

Java EE: Web Development Using Spring 3.0

page 1

Meet the expert: Greg Matus combines his practical programming skills and mentoring excellence to develop the highest quality educational programs available. His technical focus includes developing programs and projects focused around advanced application development, performance, and tuning skills and technologies. His specialties include Java, advanced J2EE / JEE, AJAX, XML, Web Services, JSP, SQL, JDBC and Database topics, Oracle, Eclipse, WebLogic, and more.

Prerequisites: This an intermediate- level Spring 3.0 training course, designed for developers who need to understand how and when to use Spring in Java and JEE applications. You should have practical basic Java development experience.

Runtime: 11:21:25

Course description: The Spring 3.0 framework gives Java developers a strong platform for creating websites and web applications. This course will cover many aspects of web development starting with Direct Web Remoting (DWR). From there the course will cover Spring and Struts which covers Model View Controller. After that the course moves on to WebFlows and testing with Unit Testing and Spring. The later sections of the course will cover the important subject of Security. They will cover Enterprise and Web security and then how to customize web security, securing Java code and user versus role based security.

Course outline:

Direct Web Remoting

- Introduction
- Direct Web Remoting Overview
- DWR Mechanics
- DWR Communication
- Installing DWR
- DWR Basics Non-Spring
- DWR Beans
- DWR Configuration File
- DWR web.xml Registration
- Testing DWR
- DWR Clients
- Spring and DWR Configuration
- DWR Servlet
- Configuring DWRSpringServlet
- Configuring DWR and Spring
- Spring Context File
- Spring DWR Client HTML
- Summary

Direct Web Remoting Demo

- Introduction
- DWR Demo: Setup
- DWR Demo: Editing files
- DWR Demo: PreTest.Java Servlet
- DWR Demo: Add Spring Support
- DWR Demo: Setting up DWR
- DWR Demo: Map Requests
- DWR Demo: HTML

- Summary

DWR Demo Continued

- Introduction
- DWR Demo: HTML Continued
- DWR Demo: Configure Spring
- Summary

Spring and Struts

- Introduction
- Struts Overview
- Struts and "Model 2" (MVC)
- Struts Dispatcher Servlet
- Summary

Spring and Struts Continued

- Introduction
- Struts Action Classes
- Configure Struts with Spring
- Extend Action Support
- Context Loader Listener
- Delegation Request Processor
- DI on Action Classes
- MVC Related Beans
- Summary

Using Struts with Spring

- Introduction
- Demo: Setup Database
- Demo: Create Database
- Demo: Struts Application
- Demo: Adding in Custom Code
- Demo: Configure with Spring
- Summary

More Using Struts with Spring

- Introduction

- Demo: StrutsConfig.xml
- Demo: ApplicationContext.xml
- Demo: Context.xml
- Demo: Define Beans
- Demo: ActionServlet.xml
- Demo: Deploy and Test
- Demo: Debugging
- Summary

Spring Web Flows

- Introduction
- Stateful Web Application
- Stateful Web App Challenges
- Traditional Stateful Web App
- Spring MVC
- Spring MVC Process
- Spring MVC Servlet
- Configure MVC Servlet
- Spring WebFlows
- Process Driven Application
- Shopping Cart Flow Diagram
- Summary

Spring Web Flows Continued

- Introduction
- Spring WebFlow Components
- WebFlow DispatcherServlet
- WebFlow Controller
- WebFlow FlowExecutor
- States
- Variables

- Coding the View
- Triggering Event: Named Button
- Triggering Event: Link
- Subflows
- Shopping Cart Flow Diagram
- Subflow Definition
- Summary

Overview of Unit Testing

- Introduction
- What is Unit Testing
- Purpose of Unit Testing
- Successful Unit Testing
- JUnit Overview
- Reasons to Use JUnit
- How JUnit Works
- Creating a TestCase
- JUnit Methods to Override
- Introducing Class Message
- Demo: Class MessageTest
- Summary

More Overview of Unit Testing

- Introduction
- Creating Class MessageTest
- Demo: Class MessageTest
- Demo: Create Test Method
- Demo: Insert Equals
- Demo: Run Test
- Demo: Recap Order of Events
- Defining a TestSuite

(Continued on page 2)

Java EE: Web Development Using Spring 3.0

page 2

- Example TestSuite
- Summary

Unit Testing and Spring

- Introduction
- Spring Makes Testing Easy
- Spring Specific Tests
- Integration Testing
- Testing SpringMVC Controller
- Testing a Controller's Output
- Request and Response Objects
- Initializing Request Object
- Streamlining Your Assertions
- Model And View Assert Methods
- Rewrite to Use New Assertions
- Summary

More Unit Testing and Spring

- Introduction
- Demo: Overview and Setup Files
- Demo: Test Controllers
- Demo: Parameters
- Demo: Response Object
- Demo: Form Object
- Demo: Command Object
- Demo: Bind Exception Object
- Demo: Direct Controller Call
- Demo: Test Setup
- Summary

Enterprise Spring Security

- Introduction
- The Main Goals of Security
- Core Security Concepts
- Spring Security Framework
- Spring Works With
- Other Spring Features
- Spring Security Transparent
- Security Works by Interception
- Functions and Types
- Authentication Managers
- Performing Authentication
- Providers Supplied by Spring
- Single Sign-on (SSO)
- Authentication Providers
- Protecting Authentication Data
- Username/Password Creation
- Compare Credentials
- Wiring in Encoders and Salts
- Configuring a DAO Provider
- More Authentication Providers
- Access Decision Managers
- Votes and Voters
- Decision Manager Example
- Run-As and After-Invocation

- Other Security with Spring
- Other Attacks on Your Web Site
- Summary

Spring Web Security

- Introduction
- Transparent Spring Security
- Spring Works by Interception
- Spring Security
- Securing a Web Page Steps 1-2
- Add Core Security to JEE Site
- The Standard Set of Filters
- Configure Web.xml
- Spring Beans Security File
- Spring Security Config File
- ApplicationContext-Security.xml
- Spring Security Extensions
- HTTP Section
- Pages Without Access Control
- Authentication Provider
- Running the Program
- Summary

Spring Web Security Demo

- Introduction
- Demo: Deploy Application
- Demo: Inventory Manager
- Demo: Add Basic Security
- Demo: Download Spring Security
- Demo: Add JAR Files
- Demo: Change/Add XML Files
- Demo: Add Filter
- Demo: Configure Basic Security
- Demo: Redeploy Application
- Demo: Test Security
- Demo: Log Off Site
- Summary

Customizing Web Security

- Introduction
- Customizing the Experience
- Protect Different Pages
- Adding a Logout Page
- Adding a Custom Login Page
- Content of the Login Page
- JSP Tags to Show Content
- Summary

Customizing Web Security Demo

- Introduction
- Demo: Overview
- Demo: Split JSP Paths
- Demo: Add Users
- Demo: Modify Index.html
- Demo: Mapping
- Demo: Test Application
- Demo: Add Logout Page

- Demo: Test Logout
- Demo: Custom Login Page
- Demo: Configure Login Page
- Demo: Test/Debug Login
- Demo: JSP Security Tags
- Demo: Test JSP Security
- Summary

Securing Java Code

- Introduction
- Overview
- Security Guidance
- Spring Security with Beans
- Securing Java Spring Beans
- Security Within the Bean
- Summary

Securing Java Code Demo

- Introduction
- Demo: Overview
- Demo: Dispatcher Servlet
- Demo: Protect Service Beans
- Demo: Global Method Security
- Demo: JARs on the Classpath
- Demo: Test Application
- Summary

User Based Web Security

- Introduction
- User Vs. Role Based Security
- User Based Security
- SecurityContextHolder
- Demo: Recap
- Demo: Modify Java Code
- Demo: Add Business Logic
- Demo: Review
- Demo: Test Application
- Summary