Document Generated: 12/31/2025

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 4 Days

Mastering Spring 5.x Developer Boot Camp (TT3335)

About This Course:

Mastering Spring and Spring Boot is a hands-on Spring training course geared for experienced Java developers who need to understand what the Spring Framework is in terms of today's systems and architectures, and how to use Spring in conjunction with other technologies and frameworks. This leading-edge course provides added coverage of Spring's Aspect-Oriented Programming and the use of Spring Boot. Students will gain hands-on experience working with Spring, using Maven for project and dependency management.

The Spring framework is an application framework that provides a lightweight container that supports the creation of simple-to-complex components in a non-invasive fashion. Spring's flexibility and transparency is congruent and supportive of incremental development and testing. The framework's structure supports the layering of functionality such as persistence, transactions, view-oriented frameworks, and enterprise systems and capabilities. This course targets Spring 6.x and Spring Boot 3.x, which includes full support for Java SE 17 and Java SE 21.

Spring makes enterprise development easier. Spring simplifies common tasks and encourages good design based on programming to interfaces. Spring makes your application easier to configure and reduces the need for many JEE design patterns. Spring puts the OO design back into your JEE application, and it integrates nicely with many view technologies and the new features of HTML5.

Course Objectives:

This 'skills-centric' course is about 50% hands-on lab and 50% lecture, designed to train attendees in Java and Spring development skills, coupling the most current, effective techniques with the soundest industry practices. Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will learn:

- Explain the issues associated with complex frameworks such as JEE and how Spring addresses those issues
- Understand the relationships between Spring and JEE, AOP, IOC and ORM.
- Write applications that take advantage of the Spring container and the declarative nature of assembling simple components into applications.
- Understand how to configure the Spring Boot framework
- Understand and work on integrating persistence into a Spring application
- Work with Spring Boot to facilitate Spring setup and configuration
- Apply Aspect Oriented Programming (AOP) to Spring applications
- Use Spring Data JPA to implement the persistence layer of an application
- Develop REST applications using Spring 5
- Work with Spring Boot to facilitate Spring setup and configuration

Audience:

- This introduction to Spring development course requires that incoming students possess intermediate-level Java programming skills and practical hands-on Java experience. This class is geared for experienced Java developers who are new to Spring, who wish to understand how and when to use Spring in Java and JEE applications.

Prerequisites:

- This introduction to Spring development course requires that incoming students possess intermediate-level Java programming skills and practical hands-on Java experience. This class is geared for experienced Java developers who are new to Spring, who wish to understand how and when to use Spring in Java and JEE applications.

Course Outline:

1. The Spring Framework

- Understand the value of Spring
- Explore Dependency Injection (DI) and Inversion of Control (IoC)
- Introduce different ways of configuring collaborators
- Spring as an Object Factory
- Initializing the Spring IoC Container

2. Configuring Spring Managed Beans

- Introduce Java-based configuration
- The @Configuration and @Bean annotations
- Define bean dependencies
- Bootstrapping Java Config
- Context Injection in Configuration classes
- Using context Profiles
- Conditionally loading beans and configurations
- Bean Life-Cycle Methods
- Lab: Spring Java Config

3. Defining Bean dependencies

- Introduce Spring annotations for defining dependencies
- Explore the @Autowired annotation
- Stereotype Annotations
- Qualifying injection points
- Lifecycle annotations

- Using properties in Java based configuration
- The @Value annotation
- Using the Candidate Components Index
- Lab: Configuring Bean Dependencies using Annotations
- Lab: Creating the Candidate Component Index (optional)

4. Introduction to Spring Boot

- Introduce the basics of Spring Boot
- Explain auto-configuration
- Introduce the Spring Initializr application
- Bootstrapping a Spring Boot application
- Lab: Introduction to Spring using Spring Boot

5. Working with Spring Boot

- Provide an overview of Spring Boot
- Introduce starter dependencies
- Introduce auto-configuration
- @Enable... annotations
- Conditional configuration
- Spring Boot Externalized Configuration
- Bootstrapping Spring Boot
- Lab: Create REST Repository using Spring Boot

6. Lesson: Introduction to Aspect Oriented Programming

- Aspect Oriented Programming
- Cross Cutting Concerns

7. Lesson: Spring AOP

- Spring AOP in a Nutshell
- @AspectJ support
- Spring AOP advice types
- AspectJ pointcut designators
- Lab: Spring AOP: Adding Interceptors

8. Spring Data Overview

- Spring Data Capabilities and Features
- Spring Data repositories
- The Repository interfaces
- Defining the JPA entity
- Persisting entities using Spring Data JPA
- Bootstrapping the Spring Data application
- Lab: Spring Data JPA Using Spring Boot
- Lab: Spring Data JPA Using Spring Boot (Part 2)
- Lab: Spring Data JPA (Without Spring Boot) (optional)

9. Spring Data Query Methods

- Querying data using Query methods
- Query builder mechanism
- Handling an Absence of Value
- Pagination and Ordering
- Asynchronous query methods
- Count and Delete Derived Query methods
- Lab: Spring Data Query Methods

10. Spring Data JPA Queries

- JPA named queries
- @Query and @NamedQuery annotations
- Defining Query parameters
- Executing native queries
- SpEL expressions in queries
- Managing the Persistence Context after updates
- Lab: Spring Data JPA Queries

11. REST principles

- Introduce the six architectural constraints of REST
- Introduce Resources and Resource representations
- Best practices for defining Resource URIs

12. Introduction to RESTful Services in Spring

- Discuss the request-response cycle of REST requests
- Defining a REST Controller in Spring
- Explain the @ResponseBody annotation
- Define request mappings
- Use path variables
- Lab: Working with Spring REST

13. Introduction to REST Clients in Spring

- Introduce RestTemplate class
- Making GET, POST, PUT, HEAD, OPTIONS and DELETE requests
- Introduce the UriTemplate class
- Using HttpEntity and RequestEntity
- Use the exchange method to define ‘complex’ requests
- Process requests and responses using callback
- Configure the RestTemplate
- Lab: Implementing the Spring REST Client

14. Bootstrapping the REST application

- Describe steps needed to bootstrap Spring REST application
- Configure Content Representation libraries

- Configure Spring MVC and map the Dispatcher Servlet
- Explain the advantages of using Spring Boot to setup the REST project
- Setup a Spring REST application using Spring Boot
- Lab: Spring Hotel Reservation
- Lab: Spring Hotel Reservation Client

15. Content Representation

- Returning different media types from service
- Introduce negotiated resource representation
- Configure Message Converters
- Lab: Spring REST Content Negotiation

16. Implementing the REST Service

- Process for Spring REST Implementation
- The Domain object
- Using Project Lombok to define the domain object
- (Not) Using Data Transfer Objects
- ResponseEntity builder interfaces
- Setting Location header using UriComponentsBuilder
- Lab: Spring REST Services

17. Error Handling

- Handling Exceptions and HTTP Status codes
- Using @ResponseStatus
- Exception handler methods in controller
- Using Controller advices to handle exception
- The ResponseStatusException class
- Lab: Spring REST Exceptions

Additional Topic: Time Permitting

These topics will be included in your course materials but may or may not be presented during the live class depending on the pace of the course and attendee skill level and participation.

18. Spring Boot Actuator

- Understand Spring Boot Actuators
- Work with predefined Actuator endpoints
- Enabling Actuator endpoints
- Securing the Actuator
- Lab: The Spring Actuator
- Lab: Secure the Spring Boot Actuator

19. Developing in Spring Boot

- Introduce Spring Boot Devtools
- Enable the ConditionEvaluationReport
- Debugging Spring Boot applications
- Lab: Spring DevTools