

## **Mastering Spring 4.3 (TT3330-S4)**

**Modality: Virtual Classroom**

**Duration: 5 Days**

**SATV Value:**

**CLC:**

**NATU:**

**SUBSCRIPTION: No**

### **About this course:**

Mastering the Spring 4.x Framework is a five-day 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, as well as Spring security. It includes complete coverage of all new features in Spring 4.x. Students will gain hands-on experience working with Spring, using Maven for project and dependency management, and, optionally, a test-driven approach (using JUnit) to the labs in the course.

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 4.x, which includes full support for Java 8 and JEE 7 (earlier versions of Java and JEE continue to be supported). Spring supports the use of lambda expressions and method references in many of its APIs.

Spring makes JEE 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.

The average salary of a Java Spring Developer is **\$117,087** per year.

### **Course Objective:**

This course provides a solid understanding of what Spring brings to the table and how to use Spring in the context of other technologies and frameworks.

Students are taken on an in-depth tour of the basic Spring framework, initially examining concepts such as Inversion of Control and Dependency Injection, and then working with the container and basic components. They will examine the improved Spring 4.x configuration management options

including XML, annotations, JavaConfig, and Spring Expression Language. The course then moves into the areas of persistence and transactions, looking at various options. Students will then look at options for handling the view portion of an MVC web architecture.

Working in a dynamic, lab-intensive hands-on coding environment, students will learn to:

- 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, JDBC, Hibernate, JSF, Struts, JMS, and EJBs.
- Write applications that take advantage of the Spring container and the declarative nature of assembling simple components into applications.
- Understand how to configure the framework with XML, annotations and JavaConfig as well as explore the advantages of each option.
- Understand and work on integrating persistence into a Spring application.
- Explain Spring's support for transactions and caching
- Understand and work with various options for integrating view-oriented frameworks for web applications into Spring.
- Work with Spring Boot and JavaConfig to more effectively and efficiently develop Spring applications.
- Work with various Spring Application Events, implementing and registering listeners, and monitoring for transactional event

## **Audience:**

This an intermediate- level Spring 4.x training course, designed for developers who need to understand how and when to use Spring in Java and JEE applications.

## **Prerequisite:**

Attendees should have practical basic Java development experience.

## **Course Outline:**

### **Module 1: Introduction to Spring**

#### **Lesson: The Spring Framework**

- Understand the value of Spring
- Explore Dependency Injection (DI) and Inversion of Control (IoC)
- Configuring collaborators
- Understand built-in property editors
- Tutorial: Setup Eclipse Oxygen for Using Maven

#### **Lesson: Configuring Spring**

- Discover the Spring Container

- Introduce the various ways to configure the Spring context
- Initialize the Spring container
- Accessing beans in the Spring container
- Configure beans using XML
- Resolving bean dependencies using XML
- Exercise: Create a Simple Spring program using XML configuration

### **Lesson: Advanced Configuration**

- Use Spring's special factory beans
- Re-use bean definitions in the configuration file using the "parent" bean concept
- Use property placeholders in the configuration file
- Create custom property editors
- Become familiar with container post-processors
- Exercise: Configuring Dependencies
- Exercise: Advanced XML Configuration

### **Lesson: Annotation based configuration**

- Introduce Spring annotation for defining dependencies
- The @Autowired annotation
- Stereotyping Annotations
- Enabling and filtering the component scan
- Qualifying injection points
- Lifecycle annotations
- The @Value annotation
- Exercise: Create a Simple Spring program using annotations

### **Lesson: Java-based Configuration**

- Introduce Java-based configuration
- The @Configuration and @Bean annotations
- Define bean dependencies
- Define bean scopes
- Bootstrapping Java Config context
- Injection in Configuration classes
- Exercise: Create a Simple Spring program using JavaConfig

### **Lesson: More Java-based configuration**

- Registering BeanFactoryPostProcessorT
- he Environment API
- Using context Profiles
- Conditionally loading beans and configurations
- Using properties in Java based configuration
- The DependsOn and Lazy annotations
- Exercise: Java Based Configuration
- Exercise: Using Profiles

## Module 2: Persistence in Spring

### Lesson: Overview: Persistence in Spring

- DAO Implementation
- Transaction Support
- Spring Support for JCache
- Spring Data: JPA to NoSQL

### Lesson: Spring JDBC

- JdbcDaoSupport - JDBC DAO Implementation
- The jdbcTemplate
- Exception Handling
- Operation Classes
- Exercise: Using Spring JDBC

## Module 3: Spring and the Web

### Lesson: MVC Overview

- The DispatcherServlet
- Workflow of Request Handling
- Using Handler Mappings
- ModelAndView and View
- Controllers
- Spring Form Tags
- Exercise: Using Spring MVC

## Module 4: Advanced Topics

### Lesson: Introduction to Aspect Oriented Programming

- Aspect Oriented Programming
- Cross Cutting Concerns

### Lesson: Spring AOP

- Spring's AOP in a Nutshell
- The Three Technologies of "Weaving"
- Spring Advice Types
- Exercise: Spring AOP - Combined Advice

## Module 5: Spring Boot

## Lesson: Spring IO Platform

- Understand the Spring IO Platform
- Understand the IO Bill of Materials
- Understand the IO Foundation
- Learn how the IO Execution will be leveraged
- Learn how Spring Cloud is used for Platform Coordination

## Lesson: Spring Boot Overview

- What is Spring Boot
- Explore Spring Boot starters
- Examine Spring Boot's AutoConfiguration as well as its command-line interface (CLI)
- Understand the Spring Boot Actuator

## Lesson: Spring Boot Introduction

- Spring Boot JPA Starter
- Examine Spring Boot's AutoConfiguration
- Understand the Spring Conditionals
- Understand Spring Boot DevTools
- Exercise: Create a "REST JPA Respository"

## Lesson: Advanced Spring Boot

- Explore additional Spring Boot starters
- Bootstrapping Spring Boot
- Understand Spring Boot Actuators
- Create and run a Spring Thymeleaf MVC application
- Exercise: Create a "Thymeleaf MVC With JPA Repository"

## Module 6: Spring Security Framework

### Lesson: Enterprise Spring Security

- Spring Security Framework
- Security Interceptors
- Authentication Managers
- Wiring in Encoders and Salts
- Access Decision Managers

### Lesson: Spring Web Security

- Spring Security Transparent to Client
- Standard Set of Filters
- Spring Security Config File
- Securing Java Code

- Securing Java Spring Beans
- Exercise: Using Spring Security (Pt 1)
- Exercise: Using Spring Security (Pt 2)
- Exercise: Using Spring Security (Pt 3)
- Exercise: Using Spring Security (Pt 4)

## **Module 7: Spring JMS**

### **Lesson: JMS Overview (Optional)**

- Java Message Service (JMS)
- The JMS Factory Model
- JMS Queue Architecture
- Topic Architecture
- Messages

### **Lesson: Spring and JMS**

- JmsTemplate
- Callback Methods
- Spring Messaging Module
- Message Converters
- MessagePostProcessor
- Destinations
- Working with @JmsListener
- Exercise: Using JMS with Spring