TT488: Core JavaServer Faces (with Spring and Hibernate Overview) Developer's Workshop (5 days)

Created in collaboration by several leading J2EE and JSF authors and industry experts, this intensive course will give you the skills to design and build scalable, secure, maintainable web applications - leveraging Trivera's extensive experience in the delivery of scalable enterprise applications with complex web interfaces based on J2EE technologies.

What You'll Learn

Helping front-end developers, back-end developers, and architects understand how they can get the most out of JavaServer Faces (JSF), this course explores the new official standard for simplifying Java web development, explaining what JSF is, how it works, and how it relates to other frameworks and technologies like Struts, Servlets, JSP, and JSTL.

Also provided is coverage of all the standard components, renderers, converters, and validators, along with advice on how to use them to create solid applications. Working in a hands-on, lab-intensive environment, students will explore and use complex user interface layouts, prototyping, and integrating templates with back-end model objects. Also covered are advanced techniques like internationalization, integration with Struts, and extending JSF with custom components, renderers, converters, and validators.

Workshop Overview

JavaServer Faces (JSF) provides event driven, component –based technology for developing J2EE web applications. This technology greatly simplifies developing web applications. Finally, Java web developers can assemble reusable UI components into rich GUI pages. A standard UI component model will create a rich third party market of reusable GUI components. JSP is to JavaServer Faces as ASP is to ASP.net.

Spring makes J2EE development easier. Spring is a J2EE framework that simplifies commons tasks and encourages good design based on programming to interfaces. Springs makes your application easier to configure and reduces the need for many J2EE design patterns (quite a few J2EE design patterns are really glorified hacks that clutter your code base). Spring puts the OO design back into your J2EE application, and it integrates nicely with JSF.

Hibernate has become the de facto OR (object relational) mapping framework. More people use Hibernate than JDO and EJB CMP-CMR combined.

Combining JSF, Spring and Hibernate is a powerful web development stack. **This course provides comprehensive hands-on content for JSF, and a basic introduction to Spring and Hibernate.**

Audience

This is an intermediate programming course geared for experienced Java developers who want to use and leverage JSF, Spring and Hibernate to build robust web applications.

Students should have at least six months prior Java development experience, including some experience using Server Side technologies (Servlets/JSPs) Students should be comfortable creating a servlet and JSP.

Optional or Alternative Courses

Students with the appropriate background for this training course may also be interested alternative courses that cover Core JavaServer Faces AND additional relate technologies:

TT485: Core JavaServer Faces – 3 days
TT489: Fast Track to Mastering Struts and Core JavaServer Faces – 5 days

TT487 Mastering J2EE Application Development using JSF, Spring and Hibernate – 5 days

Workshop Overview: Hands-On Learning

Throughout the course students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review.

This class is "technology-centric", designed to train attendees in essential J2EE development skills

coupling the most current, effective techniques with the most sound coding practices.

This workshop is about 50% hands-on lab and 50% lecture. Multiple detailed lab exercises are laced throughout the course, designed to reinforce fundamental skills and concepts learned in the lessons. Because these lessons, labs and projects are presented in a building block fashion, students will gain a solid understanding of not only the core concepts, but also how all the pieces fit together in a complete application. At the end of each lesson, developers will be tested with a set of review questions to ensure that he/she has fully understands that topic.

Tools for this Course

Although this training is skills-centric, this course may run using a variety of IDE and application server combinations, including but not limited to: IBM WebSphere Studio Application Developer (WSAD 512); IBM Rational Application Developer (RAD 6.0); Oracle JDeveloper 10g; Apache TomCat; Eclipse / MyEclipse; and others.

Our lab guides are complete with software-specific instructions, screen shots and detailed tutorials for using the software you select. Please contact Training@triveratech.com for additional details, or to make your software selection for this course. In most cases we can easily port our classes to run in the environment of your choosing.

Optional Pre-Testing & Assessment

We work with you to ensure that your resources are well spent. Through our basic pre-testing, we ensure your team is up to the challenges that this course offers. We will work with you to come up with the best solution to ensure your needs are met, whether we customize the material, or devise a different educational path to prepare for this course.

Student Materials

Student Materials include a comprehensive Student Guide complete with detailed course notes, diagrams and a copy of the presentation. Step-by-step lab instructions are clearly illustrated for maximum learning. TriveraTech students also receive a Student Developer Resource CD, complete with workshop labs and solutions; non-restricted workshop software, pertinent javadocs, technical education papers, specifications and freeware.

We're pleased to provide a detailed set up guide for all private or on-site courses, and as much assistance as you require to prepare your students or classroom for the course, to the best of our ability. Our instructors can be contacted for any advice you may require to prepare your classroom.

Workshop Topics Covered

Session 1: Introduction to JavaServer Faces
- Introduction to JavaServer Faces
- What is JavaServer Faces?
- Benefits of JSF
- JSF Development Roles
- Where Does JSF Fit?
- JSF vs. Struts
- System Requirements
- Concerns With JSF
- JSF Framework Structure

Session 2: JSF Architecture Overview
- JSF Architecture Overview
- Physical Components
- How Does JSF Work?
- The FacesServlet
- The Lifecycle Object
- Lifecycle Phases
- Writing a JSF Application
- A Sample JSF Application
- Structure of a Web Application

Session 3: Request Processing
- Request Processing
- Page Navigation
- JSF Objects
- The FacesContext Object
- The UIViewRoot Object
- The UIComponent Interface
- The ExternalContext Object
- The Application Object
- Accessing Backing Beans

Session 4: Simple JSF User Interface Components
- Simple JSF User Interface Components
- JSF Custom Tag Libraries

- The HTML Tag Library
- The <h:commandButton> Tag
- The <h:commandLink> Tag
- The <h:inputText> Tag
- The <h:outputText> Tag
- The <h:outputLabel> Tag
- The <h:messages> Tag
- The <h:selectBooleanCheckbox> Tag
- Panels

Session 5: JSP 2.0 EL Expression Language
- JSP 2.0 EL Expression Language
- What is EL?
- EL Basics
- EL Identifiers
- EL Implicit Objects
- EL Operators
- EL Notes
- JSTL
- The <c:out> tag
- The <c:set> Tag
- The <c:remove> Tag
- The <c:catch> Tag
- JSTL Logic Tags
- The <c:if> Tag
- The <c:choose> Tag
- The <c:forEach> Tag

Session 6: Event Handling
- Event Handling
- The Java 2 Event Model
- The JSF Event Model
- The ActionEvent Class
- The ValueChangeEvent Class
- Event Listeners in JSF
- The ActionListener
- The ValueChangeListener

Session 7: Data Validation
- Data Validation
- Validation
- Using Standard Validators
- Handling Error Output
- <f:validateLength>
- <f:validateLongRange> & <f:validateDoubleRange>
- Custom & Simple Validators
- Creating the Validator Class
- Creating the Tag Handler
- Registering the Validator Class
- Using the Validator Class

Session 8: Advanced Data Validation
- Advanced Data Validation
- Typical Validator Problems
- The StateHolder Interface
- The saveState() Method
- The restoreState() Method
- The transient Attribute
- Validating Dependent Fields
- The MultiFieldValidator
- The DependentFieldValidator
- The DependentFieldValidatorTag
- Updating the faces-config.xml File
- Using the New Validator
- Implement MultiFieldValidator
- Using the Custom JSP Tag

Session 9: Data Conversion and Rendering
- Data Conversion and Rendering
- Data Conversion vs. Rendering
- Converters & Custom Converters
- Writing the Converter Class
- Renderers
- Renderers in the “Real World”
- Creating the Renderer Class
- Creating the Tag Handler
- The TLD for the Tag Handler
- Registering the Renderer
- Using the Component

Session 10: Custom Graphic Controls
- Custom Graphic Controls
- Dynamic Graphics and the Web
- The <img> Tag
- Example – Creating a Bar Chart
- Creating the ChartServlet
- Update web.xml
- Create the <chartItem> Component
- Creating the ChartItemTag Class
- The UIChartItem Class
- Create the <barChart> Component
- The UIBarChart Class
- The BarChartRenderer Class
- Final Clean Up

Session 11: Spring Overview

Lesson: The Spring Framework
- What is Spring?
- Benefits of Spring
- Spring Architecture
- Spring Basics
- Application Context
- Example using Classpath Application Context
- Configuring a Bean

- Defining simple beans
- Accessing beans from application context
- Configuring Collaborators
- Configuration Properties (non collaborators)
- More complex standard properties
- Spring Property Editors
- Properties, Lists, Maps, and Sets are Supported
- Create and Destroy methods
- Three Ways to Wire up Collaborators
- Configuring Collaborators via constructor
- Overview of Aspect-oriented Programming 247
- Join Points
- Pointcuts
- Inter-type declarations
- Creating around advice with an interceptor
- Binding the around interceptor
- Using the AOP class
- Much more to AOP
- Proxy Factories
- JDBC support & template
- JDBC Helper Objects
- Defining base SQL Query Object
- Create concrete Query classes
- Using SQL Update for Deleting
- Using SQL Update for Inserting
- JdbcDAO
- Defining a JdbcDAO class
- JdbcDAO initializing collaborators
- JdbcDAO using collaborator
- Configuring JdbcDAO object
- Configuring JdbcDAO object: passing datasource
- Using AOP to apply Spring transaction support
- Types of Transaction Management
- Configuring a JTA transaction manager

Lesson: Spring Basics
- Spring Basics
- Inversion of Control (IoC)
- Dependency Pull IoC
- Contextualized Dependency Lookup
- Our First Spring Application
- Creating the Business Objects
- Creating the Dependent Objects
- Registering Objects
- Developing Workflow Objects
- Using Contextualized Dependency
- A Better Use of IoC
- The BeanFactory
- The ApplicationContext Interface
- The Spring XML Configuration File
- The <bean> Element
- Injecting Dependencies
- Collections Properties

Lesson: Advanced Spring Beans
- Advanced Spring Beans
- Constructor Dependency Injection
- The Autowire Attribute
- Lifecycle Interfaces
- The InitializingBean Interface
- The DisposableBean Interface
- Knowing Who You Are
- Lifecycle Interfaces of the BeanFactory
- Property Editors
- What is Method Injection?
- Method Injection

Session 12: Hibernate

Lesson: Overview of Hibernate
- Hibernate Overview & Features
- Down to Business
- OR mapping with Hibernate
- Plain old java object no Hibernate
- Mapping file
- Setting up Relationships
- Group contains Users
- User has Contact Info
- User associated with roles
- Employee is a User
- Types of queries in Hibernate

Lesson: DBUnit
- DBUnit Overview
- Testing DAO object that access this
- Using DBUnit (JUnit) Step By Step
- Step 1, Subclass org.dbunit.DatabaseTestCase
- Step 2, Override the setUp method
- Step 3, Create Dataset XML file
- Step 4, getConnection() to provide db connection
- Step 5, getSetUpOperation() and getTearDownOperation()
- Step 6, Define one or more testXXX methods
- Step 7, Release any resources by overriding tearDown()
- Common Pitfalls and Strategy for Integration Testing with Cactus
- DBUnit Ant Support

Lesson: Spring and Hibernate
- Using Queries with Spring/Hibernate
- Spring IOC and Hibernate
- Using named queries
- Problem

- Managing Transactions

Appendix A
- (Short) Introduction to ANT
- What is ANT?
- What are the benefits of ANT?
-
- Installing ANT
- Using ANT
- Writing a build.xml File
- Running an ANT Script

Why Choose This Course?

- This course is wholly owned by TriveraTech, and has been authored and enhanced by experienced JSF and J2EE developers. You may see this outline (and other outlines likes ours) presented by other training vendors – they are typically licensing clients of Trivera Technologies.
- Students will learn how to code, use (and reuse!) essential advanced J2EE skills and concepts properly, using best coding practices, grounding them for advanced J2EE curriculum, and will be prepared for independent study.
- Each lesson has performance driven objectives that ensure students will learn technologies and hands-on skills core to essential web services development – nothing more, nothing less. Progressive labs are designed in such a way that students get a firm grasp on fundamental skills while they work toward building a complete web services application. All lessons have clear objectives, are fundamental to learning core J2EE programming practices, and are reinforced by hands-on code labs and solid practical examples.
- TriveraTech's instructors and course authors are also skilled mentors, J2EE & web services developers and architects. We believe that learning, using and maintaining solid software execution and delivery methods are as important as gaining sharp coding skills. Best Practices for software development and execution, beyond technical coding skills, are enforced throughout all of our courses.

For Additional Information

All courses can be brought onsite for a private presentation, customized to suit your unique requirements or goals. Our team of development experts, architects and mentors are also available to help your team design and deliver your critical development project, while transferring critical skills to your team through our unique **Collaborative Mentoring Programs** and educational services. Work with our experts to set your project off on the right path with our **Project JumpStart Programs**, or have our expert architectural team assess your designs, troubleshoot your work, or make suggestions for time and cost-savings improvements with our **Project Assessment and/or Project Rescue Programs**.

Buy these books! All materials are also available for corporate license with complete instructor support and free corporate branding. We guarantee our pricing and service.

For more information about our services, or to receive a free needs analysis or education roadmap, please contact us at Training@triveratech.com or call 609-953-1515.