# Spring Boot Developer – 100+ Lab Exercises (Basic, Intermediate, Advanced)

---

## ☐ Basic Level (30+ Exercises)

### ☐ Getting Started with Spring Boot

- Lab 01: Setup Spring Boot project using Spring Initializr.
- Lab 02: Create a simple RESTful "Hello World" API.
- Lab 03: Understand Spring Boot application structure.
- Lab 04: Run Spring Boot app using embedded Tomcat.
- Lab 05: Build and run Spring Boot jar.

### ☐ Spring Boot Fundamentals

- Lab 06: Create Controller and handle HTTP GET requests.
- Lab 07: Handle POST requests with `@RequestBody`.
- Lab 08: Path variables and query parameters.
- Lab 09: Exception handling with `@ControllerAdvice`.
- Lab 10: Custom error response structure.

### ☐ Dependency Injection & Beans

- Lab 11: Create and inject Beans using `@Component` and `@Autowired`.
- Lab 12: Bean lifecycle with `@PostConstruct` and `@PreDestroy`.
- Lab 13: Configuration properties with `@Value`.
- Lab 14: Use `@Configuration` and `@Bean` annotations.
- Lab 15: Profiles and environment-specific properties.

### ☐ Data Access with Spring Data JPA

- Lab 16: Setup H2 in-memory database.
- Lab 17: Define Entity classes.
- Lab 18: Create Repository interfaces.
- Lab 19: Basic CRUD operations.
- Lab 20: Custom queries with `@Query`.

---

# □ Intermediate Level (40+ Exercises)

## □ REST API Advanced Concepts

- Lab 21: Pagination and Sorting with Pageable.
- Lab 22: Filtering data via query params.
- Lab 23: Implement PATCH requests.
- Lab 24: Use DTOs with ModelMapper.
- Lab 25: Implement validation with `@Valid` and `@NotNull`.

## □ Security with Spring Security

- Lab 26: Basic authentication setup.
- Lab 27: Role-based authorization.
- Lab 28: JWT token generation and validation.
- Lab 29: Securing REST APIs.
- Lab 30: Password encoding with BCrypt.

## □ Service Layer and Business Logic

- Lab 31: Create Service classes with `@Service`.
- Lab 32: Transaction management with `@Transactional`.
- Lab 33: Asynchronous services with `@Async`.
- Lab 34: Implement caching with Spring Cache.
- Lab 35: Use AOP for logging.

## □ Database and Persistence

- Lab 36: Migrate from H2 to MySQL/PostgreSQL.
- Lab 37: Relationships (OneToMany, ManyToOne).
- Lab 38: Cascade types and Fetch types.
- Lab 39: Auditing with `@CreatedDate` and `@LastModifiedDate`.
- Lab 40: Database migrations with Flyway.

## □ File Upload & Download

- Lab 41: Upload files with Multipart.
- Lab 42: Serve files via REST API.
- Lab 43: Store files in DB vs filesystem.
- Lab 44: Implement file size limits.
- Lab 45: Validate file types.

# ☐ Advanced Level (30+ Exercises)

## ☐ Microservices Architecture

- Lab 46: Create multiple Spring Boot microservices.
- Lab 47: Service registration with Eureka.
- Lab 48: Client-side load balancing with Ribbon.
- Lab 49: API Gateway with Spring Cloud Gateway.
- Lab 50: Distributed tracing with Sleuth and Zipkin.

## ☐ Messaging & Integration

- Lab 51: Async communication with RabbitMQ.
- Lab 52: Kafka integration.
- Lab 53: REST API integration using WebClient.
- Lab 54: Scheduled tasks with `@Scheduled`.
- Lab 55: Event-driven programming with ApplicationEvents.

## ☐ Monitoring & Health Checks

- Lab 56: Actuator endpoints setup.
- Lab 57: Custom health indicators.
- Lab 58: Metrics and Prometheus integration.
- Lab 59: Log aggregation with ELK stack.
- Lab 60: Centralized configuration with Spring Cloud Config.

## ☐ Testing & CI/CD

- Lab 61: Unit testing with JUnit 5.
- Lab 62: Mocking with Mockito.
- Lab 63: Integration testing with @SpringBootTest.
- Lab 64: Testcontainers for DB testing.
- Lab 65: Setup CI/CD pipeline for Spring Boot app.

## ☐ Cloud Deployment & Scaling

- Lab 66: Dockerize Spring Boot app.
- Lab 67: Deploy to Kubernetes cluster.
- Lab 68: Auto-scaling with Kubernetes Horizontal Pod Autoscaler.
- Lab 69: Deploy on AWS Elastic Beanstalk.
- Lab 70: Use AWS RDS for DB.

## ☐☐ Capstone Projects & Use Cases

- Lab 71: Build E-Commerce REST API with user roles and payment simulation.
- Lab 72: Employee Management System with audit logs.
- Lab 73: Social Media Feed microservice architecture.
- Lab 74: Real-time chat service with WebSocket.
- Lab 75: IoT data ingestion backend.

## ☐ Tools & Ecosystem

- IDE: IntelliJ IDEA / Eclipse
- Build Tools: Maven / Gradle
- Database: H2, MySQL, PostgreSQL
- Cloud: AWS / Azure / GCP (optional)
- Docker & Kubernetes
- CI/CD: Jenkins / GitHub Actions