

Spring Boot Training

Duration: 4 Days · **Training format:** Virtual Classroom / Online

Description

In today's software world, Spring Boot has become one of the most popular solutions for developing fast, scalable, and secure applications. In this comprehensive training, you will learn step by step, from the core components of Spring Boot to microservice architecture, enabling you to build modern, enterprise-level applications.

Topics Covered:

- **Spring Boot Fundamentals:** Project structure, IOC, DI, Restful API, error handling
- **Data Management:** Spring Data JPA, entity relationships, DAO - DTO transformations
- **Security:** JWT Authentication, Spring Security
- **Testing & Debugging:** Mockito, MockMVC, debugging techniques
- **Docker & Kafka:** Docker integration, event-driven architecture with Kafka
- **Microservice Architecture:** Spring Cloud, Eureka, Feign Client, Circuit Breaker, API Gateway

This training not only covers the theoretical aspects of Spring Boot but also provides hands-on applications, equipping you with the skills to develop real-world solutions. Through practical exercises and live demo applications, you will solidify your learning experience.

Audience

Software developers who want to build scalable applications with Spring Boot Backend
developers looking to transition into the Spring ecosystem Engineers interested in microservice architecture and distributed systems Professionals aiming to develop enterprise-grade Spring Boot projects

By the end of this training, you will have a strong command of Spring Boot and be capable of building secure, scalable, and production-ready applications.

Outline

Spring Boot Project Creation

- start.spring.io Interface
- Project Creation Parameters

Spring Boot Project Structure

- **Introduction to the Spring Framework**
 - Application Context
 - IOC / DI Concepts
 - Spring Bean Definitions
- Project Package Structure
- Application Layers
- Inter-layer Communication

REST API

- RESTful Service Standards
- Response and Request Structures
- Validation and Custom Validation Definitions

REST API Standardization with OpenAPI

- Swagger UI
- Controller Interface
- Request/Response Generation

Exception Handling

- Error Handling Standards
- Validation Errors
- Error Messages

Service Layer

- Service Layer Architecture
- Inter-service Communication and Bean Injection
- Service - Database Communication

Spring JPA and Data Entities

- Why Don't We Use Lombok?!
- Spring Data JPA Repository Types and Definitions
- Spring Data Entity Definitions
- Spring Data Entity Relationship Mapping
 - One-to-One
 - One-to-Many
 - Many-to-Many
- BaseEntity Definitions

DAO - DTO Transformations

- Conversion Methods
- MapStruct

REST API Design Patterns with Spring Boot

- Strategy Design Pattern
- Singleton Design Pattern
- Enum Type Factory Method Design Pattern

Database Migration

- Using Flyway for Database Migrations

Mockito Framework

- Mock Usage
- Controller Testing with MockMVC
- Using WireMock

Spring Security

- InMemory Authentication
- Basic Authentication
- JWT Token Authentication and Authorization

Docker Integration for Spring Boot Applications

- Creating a Dockerfile
- Building and Running a Docker Image
- Pushing and Running Images on hub.docker.com
- Creating a Docker Compose File

Apache Kafka Integration with Spring Boot

- Introduction to Event-Driven Architecture
- Spring Boot - Kafka Integration

Caching with Spring Boot

- @Cacheable, @CacheEvict, @CachePut
- Using Redis with Spring Boot

Introduction to Microservices

- What Microservices Are NOT?

Inter-Service Communication

- Spring Cloud Eureka Server/Client and Feign Client

- Fault Tolerance with Resilience4j
- Circuit Breaker, Bulkhead, and Rate Limiting

Distributed Systems

- API Gateway
- Spring Actuator
- Centralized Logging with OpenTelemetry

Microservices Configurations

- Spring Cloud Config
- Centralized Configuration Management
- Vault Integration

Inter-Service Communication 2

- Microservice Communication with gRPC

Prerequisites

Having taken the Introduction to Spring Boot training or mastering the topics explained in the Introduction to Spring Boot training.