
SPRING APACHE KAFKA

1: Introduction to Spring and Apache Kafka Integration

1.1. Introduction to Spring and Kafka

- 1.1.1. Overview of Apache Kafka
- 1.1.2. Use Cases for Kafka in Spring Applications

1.2. Setting Up the Development Environment

- 1.2.1. Installing Kafka and Zoo Keeper
- 1.2.2. Configuring Spring Framework

2: Kafka Fundamentals

2.1. Kafka Architecture

- 2.1.1. Topics, Producers, and Consumers
- 2.1.2. Partitions and Brokers

2.2. Kafka Command-Line Tools

- 2.2.1. Kafka CLI for Basic Operations
- 2.2.2. Kafka Configuration Files

3: Spring Kafka Basics

3.1. Spring Kafka Overview

- 3.1.1. Spring Kafka vs. Native Kafka Clients
- 3.1.2. Key Concepts in Spring Kafka

3.2. Configuring Kafka Templates and Producers

- 3.2.1. Creating Kafka Templates
- 3.2.2. Sending Messages with Producers

4: Kafka Consumers with Spring

4.1. Spring Kafka Consumers

- 4.1.1. Consuming Kafka Messages
- 4.1.2. Message Listener Containers

4.2. Error Handling and Offset Management

- 4.2.1. Handling Exceptions in Consumers
- 4.2.2. Managing Offsets and Consumer Groups

5: Spring Kafka Advanced Topics

5.1. Custom Serialization and Deserialization

- 5.1.1. Implementing Custom Serdes
- 5.1.2. JSON and Avro Serialization

5.2. Batch Processing with Kafka

- 5.2.1. Consuming Messages in Batches
- 5.2.2. Configuring Batch Listeners

6: Kafka Streams with Spring

6.1. Introduction to Kafka Streams

- 6.1.1. Stream Processing with Kafka
- 6.1.2. Stateful vs. Stateless Processing

6.2. Building Kafka Stream Applications with Spring

- 6.2.1. Configuring Kafka Streams
- 6.2.2. Stream Processing Topologies

7: Spring Kafka and Microservices

7.1. Microservices and Event-Driven Architecture

- 7.1.1. Event Sourcing and CQRS
- 7.1.2. Decoupling Microservices with Kafka

7.2. Spring Cloud Stream for Kafka

- 7.2.1. Implementing Event-Driven Microservices
- 7.2.2. Spring Cloud Stream Binders

8: Kafka Connect and Integration

8.1. Kafka Connect Overview

- 8.1.1. Connectors and Tasks
- 8.1.2. Source and Sink Connectors

8.2. Configuring Kafka Connect with Spring

- 8.2.1. Creating Connect Configurations
- 8.2.2. Sink Connectors for Data Sync

9: Testing and Debugging Spring Kafka Applications

9.1. Unit Testing with Embedded Kafka

- 9.1.1. Testing Kafka Components
- 9.1.2. Setting Up Embedded Kafka

9.2. Debugging Kafka Consumer Issues

- 9.2.1. Troubleshooting Kafka Consumers
- 9.2.2. Monitoring and Metrics

10: Kafka Security and Scalability

10.1. Securing Kafka with Spring Security

- 10.1.1. Authentication and Authorization
- 10.1.2. SSL Encryption for Kafka

10.2. Scaling Kafka and Spring Applications

- 10.2.1. Horizontal Scaling Strategies
- 10.2.2. Load Balancing Kafka Consumers

11: Real-World Kafka Use Cases

11.1. Log Aggregation with Kafka

- 11.1.1. Centralized Logging with Kafka
- 11.1.2. Log Ingestion and Analysis

11.2. Event Sourcing and Data Pipelines

- 11.2.1. Building Event-Driven Data Pipelines

- 11.2.2. Event Sourcing for Stateful Applications

12: Best Practices and Optimization

12.1. Kafka Best Practices

- 12.1.1. Design and Configuration Recommendations
- 12.1.2. Handling High Throughput

12.2. Spring Kafka Optimization

- 12.2.1. Performance Tuning and Efficiency
- 12.2.2. Scaling for Production

13: Final Project and Course Review

13.1. Project Proposal and Planning

- 13.1.1. Designing a Kafka-Integrated Application
- 13.1.2. Project Development Phases

13.2. Implementation and Presentation

- 13.2.1. Building the Kafka-Integrated Project
- 13.2.2. Final Project Presentation