J2EE Application Development with Spring and Hibernate
BSP-1677
Length: 5 days
Price: $2,895.00

Course Description

This course provides students with the knowledge needed to use the Spring Framework and to use the Hibernate Object/Relational Mapping (ORM) framework to implement the persistence layer of J2EE applications to develop flexible, testable and maintainable Java EE enterprise applications. After completing this course, the student should be able to: • Understand the need for the Spring framework • Use inversion of control to increase flexibility and testability of applications • Understand the concepts of object/relational mappings • Create Hibernate mappings • Understand and use aspect-oriented programming to better handle cross-cutting concerns • Integrate Spring with the Hibernate ORM framework • Retrieve and update persistent objects using Hibernate • Use HQL (Hibernate Query Language) and criteria queries • Know when to use database and application transactions • Use the Spring MVC web framework and form tag library to develop flexible web applications • Access enterprise services including JMS, email, job schedulers and JNDI • Use various Spring and Hibernate tools

Course Audience

Software designers, developers and programmers

Course Outline

1. Introduction to the Spring Framework • Topics • 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 2. Spring Dependency Injection • Topics • 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 • Awareness Interfaces • Listening For and Publishing Events • XML Schema-based Configuration • XML Schema jndi Example • XML Schema util Example

**Database Integration and Transaction Management**
- Topics
  - 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 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

4. Introduction to Hibernate
- Topics
  - What is Hibernate?
  - Relational Database Technology
  - Accessing Relational Databases from Java
  - Domain Models
  - Comparing Object and Relational Models
  - Identity Modeling
  - Granularity of Objects and Tables
  - Association Modeling
  - Navigation
  - Inheritance
  - What is Object/Relational Mapping?
  - Advantages of using an ORM Tool
  - Architecture of Hibernate
  - Hibernate API
  - Extension Interfaces
  - Hibernate Configuration
  - Hibernate Mappings
  - Hibernate Properties
  - Hibernate XML Configuration Files
  - Non-Managed and Managed Environments
  - Configuration in a Non-Managed Environment
  - Configuration in a Managed Environment
  - Binding SessionFactory to JNDI
  - A Simple Example
  - Alternatives to Hibernate
  - EJB Entity Beans
  - Java Persistence API
  - Hand-Coded Persistence Using JDBC
  - Object-Oriented Databases • Other ORM Tools

5. Basic Mappings
- Topics
  - Creating the Domain Model
  - Domain Model
  - Domain Objects
  - Hibernate Requirements
  - Object Identity
  - Primary Keys
  - Natural Keys vs. Surrogate Keys
  - Identifier Property
  - Identifier Generators
  - Hibernate Managed Identifiers
  - Mapping Metadata
  - Mapping Properties • Property Access Strategies
  - Derived Properties
  - Preventing Insert/Update
  - Using Database Schemas
  - Quoted SQL Identifiers
  - Specifying Java Packages
  - Entity and Value Types
  - Components • Bi-Directional Components
  - Other Component Details
  - Association Mappings • Many-to-One Association
  - One-to-Many Association
  - Inheritance Mapping Strategies
  - Table per Concrete Class • Table per Subclass • Table per Class Hierarchy

6. Introduction to Aspect-Oriented Programming
- Topics
  - What is AOP?
  - Cross-Cutting Concerns
  - AOP Concepts
  - AOP in Spring
  - Declarative Transactions
  - Transaction Attributes
  - Transaction Management Strategies
  - 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

7. Aspect-Oriented Programming in Spring
- Topics
  - Types of Advice
  - Before Advice
  - Before Advice Example
  - After Returning Advice
  - Around Advice
  - Around Advice Example
  - Throwing Advice
  - Pointcuts • Advisor Policies
  - NameMatchMethodPointcut
  - NameMatchMethodPointcut Example
  - Perl5RegexpMethodPointcut
  - ControlFlowPointcut
  - IntroductionInterceptor
  - IntroductionInterceptor Example
  - DelegatingIntroductionInterceptor
  - IntroductionAdvisor
  - BeanNameAutoProxyCreator
  - BeanNameAutoProxyCreator Example
  - DefaultAdvisorAutoProxyCreator
  - DefaultAdvisorAutoProxyCreator Example
  - Metadata Autoproxying
  - Spring 2.5 AOP
  - Aspects in @AspectJ
  - Pointcuts in @AspectJ
  - preparation 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

Manipulating Persistent Objects • Topics • Object States • Transient Objects • Persistent Objects • Detached Objects • Object Identity • Persisting a Transient Object • Retrieving and Updating a Persistent Object • Deleting a Persistent Object • Updating a Detached Object • Transitive Persistence • Using save-update • Differentiating Between Transient and Detached Objects

Advanced Mappings • Topics • One-to-One Associations • Many-to-Many Associations • Polymorphic Associations • Entity vs. Value Types • Built-in Mapping Types • Using Mapping Types • Custom Mapping Types • Collections of Value Types • Set of Value Types • Bag of Value Types • List of Value Types • Map of Value Types • Collections of Components • Sorting Collections

Hibernate Queries • Topics • HQL Queries • Parameter Binding • Pagination • Named Queries • Polymorphic Queries • Restriction • String Comparison • Logical Operators • Ordering the Results • Join Queries • Implicit Joins • Where Clause Joins • Criteria Queries • Polymorphic Criteria Queries • Restricting Criteria Queries • Comparing Strings in Criteria Queries • Using Logical Operators in Criteria Queries • Ordering Results in Criteria Queries • Using Joins in Criteria Queries • Using Dot Notation to Get Component Properties

Transactions, Concurrency and Performance • Topics • Database Transactions • JDBC and JTA Transactions • Hibernate Transactions • Flushing the Hibernate Session • Isolation Issues • Isolation Levels • Configuring the Isolation Level • Pessimistic Locking • Application Transactions • Application Transaction Isolation • Managed Versioning • Using Existing Properties to Implement Optimistic Locking • Caching • Hibernate Cache Architecture • The First-Level Cache • The Second-Level Cache • Cache Concurrency Strategies • Enabling the Second-Level Cache • Cache Regions • Specifying a Cache Provider • Support for Cache Providers • Explicitly Evicting Objects From the Second-Level Cache • Fetching Strategies • Fetching Strategies in Mappings • Single Point Associations • Fetching Strategies in Mappings • Collections • Fetching Strategies in Mappings • Batches • Explicit Initialization of Lazily Loaded Objects • Fetching Strategies in HQL Queries • Fetching Strategies in Criteria Queries • The N+1 Selects Issue • Report Queries • Projections • Report Queries • Aggregation • Report Queries • Grouping

Spring MVC • Topics • Spring MVC • Spring MVC Components • 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 • radiobutton Tag • textarea Tag • select Tag • option Tag • options Tag • errors Tag • Portlets • Spring 2.5 Portlet MVC Framework

Web Application Alternatives • Topics • 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 • 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

For more information call 614.481.6555 or E-mail info@babsim.com
Beans • DelegatingRequestProcessor 14. **Accessing Enterprise Services** • Topics • 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 15. **Tools and Design Techniques** • Topics • Generating the Database Schema • Generating the Domain Model • Generating the Database Schema • Using Middlegen • Using XDoclet • The Open Session in View Pattern • Using a Natural Key • Using a Composite Key • Dealing with Database Triggers 16. **Spring Web Services (Spring-WS)** • Objectives • What is Spring-WS? • Contract First Vs. Contract Last Development • Spring-WS Features • Spring-WS Modules • Spring-WS Distribution • Spring-WS Components • Spring-WS Artifacts • Request Processing Workflow • Spring-WS Example • Spring-WS Client Example • Summary

**Available Dates**

**Request a Course Date**

For more information call 614.481.6555 or E-mail info@babsim.com