



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

This course provides students with the knowledge needed to use the Spring Framework and Rational Application Developer v7 (RAD7) to develop flexible, testable and maintainable J2EE enterprise applications. This course covers v2.5 of the Spring Framework.

Objectives:

Upon completion of this course, you should be able to:

- Understand the need for the Spring framework
- Use inversion of control to increase flexibility and testability of applications
- Understand and use aspect-oriented programming to better handle cross-cutting concerns
- Integrate Spring with the Hibernate ORM framework
- Declaratively and programmatically manage transactions in Spring
- Use the Spring MVC web framework to develop flexible web applications
- Use Spring Web Flow to create conversational web applications
- Integrate Spring with the Struts web application framework
- Access enterprise services including JMS, email, job schedulers and JNDI

Topics:

- The Need for Spring
- Inversion of Control
- Wiring Beans
- Database Integration and Transaction Management
- Aspect-Oriented Programming in Spring
- Spring MVC
- Spring Web Flow
- Web Application Alternatives
- Accessing Enterprise Services

Audience:

Software designers, developers and programmers.

Prerequisites:

To succeed fully in this course, students should be able to:

- Understand Servlets and JSPs

Duration:

Five days



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

Chapter 1 - Introduction to the Spring Framework

- What is the Spring Framework?
- Spring Characteristics
- Spring Modules
- Why Spring?
- Problems with EJB
- Spring Philosophies
- Alternatives to Spring
- Simple Example
- What is Inversion of Control?
- IoC Example

Chapter 2 - Spring Dependency Injection

- Spring Containers
- Bean Creation
- Bean Destruction
- Wiring Beans
- Bean Element
- Dependency Injection – Primitives and Strings
- Dependency Injection - References
- Dependency Injection – Inner Beans
- Dependency Injection – Lists, Arrays and Sets
- Dependency Injection – Maps
- Dependency Injection - Properties
- Constructor Injection
- Constructor Injection – Handling Multiple Arguments
- Constructor vs. Setter Injection
- Autowiring
- Autowiring Example
- Autowiring
- Mixing Explicit and Autowiring
- Special Spring Interfaces
- BeanPostProcessor
- BeanFactoryPostProcessor
- PropertyPlaceholderConfigurer
- CustomEditorConfigurer
- CustomEditorConfigurer
- Awareness Interfaces
- Listening For and Publishing Events
- XML Schema-based Configuration
- XML Schema – jndi Example
- XML Schema – util Example

Chapter 3 - Database Integration and Transaction Management



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

- DAO Support in Spring
- DataAccessException
- Using DataSources
- DAO Templates
- DAO Templates and Callbacks
- Using Spring with JDBC
- Writing Data with JdbcTemplate
- Reading Data with JdbcTemplate
- Reading Single-Row Scalar Data
- Spring 2.5 NamedParameterJdbcTemplate
- ORM Tool Support in Spring
- Hibernate
- Hibernate Session Factory
- Spring LocalSessionFactoryBean
- Spring HibernateTemplate
- Spring HibernateCallback
- HibernateTemplate Convenience Methods
- LocalEntityManagerFactoryBean
- Spring JpaTemplate
- Spring JpaCallback
- JpaTemplate Convenience Features
- Transactions
- Transaction Management in Spring
- Spring Transaction Managers
- Programmatic Transactions

Chapter 4 - Introduction to Aspect-Oriented Programming

- What is AOP?
- Cross-Cutting Concerns
- AOP Concepts
- AOP in Spring
- Declarative Transactions
- Transaction Attributes
- Transaction Propagation Behavior
- Transaction Isolation Levels
- Read-Only and Timeouts
- NameMatchTransactionAttributeSource
- Declaring Transactions with Metadata
- Inheriting Transaction Details
- Autoproxying Transactions
- Spring 2.5 XML Syntax for Transaction Advice

Chapter 5 - Aspect-Oriented Programming in Spring

- Types of Advice
- Before Advice



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

- Before Advice Example
- After Returning Advice
- Around Advice
- Around Advice Example
- Throws Advice
- Pointcuts
- Advisors
- NameMatchMethodPointcut
- NameMatchMethodPointcut Example
- Perl5RegexpMethodPointcut
- ControlFlowPointcut
- IntroductionInterceptor
- IntroductionInterceptor Example
- DelegatingIntroductionInterceptor
- IntroductionAdvisor
- IntroductionAdvisor
- BeanNameAutoProxyCreator
- BeanNameAutoProxyCreator Example
- DefaultAdvisorAutoProxyCreator
- DefaultAdvisorAutoProxyCreator Example
- Metadata Autoproxying
- Spring 2.5 AOP
- @AspectJ
- Aspects in @AspectJ
- Pointcuts in @AspectJ
- execution Pointcut
- Composite Pointcuts in @AspectJ
- Before Advice in @AspectJ
- After Returning Advice in @AspectJ
- After Throwing Advice in @AspectJ
- After (Finally) Advice in @AspectJ
- Around Advice in @AspectJ
- Introductions in @AspectJ
- Accessing Parameters in @AspectJ
- Determining Parameter Names in @AspectJ
- Advice Ordering in @AspectJ
- XML Schema-based AOP Support
- Declaring Pointcuts
- Before Advice in XML Syntax
- After Returning Advice in XML Syntax
- Introductions in XML Syntax
- Advisors in XML Syntax

Chapter 6 - Spring MVC

- Spring MVC
- Spring MVC Components



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

- DispatcherServlet
- Context Loaders
- Spring MVC Example
- Mapping Web Requests
- BeanNameUrlHandlerMapping
- CommonsPathMapHandlerMapping
- Using Multiple Handler Mappings
- Controller Hierarchy
- AbstractCommandController
- SimpleFormController
- AbstractWizardFormController
- MultiActionController
- ThrowawayController
- View Resolution
- InternalResourceViewResolver
- BeanNameViewResolver
- XmlViewResolver
- ResourceBundleViewResolver
- Using Multiple View Resolvers
- Binding Views to Models
- Spring 2.5 Form Tags
- form and input Tags
- password and hidden Tags
- checkbox Tag
- checkbox Tag
- radiobutton Tag
- textarea Tag
- select Tag
- option Tag
- options Tag
- errors Tag
- Portlets
- Spring 2.5 Portlet MVC Framework

Chapter 7 - Introduction to Spring Web Flow 2

- What is Spring Web Flow 2?
- Relationship to Spring #and Spring MVC
- Spring Web Flow 2 Modules
- Integration Support
- What is a Flow?
- Book Hotel Flow Example
- Book Hotel Flow Definition
- Benefits of Spring Web Flow 2
- Required Software
- Configuration Steps
- JAR File Instructions



WT1740-5 Developing J2EE Applications using Spring 2.5 and RAD7 5 days

- Configure web.xml
- Spring Web Flow Components
- Sample Spring Web Flow Configuration File
- Summary

Chapter 8 - Defining Flows

- Recap: What is a Flow?
- Recap: Book Hotel Flow Example
- Flow Definition Language
- Skeletal Flow Definition
- States
- View State
- Transitions
- Generating Events
- Generating Events using Submit Button
- Generating Events using Links
- End States
- Actions
- Actions
- When do Actions Occur?
- Putting it all Together
- Introduction to Unified Expression Language
- Unified Expression Language Functionality
- Unified Expression Language Implicit Variables
- Flow Instances
- Flow Variables
- Variable Scopes
- Assigning and Accessing Variables
- Flow Input
- Binding a View to a Model
- Summary

Chapter 9 - Web Application Alternatives

- Velocity Templates
- Configuring Velocity in Spring
- Velocity View Resolution
- Using Velocity Utility Tools
- Accessing Session/Request Attributes in Velocity
- Binding Velocity Views to the Model
- FreeMarker
- Configuring FreeMarker in Spring
- FreeMarker View Resolution
- Accessing Session/Request Attributes in FreeMarker
- Binding FreeMarker Views to the Model
- Jakarta Tiles



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

- Configuring Tiles in Spring
- Tiles View Resolution
- Tiles Component Controller
- Creating Excel Spreadsheets
- Creating PDF Documents
- Creating Other Non-HTML Documents
- Jakarta Struts
- Using Spring with Struts
- Configuring Spring in Struts
- Spring-Aware Struts Actions
- Delegate Requests to Struts Action Beans
- DelegatingRequestProcessor

Chapter 10 - Accessing Enterprise Services

- Remoting in Spring
- Clients of RMI Services
- Exporting RMI Services
- Hessian and Burlap
- Clients of Hessian/Burlap Services
- Exporting Hessian/Burlap Services
- Spring HTTP Invoker
- Clients of HTTP Invoker Services
- Exporting HTTP Invoker Services
- Clients of EJB Services
- Clients of Web Services
- Accessing Objects in JNDI
- Using the Quartz Scheduler
- Configuring a Quartz Job
- Scheduling a Quartz Job
- Starting Quartz Jobs
- Scheduling a Quartz Job Using a Cron Expression
- MethodInvokingJobDetailFactoryBean
- Sending Email
- Message-Oriented Middleware and JMS
- Spring JMS Resources
- JMS Templates and Callbacks
- Sending JMS Messages
- Receiving JMS Messages
- Converting Messages
- Asynchronous Reception of JMS Messages
- Spring 2.5 Dynamic Language Support

Chapter 11 - Java SE 5 in Spring 2.5

- Java SE 5 Use in Spring 2.5
- @AspectJ



WT1740-5
Developing J2EE Applications using Spring 2.5 and RAD7
5 days

- JPA
- @Required
- @Required in Java Code
- Configuration of @Required
- @Transactional
- @Transactional in Java Code
- Configuration of @Transactional
- @Transactional Properties
- @Transactional Propagation
- @Transactional Isolation
- @Transactional Exceptions
- SimpleJdbcTemplate
- Regular JdbcTemplate Use
- Use of SimpleJdbcTemplate
- Summary

w/d0-3