Camunda Developer Training _Content

Duration: 40 Hours / 5 Days

Content:

Day 1 – Foundations (Camunda 7 & 8 Basics + Modeling)

Camunda Overview

- Evolution: Camunda 7 vs Camunda 8 (Key differences & similarities)
- Architecture overview (Workflow engine, Zeebe, Operate, Tasklist, Optimize)
- When to use 7 vs 8 in projects
- **Lab:** Spin up Camunda 7 (Spring Boot starter) and Camunda 8 (Docker/Cloud) environments.
- **Task:** Deploy a "Hello World" BPMN process in both.

BPMN Essentials

- Core elements (Tasks, Events, Gateways, Subprocesses)
- Modeling best practices for developers
- **Lab:** Create a process with User Task, Service Task, and End Event in IDE.
- **Task:** Deploy to both Camunda 7 & 8 and compare execution.
- **Modeling in Web Modeler vs IDE (Camunda Modeler / bpmn-js)**
- **Lab:** Import/export BPMN models between Web Modeler and Camunda Modeler.
- **Task:** Apply version control (Git) for BPMN diagrams.

Day 2 – Advanced Modeling & Decision Automation

BPMN Advanced

- Error Handling, Compensation, Event Subprocesses
- **Lab:** Extend Day 1 process with error boundary event & compensation handler.
- **Task:** Trigger failures and observe incident handling in Operate (Camunda 8) and Cockpit (Camunda 7).

- Multi-instance, Call Activities, Collaboration patterns
- **Lab:** Model 2 collaborating processes (Order & Payment) using message events.
- **Task:** Trigger cross-process communication.

Long-Running Processes & State Management

- Persistence, timers, escalation, and compensation in long-running workflows.
- **Lab:** Create an order approval workflow with intermediate timer events and escalation via message events.
- **Task:** Suspend, resume, and migrate an active long-running instance (Camunda 7 Cockpit / Camunda 8 Operate).

DMN Overview

- Hit policies, DRDs, best practices for decision modelling
- **Lab:** Build a DMN table for loan approval with hit policies.
- **Task:** Integrate DMN into BPMN process execution.

Camunda Forms

- Advanced form creation & AI-assisted builder
- Task forms in Camunda 7 vs Camunda 8 differences
- **Lab:** Create advanced task forms in Web Modeler.
- **Task:** Build an AI-assisted form for a User Task.

Day 3 – Developer Implementation in IDE

Developer Setup

- Camunda 7 SDK (Java, Spring Boot)
- Camunda 8 Client (Zeebe Java Client, Spring Zeebe)
- Using External Task / Worker pattern in 7 & 8
- **Lab:** Create a Java/Spring Boot Camunda 7 project with External Task client.
- **Lab:** Create a Spring Zeebe worker for Camunda 8.
- **Task:** Implement a Service Task worker in both.

Connectors & Integration

- Built-in connectors (REST, Kafka, DB, etc.)
- Create custom connectors (Inbound & Outbound)
- Connector templating from API specifications
- **Lab:** Use built-in REST connector to call external API.
- **Lab:** Integrate with Kafka producer/consumer.
- **Task:** Process events via Kafka and trigger workflow execution.
- **Lab:** Create a custom outbound connector (Java worker) that writes to DB.
- **Task:** Deploy connector & reuse it in multiple processes.

Fallbacks & Retry Strategies

- Worker retries, exponential backoff, incident handling, fallback subprocesses.
- **Lab:** Implement a service task with retry + error boundary event fallback (alternate API call or manual task).
- **Task:** Test failover by simulating worker/service downtime.

Day 4 – Microservices, Testing & CI/CD

Microservice Orchestration Patterns (Sagas, Choreography vs Orchestration)

Lab: Implement Saga pattern with BPMN (compensating transactions).
Task: Compare orchestration vs choreography for an order flow.

Working with APIs

- Zeebe API (CLI & Java Client)
- Camunda 7 REST API
- **Lab:** Use Zeebe CLI to start and inspect process instances.
- **Lab:** Call Camunda 7 REST API for instance creation and task completion.

Testing & Debugging

- JUnit tests for BPMN & DMN
- Mocking services for workflows
- Error handling & retries

- **Lab:** Write JUnit test for BPMN process execution (mocking service tasks).
- **Task:** Trigger incident and test retry logic.

CI/CD & Deployments

- Packaging & deployment (Docker/Kubernetes)
- Automating deployments (Helm, GitOps, Jenkins/GitHub Actions pipelines)
- Environment strategies (dev/test/prod)
- **Lab:** Containerize a Camunda worker with Docker.
- **Lab:** Use GitHub Actions/Jenkins pipeline to auto-deploy BPMN to Camunda 8 cluster.
- **Task:** Promote deployments from dev → test → prod.

Deployment & Management to CaaS/AWS

- Best practices for Helm, GitOps, autoscaling, observability on AWS.
- **Lab:** Deploy Camunda 8 cluster on Kubernetes (minikube/Kind).
- **Lab:** Deploy on AWS EKS with RDS/Postgres persistence.
- **Task:** Configure scaling (workers & brokers) and monitor in Operate.

Day 5 – Operations, Monitoring & Best Practices

Operate & Optimize

- Instance modification, migration, incident handling
- Process monitoring & reporting
- **Lab:** Use Operate to modify an active process instance.
- **Lab:** Use Optimize to generate process performance reports.

Best Practices for Developers

- BPMN/DMN design guidelines
- Worker scaling & error handling
- Security & Authentication (OAuth2, Keycloak integration)
- Versioning & process migrations
- **Lab:** Refactor BPMN model using design guidelines (naming conventions, boundaries).

- **Lab:** Implement OAuth2 authentication for workers with Keycloak.

Case Study Project

End-to-end implementation:

- Order-to-Cash workflow with BPMN + DMN
- Kafka integration for event-driven messaging
- REST API & DB integration
- CI/CD pipeline deployment
- Monitoring with Operate
- Handling long-running processes, fallbacks, and AWS deployment

Wrap-up & Q&A