DevOpsSchool
Lets Learn, Share & Practice DevOps

Apache Camel Enterprise Integration Training

Curriculum 5 Days

Day - 1

Core Concepts & First Steps with Camel

- **Objectives**
  - Understand Apache Camel architecture, core concepts, and DSLs
  - Build foundational routes using Java DSL and Spring Boot

- **Topics**
  - Introduction to Apache Camel & Camel 4.x evolution
  - Integration challenges in distributed systems
  - Enterprise Integration Patterns (EIP) overview
  - Camel context, route, endpoint, exchange, processor
  - DSLs: Java, XML, YAML
  - Setting up Camel with Maven, Gradle, and Spring Boot
  - Creating your first Camel route: file-to-log, file-to-file
  - Camel timer and scheduler components
  - Configuring Data Sources (DB: AS400, SQL Server)
  - Performing DB lookups and dynamic queries

- **Hands-On Labs**
  - Build basic file-based routes
  - Configure a route using timer and log components
  - Connect to database using Camel JDBC

Day - 2

Routing, Transformation, EIPs & Orchestration

- **Objectives**
  - Implement advanced EIPs and transformation logic
  - Handle conditional routing, orchestration, and error handling
- **Topics**
  - Content-based router (choice/when/otherwise)
  - Splitter, Aggregator, WireTap, Multicast
  - Recipient List, Dynamic Router
  - Message patterns: InOnly vs InOut
  - Custom Processors and Beans
  - Exception handling: doTry/doCatch/onException
  - Redelivery policies, exponential backoff
  - Dead Letter Channel
  - JSON, XML, CSV, Avro transformation (marshal/unmarshal)
  - XSLT in Camel routes
  - XML <-> JSON (Jackson/JAXB)
  - XML to XML (SOAP to SOAP)
  - Bean binding, type conversion
  - UTF-16 and encoding support (UTF-8, ISO)
  - GZIP compression/decompression
  - Base64 encoding/decoding (headers and payloads)
- **Hands-On Labs**
  - Create a conditional router and aggregator route
  - Add transformation logic using JSON/XML
  - Implement complex mappings using custom processors

Day - 3

Integration with External Systems (DB, Files, SOAP, REST, Email, SAP)

- **Objectives**
  - Build integrations with external systems: databases, APIs, files, queues, SAP, and email

- **Topics**
  - File Handling: Local/SFTP/FTP, polling, idempotency, encoding (pipe-delimited, UTF variants)
  - Error handling and retries for file operations
  - Database: JDBC/SQL components, named parameters, stored procedures, result mapping
  - Transactions, idempotent operations, caching DB table lookups
  - SOAP: CXF integration, headers, fault responses, WS-Security (username tokens, certificates)
  - REST APIs: Exposing endpoints, consuming APIs, handling headers, query/path parameters
  - Token-based APIs, timeouts, secured REST APIs
  - Email: IMAP/SMTP, HTML emails, attachment handling
  - SAP: Camel-SAP, JCo, BAPI, IDoc integration, queues, request-reply
  - Queues: JMS, AMQP, ActiveMQ, Artemis, durable subscriptions, selectors, DLQ

- **Hands-On Labs**
  - SOAP-to-REST transformation with headers and WS-Security
  - Implement email integration with attachments
  - File-to-queue and queue-to-database route
  - SAP request-reply messaging route

Day - 4

Aggregation, Orchestration, Scaling, Resilience, Security

- **Objectives**
  - Design robust, scalable, and secure Camel routes
  - Implement orchestration, aggregation, and resilience patterns

- **Topics**
  - Aggregation & Orchestration: Aggregator EIP, wireTap, multicast, correlation strategies, timeout handling
  - Manual vs. scheduled vs. event-driven routing
  - Scaling & Resilience: SEDA/VM, async threading, executorServiceRef, thread pools
  - Circuit breakers: Hystrix, Resilience4j
  - Load balancing, throttling, backpressure strategies
  - Security: HTTPS, OAuth2, JWT, API key validation, IP/domain whitelisting
  - WS-Security and SSL trust strategies
  - Encryption/Decryption (PGP, JCE)
  - Azure Key Vault integration for secrets

- **Hands-On Labs**
  - Implement parallel processing with SEDA/multicast
  - Create a secure REST API using token headers and SSL
  - Partial aggregation with wireTap and timeout

Day - 5

Real-World Projects, CI/CD, Monitoring, Capstone

- **Objectives:**
  - Implement advanced use cases with monitoring, CI/CD, caching, and scheduling
  - Deliver an end-to-end integration scenario
- **Topics:**
  - Large File Processing: Stream-based processing (10GB+), memory management, chunk reading
  - Caching: Ehcache/Caffeine, DB cache invalidation, memory-safe caching
  - Event-Driven Scheduling: Timer, Quartz, delayed triggers, manual vs. scheduled routes
- **Monitoring & CI/CD:**
  - Implement parallel processing with SEDA/multicast
  - Create a secure REST API using token headers and SSL
  - Partial aggregation with wireTap and timeout
- **Capstone Project:**
  - Scenario:
    - Read UTF-16 files from SFTP
    - Convert to JSON and validate schema
    - Store in SQL Server
    - Notify via Kafka
    - Expose result via secured REST API
- **Wrap-Up & Certification:**
  - Review EIPs, error handling, DSLs
  - Interview prep + scenario design
  - Share projects via GitHub
  - Post-training Q&A and certification