RESTful API Development with Spring Boot and RabbitMQ

Duration: 10 days
Prerequisites: Working Knowledge of API and Java Spring

Day 1 – Introduction to REST & Spring Boot

Topics

- REST vs SOAP
- HTTP methods & status codes
- JSON vs XML
- Spring Boot ecosystem overview
- Project setup with Spring Initializr
- Folder structure and Maven dependencies
- Running a Spring Boot application

Lab 1:

- Install JDK & IntelliJ / STS
- Create a "Hello Spring Boot" REST API returning JSON
- Explore auto-configuration and port setup

Day 2 – Spring Boot Core Concepts

Topics

- @SpringBootApplication, @Component, @Service, @Repository
- Dependency Injection and IoC Container
- Configuration files (application.properties vs YAML)
- Profiles (dev, test, prod)
- REST Controller and @RequestMapping

Lab 2:

- Build a multi-layer "Student Management" API
- Use @Service and @Repository to separate logic
- Switch between profiles (dev/prod)

Day 3 – Creating RESTful Endpoints

Topics

- Designing REST endpoints
- Path variables and query parameters
- ResponseEntity and status codes
- Exception handling (@ControllerAdvice)
- Logging and Spring Boot Actuator

Lab 3:

- Add CRUD operations (GET/POST/PUT/DELETE) for Student entity
- Implement custom error response structure
- Monitor API health via Actuator endpoints

Day 4 – Data Persistence with Spring Data JPA

Topics

- ORM basics and JPA annotations
- Connect to MySQL/PostgreSQL
- JpaRepository and custom queries
- DTO pattern and ModelMapper
- Transaction management

Lab 4:

- Configure MySQL database in Spring Boot
- Create entity classes (Student, Course)
- Perform CRUD operations with JpaRepository

Day 5 – Advanced API Design

Topics

- Validation (@Valid, custom validators)
- Pagination and sorting
- API versioning strategies
- Swagger/OpenAPI integration
- Global CORS setup and error responses

Lab 5:

- Add Bean Validation to Student entity
- Implement pagination and sorting on GET endpoint
- Generate API docs with Swagger UI

Day 6 – Introduction to RabbitMQ

Topics

- Message brokers and AMQP protocol
- RabbitMQ architecture (exchange, queue, binding)
- Installing RabbitMQ server & console
- Producer-Consumer pattern

Lab 6:

- Install and run RabbitMQ locally
- Create queue and exchange via management UI
- Send and receive test messages using CLI

Day 7 – Spring Boot Integration with RabbitMQ

Topics

- Spring AMQP and RabbitTemplate
- ConnectionFactory and MessageConverter
- @RabbitListener and manual acknowledgment
- Error handling and retries

Lab 7:

- Integrate RabbitMQ with Spring Boot
- Implement Producer service to publish JSON messages
- Implement Consumer service to listen and log messages

Day 8 – Building Event-Driven APIs

Topics

- Event-driven architecture overview
- Asynchronous REST flows
- JSON message conversion
- Publisher / Consumer pattern in API

Lab 8:

- Extend Student API to publish events on student creation
- Create separate Notification service as RabbitMQ consumer
- Validate message flow via management console

Day 9 – Security & Testing

Topics

- Introduction to Spring Security and JWT
- Role-based authorization
- Unit testing with JUnit and Mockito
- Testing RabbitMQ integration
- API testing with Postman

Lab 9:

- Secure Student API with Basic Auth / JWT
- Write unit tests for controllers and services

- Test endpoints and message queues using Postman

Day 10 – Project Implementation & Deployment

Topics

- Capstone Project: Order Processing System
- API workflow: Order → Queue → Notification
- Swagger docs integration
- Packaging and deployment (JAR, Docker)
- Course summary and best practices

Lab 10 (Capstone):

- Develop “Order Processing System with RabbitMQ”
    - Order Service publishes order events
    - Notification Service consumes and sends email/log
- Build Docker image and run containers
- Test end-to-end API workflow