Spring 3.x and Hibernate

This is a demanding and full 5-day course that covers the Spring 3.0 and Hibernate 3 technologies. It includes a focused coverage of the most useful Spring technologies, including the Spring core, Database Access, and Transaction support. The Hibernate material covers all basic areas of Hibernate as well as some advanced topics. The course starts with a fairly comprehensive coverage of the Core features of Spring, including fairly detailed explanations of the motivation behind Spring, Dependency Inversion and Dependency Injection (IoC). It includes coverage of all the basic capabilities, including the various annotation-based configuration options. The data access sections start with coverage of using the Jdbc Template and DaoSupport classes, as well as configuring DataSources. The course then moves on to the basics of Hibernate and mapping classes, then covers Spring / Hibernate integration – mostly using the Hibernate 3 getCurrentSession() support and Dependency Injection of a Session to build Spring-free DAOs. The transaction section is fairly straightforward, and covers the use of Spring's @Transactional annotation and the XML tx namespace configuration as well as integration with Hibernate.

Course Objectives:

- Understand and use Dependency Injection (DI) and the Spring container to manage application object lifecycles and dependencies.
- Program data access code using Spring's Jdbc or Hibernate templates, and create DAOs (Data Access Objects) using it's DAO support.
- Control transaction declaratively with Spring.
- Create applications that use Hibernate to map persistent Java objects to a relational database.
- Use Hibernate versioning and optimistic locking.
- Map collections and associations using Hibernate.
- Create and execute Hibernate queries using HQL and Criteria.
- Be familiar with hibernate annotations, and know the relationship between Hibernate and the Java Persistence API.

Audience: Java developers who need to work with Spring based applications.

Prerequisites: A good working knowledge of basic Java programming, interfaces, and JDBC.

Number of Days: 5 days

1 Introduction to Spring

The Challenge of Enterprise Applications
Shortcomings of Java/Java EE
What is Spring?
The Spring Modules
The Spring Distribution
Spring Introduction
Managing Beans
A Basic Spring Application
Some Bean Classes
Configuration Metadata
Declaring Beans
The Spring Container
Working with Spring
Why Bother?
Some BeanFactory Methods
Dependencies and Dependency Injection
Dependencies Between Objects
Dependency Inversion Principal
Dependency Injection (DI) in Spring
Dependency Injection Configuration

Advantages of Dependency Injection
Dependency Injection Reduces Coupling

2 More about Bean Properties

Working with Properties
Configuring Value Based Properties
Using Value Based Properties
Property Conversions
Constructor Injections
Constructor Argument Resolution
Setter Injection vs. Constructor Injection
Collection Valued Properties
Working with Collections
Configuring <list> and <set> Properties
Configuring Collections of Bean References
Map Valued Properties
java.util.Properties Valued Properties
Additional Capabilities
Factory Methods
Instance Factory Methods
Bean Aliases
Bean Definition Inheritance
Autowiring
Autowiring byType
Pros and Cons of Autowiring
To Autowire or Not to Autowire

3 The Spring Container and API

ApplicationContext
ApplicationContext Interface
ApplicationContext Implementations
Constructors
Using an ApplicationContext
Spring Resource Access
Built-in Resource Implementations
Bean Scope and Lifecycle
Bean Scope
Specifying Bean Scope
Inner Beans
Compound Names
Depends On
Bean Creation Lifecycle
Bean Creation Lifecycle Details
Using the Lifecycle Interfaces for Beans
Bean Destruction Lifecycle
BeanPostProcessor
@PostConstruct and @PreDestroy
Event Handling
MessageSources
Issues with Messages
Resource Bundles
Defining Resource Bundles
Using Resource Bundles and MessageSource
Localization/Internationalization
Paramaterizing Messages
Annotation Driven Configuration
Annotations in Spring
Enabling Spring Annotations
@Component and Auto-Detecting Beans
DI Using @Resource
Complete Declarations Using Annotations
Other Stereotype Annotations
@Resource – Additional Uses
@AutoWired
Qualifiers
Lifecycle Annotations
XML Config – Annotations and Scanning
Annotation Configuration – Pro/Con
A Note on the XML Configuration
A Brief Note on Annotations
Other Capabilities
SpEL – Spring Expression Language
Other SpEL Capabilities
Validation
Using Validation
Configuring Validation
Validation Constraints
Additional Capabilities

4 Database Access with Spring

Issues with JDBC
Problems Using JDBC Directly
Let’s Review Some Simple JDBC Usage
Simple Query on the Database
Problems with the Previous Approach
Spring Support for the DAO Pattern
Spring DAO Support
The Spring Database API
The JdbcTemplate Class
The JdbcDaoSupport Class
DataSources
Spring Jdbc Exception Hierarchy

DAO Based on Spring Classes
Configuring a DataSource
Looking up a DataSource in JNDI
Building a DAO without the Support Class
Queries and Updates
Querying with JdbcTemplate
Mapping Result Rows to Objects
Defining a RowMapper Class
Inserting/Updating
Other Kinds of Query Methods
The SimpleJdbcTemplate
The SimpleJdbcTemplate Class

5 Introduction to Hibernate

Hibernate Overview
The Issues with Persistence Layers
Object-Relational Mapping (ORM) Issues
Issues with JDBC Alone
Hibernate Benefits
Hibernate Environments
Hibernate Architecture
More Detailed Architecture
Using Hibernate
Acquiring Hibernate
Configuring Hibernate
Hibernate.cfg.xml Elements
SessionFactory Configuration
The Configuration Class
The SessionFactory Interface
SessionFactory API
The Session Interface
Sessions and Transactions
Mapping a Simple Class
Persistent Entity Classes
Persistent Classes
The Event Class
The id Property
The Hibernate Mapping File
The <hibernate-mapping> Element
The <class> Element
The EVENTS Table
Mapping the id Property with <id>
More About Primary Keys
Generating the id Value
Mapping Properties with <property>
Hibernate Mapping Types
Common Hibernate Type Mappings
Field Access or Property Access
The Mapping File
Hibernate Sessions
The Session Interface
Retrieving Persistent Objects
Logging
Hibernate.show_sql
Simple Logging Façade for Java-SLF4J
Apache Log4J
Hibernate log4j.properties file
The log4j.properties file
Modifying log4j.properties for Hibernate
Hibernate Logging Categories

6 Spring / Hibernate Integration

Contextual Sessions
Session Propagation
First-Acquiring a SessionFactory Instance
Contextual Session
Using Contextual Sessions
What is the "Current" Context
Contextual Session Scope
Spring/Hibernate Integration
Spring Support for Hibernate
LocalSessionFactoryBean
Spring Configuration of SessionFactory
Spring Free DAO
HibernateDAOSupport
Querying with HibernateTemplate

7 Updates and Queries

Inserting and Updating
Inserting Instances
Modifying a Persistent Instance
Deleting an Instance
Querying and Hibernate Query Language (HQL)
HQL Basics
Executing a Query
Other Common Query Methods
Where Clause / Restriction
HQL Operators and Expressions
Query Parameters
Using Query Parameters
Named Queries

Projection Queries
Aggregate Functions

8 **Transactions**

Hibernate Transactions
Transaction Lifecycle
Hibernate and Transactions
Hibernate Transaction Demarcation
Working with Transactions
The Hibernate Transaction API
Working in a Managed Environment
Sprig Transaction Management
Transaction Managers
Configuring Transaction Managers
Spring Transactions and Hibernate
JTA Transaction Manager
Spring Declarative Transaction Management
Transactional Scope
Transaction Attributes for Propagation
MANDATORY
NESTED
NEVER
NOT_SUPPORTED
REQUIRED
REQUIRES_NEW
SUPPORTS
Transaction Attributes
Rolling Back and Exceptions
Spring Proxies and Direct Invocation
[Optional] Spring Transactions - XML Configuration

9 **Lifecycle**

The Persistence Lifecycle
Hibernate Object States
Transient and Persistent State
Detached and Removed State
Hibernate Object States and Transitions
The Persistence Context
The Persistence Context as Cache
Synchronization to the Database
Flushing the Session
Persistence Context and Object Identity
Yes, It's Complicated
Versioning and Optimistic Locking
Using a Detached Instance
Optimistic Locking and Versioning
Version Property in Java Class
Version Element in Mapping File
Automatic Version Maintenance
Updating a Detached Instance
Session.saveOrUpdate()
The unsaved-value Attribute
Locking Objects
Common Lock Modes

10 **Relationships**

Object Relationships
Characteristics of Relationships
Directionality
Collections of Value Objects
Modeling a Set of Values
Mapping the Set of Values
Using a Set of Values
More on the Java Collection Type
Using the Java Collection Types
Modeling a List of Values
Mapping a List of Values
Sorted and Ordered Collections
Collections of Components
Mapping Collections of Components
Mapping Entity Relationships
Inheritance
Entity Inheritance
Details of Entity Inheritance
Single-Table Strategy
Class Definitions for Single-Table
Mapping for Single-Table
Sample Table Entries
Single-Table: Pros and Cons
Table per Subclass (Joined Subclass)
Mapping for Table per Subclass
Joined: Pros and Cons
Table per Concrete Class

11 **Spring and the Web**

Spring and Java EE
Java EE Web Applications
Web Application Structure
Web Application Components
ApplicationContext and Web Apps
Configuring ContextLoaderListener
Using the Application Context

12 Additional Hibernate Topics
equals() and hashCode()
Defining equals() and hashCode()
Redefining equals()
Caching
Second-Level Cache
Data Appropriate for Caching
Cache Providers
Configuring Caching
Concurrency Strategies
Managing the Caches
Design Considerations
Long Conversations
Session-Per-Conversation
Problems with Web Applications
Open Session In View Pattern
Query Efficiency Techniques
Beware of n+1 Select Issue
Prefetching Data in Batches
Data Access Object (DAO)
13 Hibernate and the Java Persistence API (JPA)
JPA Persistence API Overview
Java Persistence Environments
Hibernate and JPA
Mapping a Simple Class
Entity Classes
Event Entity Mapped with JPA
javax.persistence.Entity Annotation
The Event Class
The Id Property
Mapping Properties
Basic Mapping Types
Entity Manager and Persistence Context
The Entity Manager & Persistence Context
Persistence Unit
persistence.xml
Acquiring an EntityManager
Working with Transactions
Retrieving Persistent Objects
Inserts and Queries
Persisting a New Entity
Java Persistence Query Language
Executing a Query
WHERE Clause and Query Parameters
Named Queries
Version Property in Java Class
Versioned Class and Detached Objects
Relationships
JPA Support for Relationships
Mapping the Many-To-One Relationship
Mapping the One-To-Many Relationship
Loading and Cascading
Queries Across Relationships
Inheritance
Entity Definitions for Single-Table
Entity Definitions for Joined
Recap
Resources for More Learning