services do not communicate with each other

d) An architecture where services share a common database

22. Which of the following is a common protocol used for asynchronous messaging in microservices?

a) HTTP

b) SOAP

**c) AMQP**

d) REST

23. What is the purpose of bulkheads in microservices architecture?

a) To increase coupling between services

**b) To isolate failures and prevent them from affecting other services**

c) To provide a centralized directory of available services

d) To handle centralized configuration

24. Which of the following is NOT a common strategy for managing data consistency in microservices architecture?

a) Synchronous communication between services

b) Event sourcing

c) Distributed transactions

**d) CQRS (Command Query Responsibility Segregation)**

25. What is the role of Spring Cloud Stream in microservices architecture?

a) To provide support for service discovery

b) To implement circuit breakers

**c) To facilitate communication between microservices using messaging**

d) To manage centralized configuration

26. What is the purpose of resilience patterns in microservices architecture?

a) To increase the complexity of the system

**b) To improve fault tolerance and resilience in the face of failures**

c) To decrease the scalability of the system

d) To provide centralized logging

27. Which of the following is a common resilience pattern used in microservices architecture?

**a) Retry**

b) Tight coupling

c) Monolithic architecture

d) Direct communication between services

28. What is the purpose of distributed tracing in microservices architecture?

a) To debug issues in a single service

b) To monitor the performance of individual services

**c) To trace requests as they propagate through multiple services**

d) To enforce security policies between services

29. Which of the following is a common tool for monitoring and managing microservices in production?

a) Kubernetes

b) Docker

c) Apache Kafka

**d) ELK Stack (Elasticsearch, Logstash, Kibana)**

30. What is the purpose of polyglot persistence in microservices architecture?

a) To increase the complexity of the system

b) To decrease coupling between services

**c) To use multiple databases optimized for different types of data**

d) To handle centralized configuration

31. Which of the following is NOT a common challenge associated with migrating to microservices architecture?

a) Service discovery

b) Monolithic codebase

c) Tight coupling between services

d) Data consistency and transaction management

32. What is the purpose of centralized logging in microservices architecture?

a) To store configuration properties for services

b) To facilitate communication between services

**c) To aggregate logs from multiple services for easier monitoring and debugging**

d) To manage service discovery

33. Which of the following is a common tool for centralized logging in microservices architecture?

a) RabbitMQ

b) Apache Kafka

**c) Logstash**

d) Eureka

34. What is the purpose of a container orchestration tool like Kubernetes in microservices architecture?

**a) To manage the lifecycle of containers running microservices**

b) To handle service discovery

c) To implement circuit breakers

d) To provide centralized configuration

35. What is the role of API documentation in microservices architecture?

a) To increase the complexity of the system

b) To provide a centralized directory of available services

c) To facilitate communication between services

**d) To document the endpoints and payloads of each microservice**

36. Which of the following is NOT a common challenge associated with microservices architecture?

a) Tight coupling between services

b) Increased complexity in deployment and monitoring

c) Centralized data storage

d) Service discovery and load balancing

37. What is the purpose of using containerization technologies like Docker in microservices architecture?

a) To increase coupling between services

**b) To isolate services and their dependencies**

c) To handle centralized configuration

d) To manage service discovery

38. Which of the following is a common approach for implementing authentication and authorization in microservices architecture?

**a) Centralized authentication and authorization server**

b) Direct communication between services

c) Shared database for user management

d) Monolithic authentication service

39. What is the purpose of API gateways in microservices architecture?

a) To increase the complexity of the system

**b) To provide a single entry point for clients to access multiple microservices**

c) To isolate failures and prevent them from affecting other services

d) To provide centralized logging

40. Which of the following is a common strategy for ensuring fault tolerance in microservices architecture?

**a) Retry**

b) Tight coupling

c) Monolithic architecture

d) Shared database for user management

|  |
| --- |
| Correct answers corresponding to the Spring Microservices questions.  1. b) A distributed architecture where applications are composed of small, independent services  2. c) Independent deployment and scalability of services  3. b) To improve maintainability and scalability  4. a) A single component of the application  5. d) Increased coupling between services  6. b) A framework for building microservices-based applications  7. c) Spring Cloud  8. b) To locate services dynamically at runtime  9. a) Eureka  10. a) Spring Cloud Config  11. a) To establish direct communication between services  12. c) To trace requests as they propagate through multiple services  13. d) Centralized logging  14. a) To provide support for distributed tracing  15. a) Spring Cloud Gateway  16. c) To trace requests as they propagate through multiple services  17. c) Data consistency and transaction management  18. b) A service that handles incoming requests and routes them to appropriate microservices  19. a) Spring Cloud Stream  20. c) To facilitate communication between services using asynchronous messaging  21. b) To prevent services from communicating with each other  22. c) AMQP  23. b) To isolate failures and prevent them from affecting other services  24. d) CQRS (Command Query Responsibility Segregation)  25. a) To provide support for service discovery  26. b) To improve fault tolerance and resilience in the face of failures  27. a) Retry  28. c) To trace requests as they propagate through multiple services  29. d) ELK Stack (Elasticsearch, Logstash, Kibana)  30. c) To use multiple databases optimized for different types of data  31. c) Centralized data storage  32. c) To aggregate logs from multiple services for easier monitoring and debugging  33. c) Logstash  34. a) To manage the lifecycle of containers running microservices  35. d) To document the endpoints and payloads of each microservice  36. c) Centralized data storage  37. b) To isolate services and their dependencies  38. a) Centralized authentication and authorization server  39. b) To provide a single entry point for clients to access multiple microservices  40. a) Retry |