



## Introduction to Spring 3.0 (3 days)

---

*Introduction to Spring 3.0* is a three-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. It includes complete coverage of all new features in Spring 3.0.

---

### ► 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.

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 JSF.

Note that our Spring training covers the entire spectrum and is highly modularized. As such, we can customize courses to your specific needs. The following is a high-level listing of Spring topics to consider in building your customized Spring training:

- Core Spring Framework (including Inversion of Control, Dependency Injection, Spring Expression Language and Aspect-Oriented Programming)
- Spring and Persistence (including Spring DAOs, Transactions, and working with JDBC)
- Spring Views and Rich Interfaces working with Spring MVC
- SpringSource Tool Suite (covers tooling as well as use of the SpringSource's tc Server)

### ► Course Objectives: What You'll Learn

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. The improved Spring 3.0 configuration management options centered on the Spring Expression Language (SPeL) is covered. The course then moves into the areas of persistence and transactions, looking at various options including both JDBC and Hibernate. 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.
- Discuss the challenges to adopting Spring
- Write applications that take advantage of the Spring container and the declarative nature of assembling simple components into applications.
- Understand what the Spring Expression Language brings to the table and how to use it.
- Understand and work with various options for integrating persistence into a Spring application including the recently incorporated Object XML Mapping (OXM) functionality.
- Work with Spring's support for transactions
- Understand and work with various options for integrating view-oriented frameworks for web applications into Spring.
- Understand what the SpringSource Tool Suite (STS) provides in supporting advanced Spring development
- Optionally work with the STS and SpringSource's tc Server

The course provides a solid foundation in basic terminology and concepts, extended and built upon throughout the engagement. Processes and best practices are discussed and illustrated through both discussions and group activities.

Attending students will be led through a series of advanced topics comprised of integrated lectures, extensive hands-on lab exercises, group discussions and comprehensive demonstrations. Please see below for additional information about the hands-on lab work.

### ► Audience & Pre-requisites: Who Should Attend

This is an **intermediate-level** Spring 3.0 training course, designed for developers who need to understand how and when to use Spring in Java and JEE applications. Attendees should have practical basic Java development experience.

## ► Workshop Topics Covered / Course Syllabus

### Session: Introduction to Spring 3.0

#### Lesson: The Spring Framework

- Introduction to Spring
- The Benefits of Spring
- Drawbacks of Traditional Java EE Applications
- Goals of the Spring Framework
- The Nature of POJOs
- Spring Architecture
- Key Features of Spring
- The Core Spring Principles
- Configuring Spring
- POJOs and Interfaces – A few Problems
- A Closer Look at POJOs and Interfaces
- Spring is an Object Factory (XML)
- Dependency Injection
- Dependency Injection - Example
- A Complete Example
- A "Hello World" Spring Example
- Hello World Spring Application
- DI and Testing
- Testing with Mocks
- Testing With Mocks UML Diagram
- Spring Architecture
- Spring Jars
- Spring DI Container
- Initializing the Container
- Accessing Beans from the Container
- Configuring Objects (XML style) Spring
- XML Schema and DTD
- Core Structure of the XML File
- XML Bean Configuration
- Defining and Naming Beans
- Typical Bean Creation
- Constructor Dependencies Injected into the Bean
- Constructor Argument Matching
- Distinguishing "ref" and "value"
- Allowed conversions for "value"
- Examples of "ref" and "value"
- More Syntax Variations for "ref" and "value"
- Dependency Injection – Bean Properties
- Dependency Injection – Using Nested Syntax
- Spring has special handling for Collections
- Collection Based Properties
- Initializing to a Null Value
- Advanced Bean Creation Using Factories
- Static Bean Factory
- Method (non-static) Factories
- Factory Bean Factory
- Bean Creation - Singletons
- Managed Bean Lifecycle
- Validation of Bean Dependencies

- Autowiring
- init and destroy Methods
- Other Advanced Features
- Spring 3 Annotations
- @Resource
- @Autowired
- @Component
- @Component specializations
- @Required
- Configuring Dependencies

#### Lesson: Spring Beans and Advanced Configuration

- Overview of Factory Beans
- Using a Factory Bean for XSL Transformations
- Configuring the XSLFactory Bean
- Spring's Pre-built Factory Beans
- Bean Definition Re-use
- Using Property Files from XML Config Files
- Use PropertyPlaceholderConfigurer
- PropertyPlaceholderConfigurer
- Custom Property Editors
- Bean Post-Processors
- Advanced Configuration

#### Session: Persistence in Spring

##### Lesson: Overview of Spring's Data Access

- Overview of Data Access Support
- DAO Implementations
- Transaction Support
- Isolation Level Concepts
- Isolation Level - Constants
- Propagation Behavior
- 3 Approaches to Transaction Management
- Transaction Config – Needed in All Cases
- Programmatic Transaction
- Spring – Declarative Transactions
- The Benefits of Declarative Transactions
- Benefits of Declarative Transactions in Spring
- Two Forms of Declarative Transactions
- Method 1 – Use the Spring Config File
- Spring Config Example
- Spring Config – The Needed Schemas
- Spring Config – tx:advice and aop:config
- The Java Code is Simple
- Configuring the Transaction
- Method 2 – Use the @Transactional Annotation
- Declarative Annotations
- Declarative Inside the Code

##### Lesson: Spring JDBC

- Overview
- Spring's Help In JDBC Functionality
- Obtaining a DataSource
- Easy to Switch Between DataSources
- JdbcDaoSupport - JDBC DAO Implementation
- The jdbcTemplate
- jdbcTemplate – With RowMapper
- jdbcTemplate – with RowCallbackHandler
- JdbcDaoSupport – Complete example
- Sending SQL
- Exception Handling
- Operation Classes
- Using Spring JDBC

#### Session: Spring Views

##### Lesson: SpringMVC

- Overview of Spring MVC
- The DispatcherServlet
- The WebApplicationContext
- Workflow of Request Handling
- Mapping URLs to Controllers
- Using Handler Mappings
- Handler Mapping Examples
- Handler Interceptors
- Review of Architecture
- ModelAndView and View
- The View in ModelAndView
- A Custom View Example
- ViewResolvers
- Controllers (Finally)
- Using the MultiActionController
- Handling Form Requests
- Spring MVC Annotations
- Setting Up SpringMVC for Annotations
- ClassPath Scanning
- The Spring3 MVC namespace
- <mvc:annotation-driven>
- Handling Requests (Annotations)
- Annotation Controller Mapping
- How @RequestMapping Works
- Spring MVC 3 Handler Methods Parameters
- The Form
- The Spring Form tags
- Using a PropertyEditor
- Adding Validation
- Additional Functionality
- Using SpringMVC

#### Session: AOP in Spring

##### Lesson: Spring AOP

- Aspect Oriented Programming - AOP
- What is Aspect Oriented Programming?
- Cross Cutting Concerns
- What is an "Aspect"?
- Aspects provide 3 separate Decoupling!

- Client can Use Either - Polymorphically
- The Structure of the Proxy – version 1
- The Structure of the Proxy – version 2
- Tradeoffs Between Code Generation Styles
- Some AOP Vocabulary
- Cross Cutting Concerns – A 2D View
- More about Cross Cutting Concerns
- Why is AOP so Important ?
- Summary Crosscutting Concerns
- Spring's AOP in a Nutshell
- A Complete AOP "HelloWorld/Before" Example
- The Simplest Advice – Code to run "Before"
- We'd like to have code run "before"!
- The Complete Advice + Config File
- Review of what We've Accomplished
- A Bit about the Method Class
- The Three Technologies of "Weaving"

- The Generated Proxy Object Structure
- The Cost of Using a Proxy
- The Costs/Benefits to Using Aspects/Proxies
- Four Kinds of Advice
- Spring Advice Types
- MethodBeforeAdvice
- AfterReturningAdvice
- ThrowsAdvice
- Spring AOP - Combined Advice

#### **Session: SpringSource Tool Suite (optional)**

#### **Lesson: SpringSource Tool Suite (STS)**

##### **Overview**

- Introduction to STS
- Many Features are From Eclipse
- STS Layout
- Perspectives ....and Projects
- Basic Workbench Concepts

- Views
- Navigator View
- Problems View
- Editors
- Customizing Perspectives
- Projects
- Java Project
- Various Types of Java Editing Assistance
- Debugging Capabilities Are Powerful
- Keyboard Shortcuts
- Refactoring Overview
- Java EE Support
- Spring Support
- Spring Support is Reflected in Preferences
- Support for Creating and Building Config Files

**3328 /h13**