Java Spring Boot Mastery Roadmap (Beginner to Expert)

# 🟢 Stage 1: Spring Boot Basics

* 🔧 Concepts to Learn:
* • What is Spring Boot?
* • Spring Initializr
* • Maven/Gradle basics
* • Project structure of a Spring Boot app
* • Creating REST APIs (GET, POST, PUT, DELETE)
* • Spring Boot Annotations (@RestController, @RequestMapping, @GetMapping, etc.)
* • Configuration files (application.properties / application.yml)
* • Running and testing your first Spring Boot app
* 🧩 Projects to Build:
* • Hello World API
* • Simple Calculator API
* • To-Do List API (in-memory)

# 🟡 Stage 2: Working with Data

* 🔧 Concepts to Learn:
* • Spring Data JPA
* • H2, MySQL, PostgreSQL setup
* • Entity creation and relationships (@OneToMany, @ManyToOne, @ManyToMany)
* • Repositories (JpaRepository, CrudRepository)
* • Custom query methods
* • DTOs and Model Mappers
* • Validation with javax.validation
* 🧩 Projects to Build:
* • User Management System (CRUD)
* • Blog/Post API with relational DB
* • Product Catalog with categories

# 🟠 Stage 3: Intermediate Concepts

* 🔧 Concepts to Learn:
* • Service Layer pattern
* • Exception Handling (ControllerAdvice, custom exceptions)
* • Logging with SLF4J/Logback
* • Using Lombok (@Data, @Builder, @Slf4j)
* • Environment-based configurations (dev, test, prod)
* • Scheduling tasks (@Scheduled)
* • CommandLineRunner & ApplicationRunner
* 🧩 Projects to Build:
* • Error-handling enhanced Blog API
* • Scheduled report generator
* • Multi-environment configuration demo

# 🔵 Stage 4: Advanced Topics

* 🔧 Concepts to Learn:
* • Spring Security (Basic Auth, JWT)
* • Role-based Authorization (RBAC)
* • Filter and Interceptor usage
* • Custom authentication providers
* • Spring Boot with OAuth2 (Google, GitHub login)
* • Integrating Swagger (OpenAPI 3)
* • Unit and Integration testing (JUnit, Mockito)
* 🧩 Projects to Build:
* • JWT Auth Service with RBAC
* • OAuth2 login with Google
* • Secured Admin Dashboard API
* • API Docs with Swagger UI

# 🔴 Stage 5: Microservices & DevOps

* 🔧 Concepts to Learn:
* • Spring Cloud Config (centralized config)
* • Spring Boot Profiles
* • Service Discovery with Eureka
* • API Gateway with Spring Cloud Gateway
* • Load balancing with Ribbon
* • Circuit Breaker with Resilience4j
* • Centralized logging with ELK stack
* • Dockerizing Spring Boot apps
* • Docker Compose for multi-container setup
* 🧩 Projects to Build:
* • Employee Management Microservices (User, Department, API Gateway)
* • Spring Boot + Docker Compose app
* • Centralized config and logging demo

# 🧠 Expert Topics

* 🔧 Concepts to Learn:
* • Reactive Programming (WebFlux)
* • GraphQL with Spring Boot
* • Kafka integration (Producer/Consumer)
* • gRPC with Spring Boot
* • Spring Batch (scheduled data processing)
* • OpenTelemetry and Observability
* • CI/CD with GitHub Actions / Jenkins
* • Security best practices for production
* 🧩 Projects to Build:
* • Kafka Event Stream Microservices
* • GraphQL API for product catalog
* • Reactive Chat App with WebFlux
* • CI/CD pipeline for Spring Boot app