Spring and Hibernate Bootcamp

Course Summary

Description

This Spring and Hibernate training is an intensive five-day course that teaches students how to develop enterprise Java web applications with the Spring, Hibernate, Spring MVC and Spring Security open-source frameworks.

The class is designed to run as a hands-on, tutorial, style with more than 50% of time being devoted to writing code.

The main goal of this course is to set students who are already familiar with Java, Servlets/JSPs (possibly even other frameworks) on the right path of developing enterprise-class web applications on a best-of-breed software stack (Spring and Hibernate) while utilizing time-tested best-practices.

While we don't skip on the theory, students focus most of their energy on why they should use the particular approaches, frameworks, and techniques, and how to best apply them.

This course is based on Spring Framework 4.0, Spring Security 3.2, Hibernate 4.3, and Hibernate Validator 4.3. Additionally, the stack discussed in the class leverages Servlets 3.0, JSP 2.2, JSTL 1.2, SFL4J 1.7, and LogBack 1.1. All of the code is built with Maven 3.

Objectives

By the end of this course, students will be able to:

- Understand the benefits of Inversion of Control (IoC) and Dependency Injection Paradigms (DI) - especially when compared to tranditional J2EE approaches of wiring business objects
- Understand the benefits of object-relational mapping (ORM) frameworks - especially when compared to traditional JDBC-based data persistence approaches
- Be able to use Hibernate (with JPA-annotations) for the persistence layer - including mapping simple and complex entities, configuration, querying, session/transaction management, and tuning techniques
- Be able to use Spring Framework for the Business and DAO layers: including POJO-based development, transactions, wiring, AOP, and testing techniques
- Be able to use Spring MVC to replace Servlets, get support for binding, validation, navigation, error handling, etc.
- Become familiar with Spring WebFlow to model complex user flows in a web application
- Take advantage of Spring Security framework to set up advanced security controls around a Spring Web application - including authentication, authorization, and data transport security (SSL)
- Be able to monitor and tune a Spring/Hibernate based web application
- Learn how to use Eclipse for Java EE to help speed up the development efforts
- Learn how to automate common development tasks with Ant
- Learn best practices, and overall differences between Spring/Hibernate-based applications when compared to other solutions

Topics

- Developing with Java EE
- Hibernate
- Spring Framework
- Spring Security Framework

Spring and Hibernate Bootcamp

Course Summary (cont'd)

Audience

The Spring and Hibernate training course is designed for Java web developers and architects who want to get right to the core of why these frameworks are the best-of-breed in the OOS-Java-community and learn how to effectively take advantage of them in their own applications.

Prerequisites

Participants must be familiar with:

- Java
- The general principles of object oriented programming (OOP)

To get a smooth intro to Spring and Hibernate students should be familiar with:

- Servlets and JSPs
- SQL
- JDBC

Prior exposure to the following concepts and technologies is recommended but not required:

- Familiarity with Java reflection and annotations
- General understanding of web development principles and technologies (HTML, CSS, JavaScript)
- Basic understanding of the HTTP protocol (concepts like SSL, redirection, GET vs POST, cookies)
- Familiarity with MySQL (or any other relational database)
- Familiarity with Eclipse IDE (or any other IDE)
- Familiarity with Tomcat (or any other Java EE application server)
- Familiarity with Ant

Duration

Five days

Spring and Hibernate Bootcamp

Course Outline

I. Developing with Java EE
- A. Review of Java EE (a.k.a. J2EE)
- B. Review of Java Web Application development with JDBC and Servlets/JSPs/EL
- C. Overview of Data Access Object (DAO) design pattern
- D. The pain of developing with JDBC and Servlets

II. Hibernate
- A. Overview of Object-relational-mapping (ORM) frameworks
- B. Overview of Hibernate: advantages, features, etc.
- C. Overview of EJB3's JPA
- D. Switching from JDBC to Hibernate
- E. Installing and configuring Hibernate
- F. Mapping entities with Hibernate Annotations
  1. Mapping associations and collections
     - a) One-to-one, one-to-many, many-to-many
     - b) Understanding directionality (e.g. bi-directional)
  2. Mapping inheritance (understanding different strategies)
  3. Mapping embedded properties
- G. Getting and using Hibernate session
- H. Understanding transactions
- I. Storing entities
- J. Retrieving data
  1. Hibernate Query Language (HQL)
  2. Named queries
  3. Criteria API
  4. Query by example
  5. Filtering data
- K. Understanding entity states: transient, persistent, and detached
- L. Tuning Hibernate
  1. Monitoring
  2. Logging
  3. Lazy loading
  4. Fetching strategies
  5. Caching
- M. Best-practices: examples, and labs

III. Spring Framework
- A. Overview of Spring
  1. Drawbacks of previous approaches
  2. Inversion of control (IoC) with Dependency Injection (DI) design patterns
  3. Advantages of Spring: programming to interfaces, testability, modularity, etc.
  4. Spring framework as a replacement for EJB
  5. Overview of the Spring Container
- B. Wiring beans with Spring
  1. Via XMl
  2. Via Annotations
  3. Auto-wiring
- C. Spring-Hibernate integration
  1. Session management including open-session-in-view (OSIV) strategy
- D. Transaction management
  1. Via Annotations (@Transactional)
  2. Via XML
  3. Understanding transaction propagation
- E. Understanding AOP in Spring
- F. Overview of Model-View-Controller (MVC) design pattern
- G. Overview of Spring MVC
- H. Switching from Servlets to Spring
- I. Installing and configuring Spring MVC
- J. Overview of Spring Controllers
  1. Controller (interface) vs @Controller
  2. Handler (controller) mapping strategies
  3. View resolver strategies
- K. Understating Spring MVC data binding and creating custom binding strategies

Spring and Hibernate Bootcamp

Course Outline (cont'd)

L. Overview of Spring validation with JSR 303 with Hibernate Validator (@Valid, @NotNull, @Size, @NotEmpty, @Email, etc.)
   1. Custom validation annotations and validators

M. Implementing REST on top of Spring MVC

N. Implementing conversations on top of Spring WebFlow

O. Best-practices: examples, and labs

IV. Spring Security Framework

A. Overview of container-managed security

B. Overview of Spring Security framework

C. Switcing from container-managed to Spring security

D. Installing and configuring Spring Security

E. Exposing security context in the application

F. Creating authentication and authorization-aware user interfaces

G. Securing the business layer with JSR 250-style method-based annotations (AOP)