

1. Basics & Prerequisites

Java & Backend Fundamentals

- Core Java (OOP, Collections, Streams, Exception Handling)
 - Java 8+ features (Lambda, Streams, Optional)
 - Maven / Gradle
 - REST API basics
 - HTTP methods (GET, POST, PUT, DELETE, PATCH)
 - HTTP status codes
 - JSON & Jackson
 - JPA & Hibernate basics
 - SQL basics
-

2. Introduction to Microservices Architecture

- Monolithic vs Microservices
 - SOA vs Microservices
 - Advantages & Disadvantages of Microservices
 - Microservices Characteristics
 - Real-world Microservices Use Cases
 - CAP Theorem
 - 12-Factor App Principles
-

3. Spring Boot Fundamentals

- What is Spring Boot?
 - Spring Boot vs Spring Framework
 - Spring Boot Architecture
 - Starter Dependencies
 - Auto Configuration
 - Spring Boot Project Structure
 - `application.properties` vs `application.yml`
 - Profiles (`dev`, `test`, `prod`)
 - `CommandLineRunner` & `ApplicationRunner`
-

4. Building REST APIs with Spring Boot

- `@RestController`
- `@RequestMapping`, `@GetMapping`, `@PostMapping`
- `PathVariable` vs `RequestParam`
- `RequestBody` & `ResponseBody`

- DTO Design Pattern
 - Validation (@Valid, @NotNull, @Size)
 - Exception Handling
 - @ExceptionHandler
 - @ControllerAdvice
 - ResponseEntity usage
-

5. Database Integration

Spring Data JPA

- JPA Architecture
- Entity Mapping
- Relationships
 - OneToOne
 - OneToMany
 - ManyToMany
- Fetch Types (Eager vs Lazy)
- Cascade Types
- JPQL
- Native Queries
- Pagination & Sorting
- Auditing (CreatedDate, UpdatedDate)

Databases

- MySQL
 - PostgreSQL
 - Oracle
 - MongoDB (NoSQL – optional)
-

6. Microservices Design Patterns

- Database per Service
 - API Gateway Pattern
 - Service Registry Pattern
 - Circuit Breaker Pattern
 - Saga Pattern
 - Event-Driven Architecture
 - CQRS (Basic understanding)
-

7. Service Discovery (Eureka)

- What is Service Discovery?
 - Netflix Eureka Architecture
 - Eureka Server setup
 - Eureka Client configuration
 - Service Registration & Discovery
 - Load balancing with Eureka
-

8. API Gateway

Spring Cloud Gateway

- What is API Gateway?
 - Gateway vs Load Balancer
 - Routing Configuration
 - Predicate Filters
 - Global Filters
 - Custom Filters
 - Rate Limiting
 - Security at Gateway level
-

9. Inter-Service Communication

RestTemplate (Deprecated – Basics)

- Synchronous communication

WebClient (Recommended)

- Reactive REST calls
- Mono & Flux basics

OpenFeign

- Feign Client setup
 - Feign with Eureka
 - Error Handling
 - Feign Interceptors
-

10. Centralized Configuration

Spring Cloud Config Server

- Why Config Server?
 - Config Server Architecture
 - Git-based Configuration
 - Refresh Configurations
 - @RefreshScope
 - Environment-specific configs
-

11. Fault Tolerance & Resilience

Resilience4j

- Circuit Breaker
 - Retry
 - Rate Limiter
 - Bulkhead
 - Fallback Methods
 - Monitoring failures
-

12. Distributed Logging & Tracing

- Logging best practices
 - SLF4J & Logback
 - Centralized logging concepts
 - Sleuth (Trace ID, Span ID)
 - Zipkin overview
-

13. Security in Microservices

Spring Security Basics

- Authentication vs Authorization
- Filters & Security Context

JWT Security

- JWT Architecture
- Token Generation
- Token Validation
- Role-based Authorization
- Securing APIs
- Securing Gateway

OAuth2 (Overview)

- Authorization Server
 - Resource Server
 - Keycloak (Intro)
-

14. Messaging & Event-Driven Microservices

Apache Kafka

- Kafka Architecture
- Topics, Partitions
- Producers & Consumers
- Consumer Groups
- Kafka with Spring Boot
- Kafka Serialization (JSON, Avro)
- Error Handling
- Retry & Dead Letter Topic

RabbitMQ (Optional)

- Exchanges
 - Queues
 - Routing keys
-

15. Distributed Transactions

- Problem with transactions in Microservices
 - Saga Pattern
 - Choreography
 - Orchestration
 - Compensation Transactions
-

16. Microservices Testing

- Unit Testing (JUnit 5)
- Mockito
- Integration Testing
- TestContainers
- Contract Testing (Intro)
- Postman Collections

17. Containerization

Docker

- Docker Architecture
 - Dockerfile
 - Docker Image
 - Docker Container
 - Docker Compose for Microservices
 - Multi-service setup
-

18. Deployment & Cloud

Cloud Concepts

- Cloud basics
- AWS overview

AWS Services (Microservices)

- EC2
 - RDS
 - S3
 - IAM
 - Load Balancer
 - ECS / EKS (Intro)
-

19. Monitoring & Observability

- Actuator
 - Health Checks
 - Metrics
 - Prometheus (Overview)
 - Grafana (Overview)
-

20. Performance & Scalability

- Horizontal vs Vertical Scaling
- Load Balancing

- Caching Strategies
 - Redis Basics
 - API performance tuning
-

21. Real-Time Microservices Projects (IMPORTANT)

Example Projects

1. **Order Service**
2. **Payment Service**
3. **Inventory Service**
4. **User Service**
5. **Notification Service**

Features

- Eureka
 - API Gateway
 - JWT Security
 - Kafka integration
 - Central Config Server
 - Dockerized services
 - Inter-service communication
-

22. Interview Preparation (Microservices)

Common Questions

- Monolith vs Microservices
 - How API Gateway works?
 - How services communicate?
 - How to handle failures?
 - How to secure microservices?
 - How to manage configurations?
 - Kafka vs REST
 - Database per service
-

23. Best Practices

- Proper service boundaries
- DTO usage
- Avoid tight coupling

- Centralized logging
- Versioning APIs
- Backward compatibility