

Spring Framework for J2EE

Length:

2 days

Description:

The Spring Framework was created to make developing JEE applications easier. This course was also written to make learning how to use Spring easier. The goal of the class is to present the Spring Framework in a way that is most relevant to a JEE developer by providing practical code examples from real-world experiences.

Since Spring is a modular framework, this course will present it in the same way. Not all developers have the same needs. Some may want to learn the Spring Framework from the ground up, while others may want to pick and choose different topics and go at their own pace. The class will present the topics for students learning the tool for the first time as well as a guide for others needing to dig into specific features much deeper.

Audience:

This course is designed for Java Enterprise developers.

Prerequisites:

The following prerequisites are suggested:

- Have an understanding of Java Enterprise development (J2EE).
- Have an understanding of Enterprise JavaBeans.

Course Content:**Chapter 1 - Core Spring**

Springing into action

- What is Spring?
- A Spring jump start
- Understanding dependency injection
- Applying aspect-oriented programming
- Summary

Basic bean wiring

- Containing your beans
- Creating beans
- Injecting into bean properties
- Autowiring
- Controlling bean creation
- Summary

Advanced bean wiring

- Declaring parent and child beans
- Applying method injection
- Injecting non-Spring beans
- Registering custom property editors
- Working with Spring's special beans
- Scripting beans
- Summary

Advising beans

- Introducing AOP
- Creating classic Spring aspects
- Autoproxying

- Declaring pure-POJO aspects
- Injecting AspectJ aspects
- Summary

Chapter 2 - Enterprise Spring

Hitting the database

- Learning Spring's data access philosophy
- Configuring a data source
- Using JDBC with Spring
- Integrating Hibernate with Spring
- Spring and the Java Persistence API
- Spring and iBATIS
- Caching
- Summary

Managing transactions

- Understanding transactions
- Choosing a transaction manager
- Programming transactions in Spring
- Declaring transactions
- Summary

Securing Spring

- Introducing Spring Security
- Authenticating users
- Controlling access
- Securing web applications
- View-layer security
- Securing method invocations
- Summary

Spring and POJO-based remote services

- An overview of Spring remoting
- Working with RMI
- Remoting with Hessian and Burlap
- Using Spring's HttpInvoker
- Spring and web services
- Summary

Building contract-first web services in Spring

- Introducing Spring-WS
- Defining the contract (first!)
- Handling messages with service endpoints
- Wiring it all together
- Consuming Spring-WS web services
- Summary

Spring messaging

- A brief introduction to JMS
- Using JMS with Spring
- Creating message-driven POJOs
- Using message-based RPC
- Summary

Spring and Enterprise JavaBeans

- Wiring EJBs in Spring
- Developing Spring-enabled EJBs (EJB 2.x)

- Spring and EJB3
- Summary

- Accessing enterprise services
 - Wiring objects from JNDI
 - Sending email
 - Scheduling tasks
 - Managing Spring beans with JMX
 - Summary

Chapter 3 - Client-side Spring

- Handling web requests
 - Getting started with Spring MVC
 - Mapping requests to controllers
 - Handling requests with controllers
 - Handling exceptions
 - Summary

- Rendering web views
 - Resolving views
 - Using JSP templates
 - Laying out pages with Tiles
 - Working with JSP alternatives
 - Generating non-HTML output
 - Summary

- Using Spring Web Flow
 - Getting started with Spring Web Flow
 - Laying the flow groundwork
 - Advanced web flow techniques
 - Integrating Spring Web Flow with other frameworks
 - Summary

- Integrating with other web frameworks
 - Using Spring with Struts
 - Working Spring into WebWork 2/Struts 2
 - Integrating Spring with Tapestry
 - Putting a face on Spring with JSF
 - Ajax-enabling applications in Spring with DWR
 - Summary