

Core Spring Quick Start | Spring Essentials, Spring Boot, AOP, Persistence & More

Duration: 2 Day(s)

Course Overview

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 5.x, which includes full support for Java SE 11 and Java EE 8. Spring supports the use of lambda expressions and method references in many of its APIs.

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 enterprise application, and it integrates nicely with many view technologies and the new features of HTML5.

Core Spring Quick Start: Introduction to Spring 5.x and Spring Boot 2.x 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. 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.

Review this course online at <https://www.alta3.com/courses/TT3320>

Objectives

- Explain how Spring addresses complex JEE framework issues.
- Understand and configure Spring Boot for application development.
- Implement persistence and transaction management in Spring applications.
- Apply Aspect Oriented Programming within Spring.

Who Should Attend

- Experienced Java developers
- Enterprise application developers
- Software architects
- Developers transitioning from JEE to Spring

Prerequisites

Take Before: Students should have incoming practical skills aligned with those in the course(s) below, or should have attended the following course(s) as a pre-requisite:

- TT2104: Fast Track to Core Java Programming for OO Developers (C+, C#, etc.)

Course Outline

The Spring Framework

1. Understand the value of Spring

2. Explore Dependency Injection (DI) and Inversion of Control (IoC)
3. Introduce different ways of configuring collaborators
4. Spring as an Object Factory
5. Initializing the Spring IoC Container

Configuring Spring Managed Beans

6. Introduce Java-based configuration
7. The @Configuration and @Bean annotations
8. Define bean dependencies
9. Bootstrapping Java Config
10. Context Injection in Configuration classes
11. Using context Profiles
12. Conditionally loading beans and configurations
13. Bean Life-Cycle Methods

Defining Bean Dependencies

14. Introduce Spring annotations for defining dependencies
15. Explore the @Autowired annotation
16. Stereotype Annotations
17. Qualifying injection points
18. Lifecycle annotations
19. Using properties in Java based configuration
20. The @Value annotation
21. Using the Candidate Components Index

Introduction to Spring Boot

22. Introduce the basics of Spring Boot
23. Explain auto-configuration
24. Introduce the Spring Initializr application
25. Bootstrapping a Spring Boot application

Working with Spring Boot

26. Provide an overview of Spring Boot
27. Introduce starter dependencies
28. Introduce auto-configuration
29. @Enable... annotations
30. Conditional configuration
31. Spring Boot Externalized Configuration
32. Bootstrapping Spring Boot

Introduction to Aspect Oriented Programming

- 33. Aspect Oriented Programming
- 34. Cross Cutting Concerns

Spring AOP

- 35. Spring AOP in a Nutshell
- 36. @AspectJ support
- 37. Spring AOP advice types
- 38. AspectJ pointcut designators

Spring Data Overview

- 39. Spring Data Capabilities and Features
- 40. Spring Data repositories
- 41. The Repository interfaces
- 42. Defining the JPA entity
- 43. Persisting entities using Spring Data JPA
- 44. Bootstrapping the Spring Data application

Spring Data Query Methods

- 45. Querying data using Query methods
- 46. Query builder mechanism
- 47. Handling an Absence of Value
- 48. Pagination and Ordering
- 49. Asynchronous query methods
- 50. Count and Delete Derived Query methods