O&B ACADEMY

COURSE SYLLABUS

JAVA-202

Professional Spring Boot With JPA

JAVA-202

Professional Spring Boot with JPA

OVERVIEW

Skill Level	:	Intermediate
Suitable for	:	Experienced developers who would like to know Spring in depth, accelerate development with Spring Boot, and gain a solid understanding of object-relational mapping provided by JPA.
Duration	:	5 Days

Spring is a dependency-injection application framework that accelerates development, making microservices quick and easy.

This 5-day course provides participants with an in-depth coverage of the major features of Spring Framework, Spring Boot, and JPA with Hibernate. This includes configuration, data access, object-relational mapping, REST, AOP, auto-configuration, security, and Spring testing framework to build enterprise and microservices applications.

This is designed for experienced developers who would like to use Spring, accelerate it with Spring Boot, and gain a solid understanding of object-relational mapping provided by JPA.

PREREQUISITES

- JAVA-201 — Professional Java (or equivalent experience/training)

COURSE OUTLINE

Spring configuration using Java Configuration and Annotations

- Overview of Spring
- Creating an application context
- Multiple configuration files
- Bean scope
- External properties
- Profiles
- Spring expression language
- Proxying
- Annotation-based dependency injection
- Good practices for component scanning
- Java @Configuration vs annotation
- @PostConstruct and @PreDestroy
- Stereotypes and meta annotations
- Bean lifecycle

Aspect-oriented programming (AOP) with Spring

- Core AOP Concepts
- Defining Pointcuts
- Implementing Advice

Test Spring-based applications using JUnit 5

- Spring's integration test support
- Profile selection with @ActiveProfiles
- Test data setup with @Sql

Use Spring to access relational databases - JDBC

- Spring's DataAccessException hierarchy
- Implementing caching
- Problems with traditional JDBC
- JdbcTemplate

Use Spring support for transactions

- Transaction management
- Traditional JDBC vs Spring
- Transaction propagation
- Rollback rules
- Transactions and integration testing

Use JPA with Spring and Spring data

- JPA configuration in Spring
- JPA configuration using Spring Boot
- Spring Data JPA dynamic repositories

Simplifying applications with Spring Boot auto-configuration, and starters

- Spring's Introduction to Spring Boot
- Auto-configuration
- Dependencies and "starters"
- Packaging
- External configuration
- Profiles
- Logging

Build a simple MVC application using Spring Boot

- Introduction to Spring MVC
- Controller programming model and method parameters
- Using @RequestMapping annotations
- Accessing request data
- Configuring Spring MVC with Spring Boot
- Spring Boot packaging options, JAR or WAR
- Testing controllers

Implement REST with Spring MVC and RestTemplate

- Introduction to REST-style architecture
- Map requests based on HTTP method
- Access request data and set response data
- Building responses explicitly
- Build valid URIs (for `Location` HTTP 201 Created response header)
- Putting it all together

Spring Security

- Security concepts
- Spring Security overview
- Security in a Servlet environment
- Intercepting URLs
- Configuring authentication
- Method security

Utilize Spring Boot enhancements for testing

- Spring Boot testing overview
- Integration testing with @SpringBootTest
- Slices to test different layers of the application
- Testing controllers with @WebMvcTest
- Testing Spring Data JPA queries with @DataJpaTest

Use JPA to map Java objects to relational data

- Getting started
- Accessing entity state
- Mapping simple types
- @Enumerated, @Temporal, @Transient
- Mapping primary Keys: @Id, @GeneratedValue

Use JPA to persist, merge, and remove entities

- Persisting, removing, updating entities
- Detaching and merging entities

Using queries

- JPQL, criteria API, native SQL

Mapping components

- Embeddable classes and collection of embeddable classes and basic types
- Composite keys

Defining customer user-types for value objects

- Hibernate UserType
- JPA AttributeConverter

Inheritance: considerations and trade-offs

Mapping relationships

- @ManyToOne
- @OneToMany
- @ManyToMany
- @OneToOne

Use JPA to map Java objects to relational data

- Cascading operations and orphanRemoval
- Eager vs lazy fetching
- LazyInitializationException
- Entity graphs

Optimistic and pessimistic locking

Callback methods and entity listeners

Software and Hardware Requirements

- Spring Tool Suite
- Java
- Maven

O&B ACADEMY

Engineering for the Real World

Enquiries

+63 2 5322 2307

training-sales@orangeandbronze.com