

Mastering the Spring 3.0 Framework (TT3330)

Length: 5 days

Description

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's Aspect-Oriented Programming (AOP) framework enables developers to declaratively apply common features and capabilities across data types in a transparent fashion.

Audience

Experienced Java Developers

Prerequisite

This an **intermediate** level Spring 3 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.

Topics

INTRODUCTION TO THE SPRING 3.0 FRAMEWORK

- Understanding the value of Spring
- Spring Overview
- Goals
- Features
- Inversion of Control
- Dependency Injection (DI)
- DI and Testing
- Spring Application Architectures
- Spring Container
- Managing the Container
- Access to Services and Resources
- Application Contexts
- Beans as Components
- Beans and Factories
- XML Bean Configuration
- Bean Definition and Dependencies
- Autowiring
- Bean Lifecycle
- Factory Beans

- Customization Options
- Post-Processors
- Property Editors
- Data Handling
- Validation using the Spring Validator
- Type Conversion using Converters
- Formatting using Formatters
- Spring Expression Language (SPeL)
- Using SPeL
- Evaluating Expressions
- Language Constructs
- Operators and Assignment
- Variables and Functions

DATA ACCESS

NOTE: Typically select either JDBC or Hibernate coverage

- Data Access Pattern
- Overview of Persistence Layer and Transactions
- Spring DAO
- DAO Implementations

Mastering the Spring 3.0 Framework (TT3330)

- Transaction Overview
- Spring Transactions
- Isolation Levels
- Propagation Behaviors
- Defining Spring Transactions
- Programmatic
- Declarative in Config Files
- Using Annotations and Config Files
- Working with Demarcation
- Managing Spring Transactions
- Spring JDBC
- Spring JDBC Architecture
- Managing DataSources
- JDBC DAO Support
- Working with JDBC Template
- Database Operations
- Handling JDBC Exceptions
- Spring Hibernate
- Benefits of Spring Hibernate
- Spring - Hibernate Architecture
- DAO Implementation
- Transaction Management
- Hibernate Template
- Solving Open Session in View
- XML Interoperability
- Object XML Mapping in Spring
- Marshalling and UnMarshalling
- Implementation Options
- Working with the JAXB Marshaller
- Views and ViewResolvers
- Work Flow
- Validation
- Spring and Struts
- Spring/Struts Architecture
- Integrating Struts into Spring
- Using Spring's ActionSupport
- Using DI with Struts
- Spring and JSF
- Spring/JSF Architecture
- Integrating JSF into Spring
- Resolving Spring bean references from JSF
- Making Spring bean references from JSF

ASPECT-ORIENTED PROGRAMMING

- AOP Benefits and Concepts
- Aspect Defined
- Decoupling Through Aspects
- Code Generation Styles
- Cross-Cutting Concerns
- Spring's AOP Framework
- Advice and Weaving
- Proxies: Cost/Benefit
- Types of Advice
- Interceptor Chain
- Joinpoints and PointCuts
- Advisors
- Working with Proxies
- Annotations and AOP
- Aspects, Advice, and Pointcuts Using Annotation
- Introductions
- Introductions and Annotations

SPRING VIEWS

NOTE: Typically select either MVC, Struts, or Hibernate coverage

- Spring/Web Framework Architecture
- Spring MVC
- Spring MVC Architecture
- Spring MVC Components
- Spring MVC Flow
- Dispatcher and Controllers
- Handlers and Mapping
- Interceptors

SPRING SECURITY FRAMEWORK

- Understand basic security concepts
- Overview of Spring Security Framework
- Integrating Spring Security
- Protection Using Interceptors

Mastering the Spring 3.0 Framework (TT3330)

- Types of Filters
- Secure Web Pages
- Use Multiple Roles – Secure pages based on roles
- Allow login and logout
- Create custom login pages
- Use JSP Tags to conditionally allow content based on roles
- Securing Services based on roles
- Securing Services based on identity

SPRING AND EJBs (OPTIONAL)

- EJB Overview
- Using EJBs in Spring
- Finding the EJBs Using JNDI
- Using Session EJBs from Spring
- Local and Remote EJBs
- Implementing EJBs
- Spring's EJB Implementation Overview
- Spring and Session EJBs
- Spring and MDBs

SPRING AND JMS (OPTIONAL)

- JMS Overview
- Spring/JMS Architecture
- Working with the JMS Template
- Callback Methods
- Receiving Synchronous Messages
- Message Converters
- Working with JMS Destinations

SPRINGSOURCE TOOL SUITE (STS) (OPTIONAL)

- STS Overview
- Spring-Oriented Wizards
- Spring Configuration Editor
- Visual Development Tools
- Interacting with tc Server
- Deploying to tc Server