

Java/Spring Boot Interview Questions

1. What is Spring Framework?
2. What problems does Spring solve?
3. What is Dependency Injection (DI)?
4. What are Spring Beans?
5. Difference between `@Component`, `@Service`, `@Repository`?
6. What is `@Autowired`?
7. What is Spring Boot? How is it different from Spring?
8. What is `application.properties` / `application.yml` used for?
9. What is an embedded server in Spring Boot?
10. How do you create a REST API in Spring Boot?
11. What is `@RestController` and how is it different from `@Controller`?
12. What are Spring Boot starters?
13. What is Spring Boot auto-configuration?
14. What are Spring Boot profiles?
15. How do you configure different environments in Spring Boot?
16. What is Spring ApplicationContext?
17. Difference between BeanFactory and ApplicationContext?
18. What is component scanning in Spring?
19. What is the purpose of `@Configuration` annotation?
20. How to define beans using Java config?
21. How to define beans using XML config?
22. What is `@PostConstruct` and `@PreDestroy`?
23. What is the purpose of `@Scope` annotation?
24. Difference between singleton and prototype bean scope?
25. What is `@Value` annotation?
26. What are Spring events?
27. What is `ApplicationEventPublisher`?
28. Difference between dependency injection via constructor vs setter?
29. How to handle exceptions in Spring MVC?
30. What is `@ControllerAdvice`?
31. What is `@ExceptionHandler`?
32. What is `@RestControllerAdvice`?
33. How do you validate request parameters in Spring Boot?
34. What is `@Valid` and `@Validated`?
35. What is Bean Validation (JSR 303)?
36. How do you customize validation messages in Spring Boot?
37. What is `@PathVariable`?
38. What is `@RequestParam`?
39. Difference between `@RequestBody` and `@ModelAttribute`?
40. What is `@RequestMapping` and its HTTP method shortcuts?
41. What is `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`?
42. What is `ResponseEntity`?
43. How do you handle CORS in Spring Boot?
44. How to enable Swagger / OpenAPI documentation?
45. What is Spring Boot actuator?
46. How to monitor application metrics with Actuator?
47. What is `/health` endpoint in Spring Boot?

Java/Spring Boot Interview Questions

48. How to customize Actuator endpoints?
49. What is `spring-boot-devtools` used for?
50. How to enable hot reload in Spring Boot?

51. What is Spring Data JPA?
52. How to define entities in JPA?
53. Difference between `@Entity` and `@Table`?
54. What is `@Id` and `@GeneratedValue`?
55. Difference between `.GenerationType.IDENTITY` and `.GenerationType.AUTO`?
56. What is `@Column` used for?
57. What is `@ManyToOne` vs `@OneToOne`?
58. What is `@OneToOne` vs `@ManyToMany`?
59. What is lazy vs eager fetching?
60. What is `EntityManager`?
61. What is `CrudRepository`?
62. What is `JpaRepository`?
63. What is `PagingAndSortingRepository`?
64. How to write custom queries with `@Query`?
65. What is query derivation in Spring Data JPA?
66. What is transaction management in Spring?
67. Difference between programmatic and declarative transactions?
68. What is `@Transactional` annotation?
69. How to handle propagation in transactions?
70. How to handle isolation levels in transactions?
71. How to rollback transactions manually?
72. How to handle exceptions in transactions?
73. What is Spring Boot Security?
74. How to secure REST APIs with Spring Security?
75. Difference between authentication and authorization?
76. What is OAuth2?
77. How to implement JWT authentication in Spring Boot?
78. How to configure in-memory user authentication?
79. How to use `UserDetailsService`?
80. What is `BCryptPasswordEncoder`?
81. How to protect endpoints with roles?
82. How to configure HTTP Basic and Form login?
83. What is CSRF and how to handle it?
84. How to implement method-level security (`@PreAuthorize`)?
85. What is Spring Boot logging?
86. How to configure logging with Logback?
87. How to create custom log patterns?
88. What is Spring Boot Actuator metrics?
89. How to monitor database connections in Actuator?
90. What is Spring Boot CLI?
91. How to create Spring Boot starter project?
92. How to configure YAML vs properties?
93. What are Spring Boot dependencies management?
94. What is embedded Tomcat, Jetty, or Undertow?
95. How to customize server port and context path?
96. What is Spring Boot profiles for dev, test, prod?

Java/Spring Boot Interview Questions

97. How to configure environment-specific beans?
98. How to create a REST API with pagination and sorting?
99. How to implement exception handling globally?
100. How to deploy Spring Boot as a standalone JAR or WAR?

Java/Spring Boot Interview Questions

1. What is multithreading? Lifecycle of a thread?
2. Difference between Runnable and Thread class?
3. What are thread priorities?
4. What is thread synchronization?
5. Difference between `wait()`, `notify()`, and `notifyAll()`?
6. What is ExecutorService in Java?
7. What is Callable vs Runnable?
8. Difference between ConcurrentHashMap and HashMap?
9. What is thread-safe collection?
10. Difference between ReentrantLock and synchronized?
11. What is volatile keyword?
12. What are atomic variables?
13. Difference between synchronized block and method?
14. What is deadlock? How to prevent it?
15. What is livelock?
16. What is starvation in threads?
17. Difference between yield() and sleep()?
18. Difference between join() and wait()?
19. What is daemon thread?
20. Difference between user thread and daemon thread?
21. What is thread pool?
22. Difference between fixed and cached thread pool?
23. What is Future and CompletableFuture?
24. Difference between synchronous and asynchronous execution?
25. What is fork/join framework?
26. Difference between parallel and sequential streams?
27. How to use Java 8 Stream API?
28. What are functional interfaces?
29. Difference between lambda expressions and anonymous classes?
30. What are default and static methods in interface?
31. What is method reference?
32. What is Optional class and how to use it?
33. Difference between Map, Filter, and Reduce in streams?
34. What are Java Generics?
35. Difference between List extends Object?
36. What are wildcards in Java Generics?
37. Difference between checked and unchecked exceptions?
38. How to create a custom exception?
39. Difference between final, finally, and finalize?
40. Difference between shallow copy and deep copy?
41. How to serialize and deserialize objects?
42. What is transient keyword?
43. Difference between Comparable and Comparator?
44. How to sort a list using Comparator?
45. What is Java Reflection API?
46. Difference between Class.forName() and object.getClass()?
47. What are Java annotations?
48. What is Java Memory Model?
49. Difference between stack, heap, and metaspace?
50. Difference between String, StringBuilder, and StringBuffer?

Java/Spring Boot Interview Questions

Intermediate – Spring Boot

51. What is Spring Boot auto-configuration?
52. Difference between `@Controller` and `@RestController`?
53. What are Spring Boot starters?
54. What is `@SpringBootApplication` annotation?
55. How to create a REST API in Spring Boot?
56. What is `ResponseEntity`?
57. What is `@RequestMapping` and its HTTP method shortcuts?
58. What is `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`?
59. What is `@PathVariable`?
60. What is `@RequestParam`?
61. Difference between `@RequestBody` and `@ModelAttribute`?
62. How to implement exception handling in Spring Boot?
63. What is `@ControllerAdvice`?
64. What is `@ExceptionHandler`?
65. What is `@RestControllerAdvice`?
66. How to validate request parameters in Spring Boot?
67. What is `@Valid` and `@Validated`?
68. How to customize validation messages in Spring Boot?
69. How to configure CORS in Spring Boot?
70. What is Spring Boot actuator?
71. How to monitor application metrics with Actuator?
72. What is `/health` endpoint?
73. How to customize Actuator endpoints?
74. How to use Spring Boot devtools?
75. How to enable hot reload in Spring Boot?
76. Difference between YAML and properties configuration?
77. How to define environment-specific properties?
78. What is `@Profile` annotation?
79. How to use `@Value` annotation?
80. Difference between BeanFactory and ApplicationContext?
81. How to define beans using Java config?
82. How to define beans using XML config?
83. Difference between singleton and prototype bean scope?
84. What is `@PostConstruct` and `@PreDestroy`?
85. What is dependency injection via constructor vs setter?
86. How to load external configuration files?
87. What is Spring Boot CLI?
88. How to create Spring Boot starter project?
89. What is embedded Tomcat, Jetty, or Undertow?
90. How to customize server port and context path?
91. How to deploy Spring Boot as standalone JAR or WAR?
92. What is logging in Spring Boot?
93. How to configure logging with Logback?
94. How to create custom log patterns?
95. How to monitor database connections in Actuator?

Java/Spring Boot Interview Questions

96. How to implement global exception handling?
97. How to implement pagination and sorting?
98. How to secure REST APIs with Spring Security?
99. How to enable HTTP Basic or Form Login?
100. How to configure method-level security (`@PreAuthorize`)?

Microservices & Scenario-Based Questions

101. What is microservices architecture?
102. Difference between monolith and microservices?
103. What is service discovery?
104. What is Eureka Server and Eureka Client?
105. What is Ribbon?
106. What is Feign Client?
107. What is Hystrix circuit breaker?
108. How to handle fallback methods in microservices?
109. What is Spring Cloud Gateway?
110. How to route requests in API gateway?
111. What is load balancing in microservices?
112. Difference between client-side and server-side load balancing?
113. What is distributed tracing?
114. What is Spring Cloud Sleuth?
115. How to propagate logs across microservices?
116. What is Zipkin?
117. How to handle centralized configuration in microservices?
118. What is Spring Cloud Config Server?
119. How to implement dynamic configuration refresh?
120. What is circuit breaker pattern?
121. How to implement retry strategy in microservices?
122. What is bulkhead pattern?
123. What is idempotency in microservices?
124. How to implement rate-limiting?
125. How to implement caching in microservices?
126. What is distributed caching?
127. How to use Redis in microservices?
128. How to implement messaging with RabbitMQ?
129. How to implement messaging with Kafka?
130. What is event-driven architecture?
131. How to handle eventual consistency?
132. What is saga pattern?
133. What is two-phase commit (2PC)?
134. How to implement API versioning?
135. How to implement global exception handling?
136. How to implement reactive APIs with Spring WebFlux?
137. Difference between WebFlux and Spring MVC?
138. What are Mono and Flux?
139. How to handle backpressure in reactive streams?
140. What is RSocket in Spring Boot?

Java/Spring Boot Interview Questions

141. How to implement JWT authentication across microservices?
142. How to implement OAuth2 authorization code flow?
143. How to implement refresh tokens?
144. How to secure inter-service communication?
145. What is service registry?
146. How to implement health checks for microservices?
147. What is Spring Boot actuator health indicators?
148. How to monitor CPU/memory usage in microservices?
149. How to implement logging and correlation IDs?
150. How to implement tracing with Sleuth and Zipkin?
151. How to handle database failures in microservices?
152. How to handle partial failures in distributed systems?
153. How to implement retry with exponential backoff?
154. How to implement bulkhead for service isolation?
155. How to implement failover strategies?
156. How to design microservices for high throughput?
157. How to design APIs for scalability?
158. How to handle large payloads?
159. How to implement async processing in Spring Boot?
160. How to implement scheduled tasks?
161. How to implement batch processing with Spring Batch?
162. How to handle transactions in distributed systems?
163. How to implement saga pattern for distributed transactions?
164. How to implement CQRS pattern?
165. How to implement event sourcing?
166. How to implement API caching?
167. How to implement rate-limiting?
168. How to monitor service SLAs?
169. How to implement metrics collection?
170. How to use Prometheus and Grafana with Spring Boot?
171. How to implement circuit breaker pattern?
172. How to fallback to default responses?
173. How to implement retries with Hystrix or Resilience4j?
174. How to implement dynamic configuration with Spring Cloud Config?
175. How to implement feature toggles?
176. How to implement blue-green deployment for microservices?
177. How to implement canary deployment?
178. How to handle versioning in microservices?
179. How to implement API documentation with Swagger?
180. How to implement API documentation with OpenAPI 3?
181. How to implement HATEOAS in REST APIs?
182. How to implement validation in microservices?
183. How to implement global exception handling?
184. How to implement centralized logging?
185. How to implement ELK stack (Elasticsearch, Logstash, Kibana)?
186. How to implement monitoring and alerting?
187. How to implement security for sensitive endpoints?
188. How to implement SSL/TLS in microservices?
189. How to implement OAuth2 / OpenID Connect?
190. How to implement JWT token propagation?

Java/Spring Boot Interview Questions

191. How to implement correlation IDs in logs?
192. How to implement asynchronous messaging?
193. How to implement Kafka consumer groups?
194. How to implement message partitioning?
195. How to handle message duplication?
196. How to implement idempotent consumers?
197. How to implement transactional messaging?
198. How to implement database sharding?
199. How to implement multi-datasource configuration?
200. How to implement fallback databases in case of failures?
201. How to implement dynamic thread pools?
202. How to implement connection pooling?
203. How to optimize database queries?
204. How to implement caching for queries?
205. How to implement distributed locking?
206. How to implement optimistic locking?
207. How to implement pessimistic locking?
208. How to implement retry with Spring Retry?
209. How to implement circuit breaker with Resilience4j?
210. How to implement request throttling?
211. How to implement API gateway rate limiting?
212. How to implement health monitoring for microservices?
213. How to implement service self-healing?
214. How to implement logging best practices?
215. How to implement exception handling best practices?
216. How to implement performance monitoring?
217. How to implement alerting for failures?
218. How to implement distributed transactions with Saga?
219. How to implement rollback strategies for distributed systems?
220. How to handle network failures in microservices?
221. How to design APIs for high concurrency?
222. How to handle concurrent updates in microservices?
223. How to implement feature flags in Spring Boot?
224. How to implement blue-green deployment with Kubernetes?
225. How to implement canary deployment with Kubernetes?
226. How to implement graceful shutdown of services?
227. How to implement service retries with exponential backoff?
228. How to implement async REST API calls?
229. How to implement WebSocket communication in Spring Boot?
230. How to implement SSE (Server-Sent Events)?
231. How to implement event-driven architecture?
232. How to implement Kafka producers and consumers?
233. How to implement RabbitMQ producers and consumers?
234. How to implement idempotent message handling?
235. How to implement message deduplication?
236. How to implement service versioning in microservices?
237. How to implement API versioning strategies?
238. How to implement A/B testing in microservices?
239. How to implement logging correlation across services?
240. How to implement distributed tracing?

Java/Spring Boot Interview Questions

- 241. How to implement performance benchmarking?
- 242. How to implement resilience patterns?
- 243. How to implement fail-fast patterns?
- 244. How to implement retry with fallback?
- 245. How to implement async messaging with Kafka?
- 246. How to implement transactional messaging with Kafka?
- 247. How to implement monitoring and alerting with Prometheus?
- 248. How to implement dashboards with Grafana?
- 249. How to implement centralized exception handling?
- 250. How to implement high-availability microservices?

Java/Spring Boot Interview Questions

Advanced Core Java

1. What is JVM architecture?
2. Difference between JIT and AOT compilation?
3. What is class loading mechanism in Java?
4. Difference between static and dynamic class loading?
5. What are Java bytecode instructions?
6. Difference between stack frame and stack memory?
7. What is method area in JVM?
8. What is metaspace in Java 8+?
9. What is PermGen in earlier Java versions?
10. How does garbage collector work in JVM?
11. Difference between minor GC and major GC?
12. What is GC tuning?
13. Difference between Serial, Parallel, CMS, and G1 garbage collectors?
14. What is escape analysis?
15. What is scalar replacement?
16. How does Java allocate objects in Eden, Survivor, and Tenured spaces?
17. What is memory leak in Java?
18. How to detect memory leaks?
19. Difference between stack, heap, and metaspace?
20. How are local variables stored in stack memory?
21. How are objects stored in heap memory?
22. Difference between shallow copy and deep copy?
23. Difference between == and equals() in custom objects?
24. What is reflection in Java?
25. How to create objects dynamically using reflection?
26. How to access private fields using reflection?
27. What is annotations processing?
28. Difference between compile-time and runtime annotations?
29. How does ClassLoader work?
30. What is bootstrap, extension, and application class loader?
31. How to implement custom ClassLoader?
32. What is dynamic proxy in Java?
33. Difference between static proxy and dynamic proxy?
34. How to use Proxy.newProxyInstance()?
35. What are functional interfaces?
36. Difference between lambda expression and anonymous class?
37. What is Stream API in Java?
38. Difference between sequential and parallel streams?
39. What are intermediate and terminal operations in streams?
40. How to use filter(), map(), and reduce()?
41. What is Optional class?
42. Difference between Optional.of(), Optional.ofNullable(), and Optional.empty()?
43. How to avoid NullPointerException using Optional?
44. What is CompletableFuture?
45. How to combine multiple CompletableFutures?
46. Difference between thenApply(), thenAccept(), thenCompose()?

Java/Spring Boot Interview Questions

47. What is ForkJoinPool?
48. Difference between RecursiveTask and RecursiveAction?
49. How to implement parallelism using Fork/Join?
50. What is ReentrantLock and how does it work?
51. Difference between synchronized and ReentrantLock?
52. What is ReadWriteLock?
53. What is StampedLock?
54. Difference between fair and unfair locks?
55. What is a deadlock and how to detect it?
56. What is livelock?
57. What is thread starvation?
58. Difference between yield() and sleep()?
59. Difference between join() and wait()?
60. How to use CountDownLatch?
61. How to use CyclicBarrier?
62. How to use Phaser?
63. What is semaphore in Java?
64. What is blocking queue?
65. Difference between ArrayBlockingQueue and LinkedBlockingQueue?
66. Difference between ConcurrentHashMap and HashMap?
67. How to implement thread-safe collections?
68. What is atomic variable?
69. Difference between AtomicInteger and synchronized integer?
70. How to implement producer-consumer problem?
71. How to implement thread-safe singleton?
72. Difference between volatile and synchronized?
73. What is memory consistency error?
74. Difference between Java heap and native memory?
75. How to profile Java application memory and CPU usage?

Advanced Spring Boot

76. What is Spring Boot auto-configuration?
77. Difference between Spring MVC and Spring WebFlux?
78. What is reactive programming in Spring Boot?
79. What are Mono and Flux?
80. How to handle backpressure in reactive streams?
81. Difference between reactive and imperative programming?
82. How to implement REST API using WebFlux?
83. How to implement reactive repository with Spring Data R2DBC?
84. How to implement exception handling in WebFlux?
85. What is functional routing in WebFlux?
86. How to implement security in reactive Spring Boot?
87. What is reactive authentication manager?
88. How to use WebClient in Spring Boot?
89. Difference between RestTemplate and WebClient?
90. How to configure reactive data sources?
91. How to implement JWT authentication in Spring Boot?

Java/Spring Boot Interview Questions

92. How to implement OAuth2 login?
93. How to implement OAuth2 client credentials flow?
94. How to secure REST APIs with roles and permissions?
95. What is Spring Security filter chain?
96. How to implement CSRF protection?
97. How to implement method-level security?
98. How to implement CORS configuration?
99. How to implement exception handling globally?
100. How to use ControllerAdvice in Spring Boot?
101. How to configure Swagger / OpenAPI documentation?
102. How to customize Swagger UI?
103. How to implement Spring Boot actuator endpoints?
104. How to monitor application health?
105. How to implement custom health indicators?
106. How to monitor metrics (CPU, memory, DB connections)?
107. How to implement request logging?
108. How to implement structured logging?
109. How to implement centralized logging with ELK stack?
110. How to use Micrometer for metrics collection?
111. How to integrate Prometheus and Grafana?
112. How to implement Spring Boot with Docker?
113. How to create Dockerfile for Spring Boot application?
114. How to run multiple Spring Boot instances in Docker?
115. How to implement Spring Boot with Kubernetes?
116. How to implement configuration management in Kubernetes?
117. How to use ConfigMap and Secret in Kubernetes?
118. How to implement health probes in Kubernetes?
119. How to implement liveness and readiness probes?
120. How to implement reactive data repositories?
121. How to implement reactive MongoDB repository?
122. How to implement reactive Redis repository?
123. How to implement caching with Spring Cache?
124. How to implement distributed caching with Redis?
125. How to implement Spring Boot messaging with Kafka?
126. How to implement Spring Boot messaging with RabbitMQ?
127. How to implement asynchronous processing in Spring Boot?
128. How to use @Async annotation?
129. How to implement scheduled tasks in Spring Boot?
130. How to implement batch processing with Spring Batch?
131. How to implement error handling in Spring Batch?
132. How to implement retries in batch processing?
133. How to implement Spring Boot email service?
134. How to implement file upload/download REST APIs?
135. How to implement logging best practices in Spring Boot?
136. How to implement exception logging?
137. How to implement structured JSON logs?
138. How to implement Spring Boot with Micrometer metrics?
139. How to implement request tracing?
140. How to implement correlation ID across requests?
141. How to implement Spring Boot with Zipkin?

Java/Spring Boot Interview Questions

- 142. How to implement distributed tracing with Sleuth?
- 143. How to implement application monitoring best practices?
- 144. How to implement performance tuning in Spring Boot?
- 145. How to implement connection pooling with HikariCP?
- 146. How to implement data source routing?
- 147. How to implement multi-datasource configuration?
- 148. How to implement Spring Boot security with LDAP?
- 149. How to implement Spring Boot security with OAuth2?
- 150. How to implement Spring Boot security with JWT tokens?

Java/Spring Boot Interview Questions

Advanced Microservices & Scenario-Based Questions

151. What is microservices architecture?
152. Difference between monolith and microservices?
153. What is service discovery?
154. What is Eureka Server and Eureka Client?
155. What is Ribbon?
156. What is Feign Client?
157. What is Hystrix circuit breaker?
158. How to handle fallback methods in microservices?
159. What is Spring Cloud Gateway?
160. How to route requests in API gateway?
161. What is load balancing in microservices?
162. Difference between client-side and server-side load balancing?
163. What is distributed tracing?
164. What is Spring Cloud Sleuth?
165. How to propagate logs across microservices?
166. What is Zipkin?
167. How to handle centralized configuration in microservices?
168. What is Spring Cloud Config Server?
169. How to implement dynamic configuration refresh?
170. What is circuit breaker pattern?
171. How to implement retry strategy in microservices?
172. What is bulkhead pattern?
173. What is idempotency in microservices?
174. How to implement rate-limiting?
175. How to implement caching in microservices?
176. What is distributed caching?
177. How to use Redis in microservices?
178. How to implement messaging with RabbitMQ?
179. How to implement messaging with Kafka?
180. What is event-driven architecture?
181. How to handle eventual consistency?
182. What is saga pattern?
183. What is two-phase commit (2PC)?
184. How to implement API versioning?
185. How to implement global exception handling?
186. How to implement reactive APIs with Spring WebFlux?
187. Difference between WebFlux and Spring MVC?
188. What are Mono and Flux?
189. How to handle backpressure in reactive streams?
190. What is RSocket in Spring Boot?
191. How to implement JWT authentication across microservices?
192. How to implement OAuth2 authorization code flow?
193. How to implement refresh tokens?
194. How to secure inter-service communication?
195. What is service registry?
196. How to implement health checks for microservices?

Java/Spring Boot Interview Questions

197. What is Spring Boot actuator health indicators?
198. How to monitor CPU/memory usage in microservices?
199. How to implement logging and correlation IDs?
200. How to implement tracing with Sleuth and Zipkin?

Design Patterns & Best Practices

201. What is Singleton pattern? How to implement in Java?
202. What is Factory pattern?
203. What is Abstract Factory pattern?
204. What is Builder pattern?
205. What is Prototype pattern?
206. What is Observer pattern?
207. What is Strategy pattern?
208. What is Decorator pattern?
209. What is Adapter pattern?
210. What is Facade pattern?
211. How to implement Singleton in multithreaded environment?
212. How to implement Builder pattern for complex objects?
213. How to implement Observer pattern for event handling?
214. How to implement Strategy pattern for dynamic algorithms?
215. How to implement Decorator pattern for enhancing functionality?
216. How to implement Adapter pattern for integrating legacy code?
217. How to implement Facade pattern for simplifying complex subsystems?
218. How to implement Proxy pattern for caching or access control?
219. How to implement Template Method for standardizing workflow?
220. How to implement Command pattern for task scheduling?
221. How to implement Composite pattern?
222. How to implement Bridge pattern?
223. How to implement Chain of Responsibility pattern?
224. How to implement Mediator pattern?
225. How to implement Flyweight pattern?
226. How to implement Memento pattern?
227. How to implement State pattern?
228. How to implement Visitor pattern?
229. How to implement Iterator pattern?
230. How to implement Null Object pattern?
231. How to implement Dependency Injection pattern?
232. How to implement Inversion of Control pattern?
233. How to implement Template pattern?
234. How to implement Observer pattern with Java events?
235. How to implement Publisher-Subscriber pattern?
236. How to implement Singleton using Enum?
237. How to implement Thread-safe Singleton using double-checked locking?
238. How to implement Strategy pattern with Spring Beans?
239. How to implement Decorator pattern for REST APIs?
240. How to implement Adapter pattern for multiple APIs?
241. How to implement Facade pattern to simplify microservices calls?

Java/Spring Boot Interview Questions

- 242. How to implement Proxy pattern for API rate-limiting?
- 243. How to implement Template method for batch processing?
- 244. How to implement Observer pattern for logging events?
- 245. How to implement Command pattern for scheduling tasks?
- 246. How to implement Builder pattern for DTO objects?
- 247. How to implement Flyweight pattern for memory optimization?
- 248. How to implement Memento pattern for undo functionality?
- 249. How to implement State pattern for workflow management?
- 250. How to implement Visitor pattern for dynamic operations on objects?

Java/Spring Boot Interview Question

Advanced Microservices & Distributed Systems

1. How to design microservices for high scalability?
2. How to design microservices for high availability?
3. How to implement service discovery with Eureka?
4. How to implement client-side load balancing?
5. How to implement server-side load balancing?
6. What is API Gateway? Why use it?
7. Difference between Zuul and Spring Cloud Gateway?
8. How to implement routing and filtering in Gateway?
9. How to implement centralized configuration management?
10. What is Spring Cloud Config Server?
11. How to implement dynamic property refresh in microservices?
12. How to implement feature toggles?
13. How to implement distributed tracing?
14. What is Spring Cloud Sleuth?
15. How to integrate Sleuth with Zipkin?
16. How to implement distributed logging?
17. How to implement correlation IDs?
18. How to implement observability best practices?
19. How to monitor service health?
20. How to implement custom health indicators?
21. How to implement circuit breaker pattern?
22. What is Hystrix and why use it?
23. How to implement fallback methods?
24. What is Resilience4j?
25. How to implement retries with exponential backoff?
26. What is bulkhead pattern?
27. What is idempotency and why is it important?
28. How to implement rate-limiting in APIs?
29. How to implement distributed caching?
30. How to integrate Redis with microservices?
31. How to implement messaging with Kafka?
32. How to implement messaging with RabbitMQ?
33. What is event-driven architecture?
34. How to handle eventual consistency?
35. What is saga pattern for distributed transactions?
36. How to implement saga pattern in microservices?
37. Difference between choreography and orchestration in sagas?
38. What is two-phase commit (2PC)?
39. When to use 2PC vs Saga?
40. How to implement distributed transactions with Spring Boot?
41. How to handle partial failures in microservices?
42. How to implement fallback databases?
43. How to implement async communication between services?
44. How to implement synchronous communication between services?
45. What is gRPC and how is it used?
46. What is Thrift?
47. Difference between gRPC and REST?
48. How to implement service retries with Spring Retry?

Java/Spring Boot Interview Question

49. How to handle network failures in distributed systems?
50. How to implement backpressure handling?
51. How to implement message deduplication?
52. How to implement idempotent consumers?
53. How to implement transactional messaging?
54. How to handle database sharding?
55. How to implement multi-datasource configuration?
56. How to implement fallback strategies for database failures?
57. How to implement dynamic thread pools?
58. How to implement connection pooling?
59. How to optimize database queries in microservices?
60. How to implement caching for database queries?
61. How to implement distributed locking?
62. How to implement optimistic locking?
63. How to implement pessimistic locking?
64. How to implement async REST API calls?
65. How to implement WebSocket communication?
66. How to implement SSE (Server-Sent Events)?
67. How to implement real-time messaging with Kafka?
68. How to implement real-time messaging with RabbitMQ?
69. How to implement CQRS (Command Query Responsibility Segregation)?
70. How to implement Event Sourcing?
71. How to implement database versioning?
72. How to implement eventual consistency with event queues?
73. How to monitor SLAs of services?
74. How to implement service metrics collection?
75. How to implement dashboards with Prometheus/Grafana?
76. How to implement centralized exception handling?
77. How to implement service retries with circuit breaker?
78. How to implement graceful degradation?
79. How to implement high-throughput APIs?
80. How to implement caching for API responses?
81. How to implement API pagination?
82. How to implement API sorting?
83. How to implement API filtering?
84. How to implement API versioning?
85. How to implement canary deployment?
86. How to implement blue-green deployment?
87. How to implement rolling deployments?
88. How to implement CI/CD with microservices?
89. How to implement Kubernetes readiness/liveness probes?
90. How to implement service auto-scaling?
91. How to implement horizontal scaling of services?
92. How to implement vertical scaling of services?
93. How to implement chaos testing in microservices?
94. How to implement fault injection?
95. How to implement retries for messaging failures?
96. How to implement dead-letter queues?
97. How to implement monitoring of message brokers?
98. How to implement observability with distributed tracing?

Java/Spring Boot Interview Question

99. How to implement logging best practices for microservices?
100. How to implement security best practices in microservices?

Real-Time Scenario Questions

101. How would you design a real-time chat application?
102. How would you handle 1M concurrent users?
103. How would you design a stock market live feed system?
104. How would you implement backpressure for real-time streams?
105. How would you implement message ordering in distributed systems?
106. How would you design a notification system?
107. How would you implement rate-limiting for a messaging service?
108. How would you design a live scoreboard system?
109. How would you implement horizontal scaling for WebSocket connections?
110. How would you implement fault tolerance in a chat system?
111. How would you handle user presence in real-time applications?
112. How would you implement multi-region deployment?
113. How would you implement consistent hashing for distributed caches?
114. How would you implement leader election in distributed systems?
115. How would you design a distributed queue?
116. How would you implement a real-time bidding system?
117. How would you handle duplicate messages in distributed systems?
118. How would you implement priority queues for microservices?
119. How would you design a system for real-time analytics?
120. How would you implement in-memory caching for high-speed queries?
121. How would you implement Redis Pub/Sub for notifications?
122. How would you implement Kafka for high-throughput messaging?
123. How would you design a social media timeline system?
124. How would you design a collaborative document editing system?
125. How would you implement real-time file sharing?
126. How would you implement distributed rate-limiting?
127. How would you implement idempotent API calls?
128. How would you implement multi-tenant architecture?
129. How would you design a live video streaming service?
130. How would you implement adaptive bitrate streaming?
131. How would you design a push notification system for mobile apps?
132. How would you design a system for real-time leaderboard updates?
133. How would you implement optimistic locking for real-time updates?
134. How would you design a system to handle sudden traffic spikes?
135. How would you implement caching strategies for real-time feeds?
136. How would you implement eventual consistency for distributed counters?
137. How would you implement service retries for failed requests?
138. How would you implement load shedding for overloaded services?
139. How would you implement fault tolerance with circuit breakers?
140. How would you implement fallback responses for critical services?
141. How would you implement real-time fraud detection?
142. How would you implement monitoring for real-time transactions?
143. How would you implement alerting for critical system failures?
144. How would you implement distributed tracing for real-time systems?
145. How would you implement analytics for live traffic monitoring?

Java/Spring Boot Interview Question

146. How would you implement dynamic throttling for real-time APIs?
147. How would you implement batch processing along with real-time processing?
148. How would you implement idempotency in messaging systems?
149. How would you implement priority-based message processing?
150. How would you implement horizontal scaling for real-time microservices?
151. How would you implement leaderboards with millions of users?
152. How would you implement low-latency messaging for gaming apps?
153. How would you implement streaming ETL for real-time data pipelines?
154. How would you implement message ordering in Kafka?
155. How would you implement partitioning in Kafka?
156. How would you implement Kafka consumer group scaling?
157. How would you implement fault-tolerant Kafka producers?
158. How would you implement exactly-once semantics in Kafka?
159. How would you implement transactional writes in Kafka?
160. How would you implement backpressure handling in reactive systems?
161. How would you implement SSE in microservices?
162. How would you implement WebSockets for millions of connections?
163. How would you implement sticky sessions?
164. How would you implement session replication?
165. How would you implement caching strategies for live dashboards?
166. How would you implement queue prioritization?
167. How would you implement latency monitoring for real-time systems?
168. How would you implement hotspot detection in distributed caches?
169. How would you implement failover strategies in real-time pipelines?
170. How would you implement blue-green deployments for microservices?
171. How would you implement canary deployments for critical APIs?
172. How would you implement real-time metrics collection?
173. How would you implement service observability in Kubernetes?
174. How would you implement alerting for SLA violations?
175. How would you implement monitoring dashboards for microservices?
176. How would you implement automated rollback for failed deployments?
177. How would you implement hot code reload in production?
178. How would you implement configuration refresh in running services?
179. How would you implement high-availability Kafka cluster?
180. How would you implement cross-region replication for messaging?
181. How would you implement distributed locks for critical sections?
182. How would you implement atomic counters across services?
183. How would you implement leader election for batch jobs?
184. How would you implement consistency for distributed caches?
185. How would you implement rate-limiting with Redis?
186. How would you implement service throttling?
187. How would you implement distributed tracing in multi-cluster setups?
188. How would you implement graceful degradation for dependent services?
189. How would you implement circuit breaker metrics and alerts?
190. How would you implement bulkhead pattern for heavy services?
191. How would you implement Saga pattern for multi-service transactions?
192. How would you implement orchestration vs choreography in sagas?
193. How would you implement multi-step workflows in microservices?
194. How would you implement retry with exponential backoff?
195. How would you implement dead-letter queues?

Java/Spring Boot Interview Question

- 196. How would you implement message replay?
- 197. How would you implement transaction compensation?
- 198. How would you implement distributed semaphore?
- 199. How would you implement distributed queue monitoring?
- 200. How would you implement service SLAs tracking?

System Design Patterns & Best Practices

- 201. How to implement Singleton pattern in microservices?
- 202. How to implement Factory pattern in Spring Boot?
- 203. How to implement Builder pattern for DTOs?
- 204. How to implement Observer pattern for events?
- 205. How to implement Strategy pattern for dynamic logic?
- 206. How to implement Adapter pattern for legacy services?
- 207. How to implement Facade pattern for simplifying APIs?
- 208. How to implement Proxy pattern for caching?
- 209. How to implement Decorator pattern for cross-cutting concerns?
- 210. How to implement Template pattern for workflows?
- 211. How to implement Command pattern for task scheduling?
- 212. How to implement Chain of Responsibility pattern?
- 213. How to implement Mediator pattern for service orchestration?
- 214. How to implement Flyweight pattern for memory optimization?
- 215. How to implement Memento pattern for undo operations?
- 216. How to implement State pattern for workflow management?
- 217. How to implement Visitor pattern for dynamic operations?
- 218. How to implement Null Object pattern?
- 219. How to implement Dependency Injection pattern?
- 220. How to implement Inversion of Control pattern?
- 221. How to implement logging patterns in microservices?
- 222. How to implement monitoring best practices?
- 223. How to implement failover patterns?
- 224. How to implement retry patterns?
- 225. How to implement circuit breaker patterns?
- 226. How to implement bulkhead pattern in real systems?
- 227. How to implement event sourcing pattern?
- 228. How to implement CQRS pattern?
- 229. How to implement saga orchestration pattern?
- 230. How to implement distributed caching patterns?
- 231. How to implement idempotency patterns?
- 232. How to implement rate-limiting patterns?
- 233. How to implement distributed locking patterns?
- 234. How to implement retry with exponential backoff pattern?
- 235. How to implement graceful degradation pattern?
- 236. How to implement fail-fast pattern?
- 237. How to implement health check patterns?
- 238. How to implement observability patterns?
- 239. How to implement metrics collection patterns?
- 240. How to implement alerting patterns?

Java/Spring Boot Interview Question

- 241. How to implement monitoring dashboards patterns?
- 242. How to implement CI/CD patterns for microservices?
- 243. How to implement blue-green deployment patterns?
- 244. How to implement canary deployment patterns?
- 245. How to implement logging correlation patterns?
- 246. How to implement tracing patterns?
- 247. How to implement error handling patterns?
- 248. How to implement message deduplication patterns?
- 249. How to implement distributed transaction patterns?
- 250. How to implement workflow orchestration patterns?
- 251. How to implement multi-region deployment patterns?
- 252. How to implement cross-region data replication?
- 253. How to implement API versioning patterns?
- 254. How to implement multi-tenant architecture patterns?
- 255. How to implement session management patterns?
- 256. How to implement WebSocket architecture patterns?
- 257. How to implement server-sent events architecture?
- 258. How to implement reactive pipelines patterns?
- 259. How to implement asynchronous processing patterns?
- 260. How to implement scheduled tasks patterns?
- 261. How to implement batch processing patterns?
- 262. How to implement distributed workflow patterns?
- 263. How to implement distributed queues patterns?
- 264. How to implement microservice scaling patterns?
- 265. How to implement load balancing patterns?
- 266. How to implement fault tolerance patterns?
- 267. How to implement chaos engineering patterns?
- 268. How to implement distributed semaphore patterns?
- 269. How to implement distributed counter patterns?
- 270. How to implement caching patterns?
- 271. How to implement performance tuning patterns?
- 272. How to implement resiliency patterns?
- 273. How to implement throttling patterns?
- 274. How to implement service discovery patterns?
- 275. How to implement observability for high throughput systems?
- 276. How to implement monitoring for real-time systems?
- 277. How to implement logging for multi-service architectures?
- 278. How to implement exception handling for distributed systems?
- 279. How to implement rate-limiting in reactive systems?
- 280. How to implement retries for message queues?
- 281. How to implement event-driven microservices patterns?
- 282. How to implement fault-tolerant event sourcing?
- 283. How to implement saga pattern with compensation?
- 284. How to implement distributed transaction rollback?
- 285. How to implement async messaging with Kafka?
- 286. How to implement async messaging with RabbitMQ?
- 287. How to implement priority queue patterns?
- 288. How to implement message replay patterns?
- 289. How to implement distributed locks in Redis?
- 290. How to implement concurrency handling patterns?

Java/Spring Boot Interview Question

- 291. How to implement reactive streaming patterns?
- 292. How to implement backpressure patterns?
- 293. How to implement real-time monitoring patterns?
- 294. How to implement alerting and SLA monitoring?
- 295. How to implement service resiliency patterns?
- 296. How to implement real-time API analytics?
- 297. How to implement distributed tracing patterns?
- 298. How to implement performance benchmarking patterns?
- 299. How to implement failover testing patterns?
- 300. How to implement high-availability microservices architecture?