Spring Web Flow
Course Summary

Description
This course enables the experienced Java developer to use the Spring Web Flow framework to create simple and complex web applications. Web Flow represents a powerful new approach to designing and developing web applications, and can also draw on the configuration capabilities of Spring itself. We begin with an introduction to the Spring Core module which implements the configurable object container available to Web Flow applications as an application context and get oriented to the Spring web framework, as underpinnings of the Web Flow system.

Then we dive into Web Flow itself, and develop key concepts of flow, state, transition, and action. We get moderately complex applications up and running with these elements, and see how the JavaBeans used to power a flow can also take advantage of Spring configuration and dependency injection. Then we move into more advanced techniques including custom converters, validators, and subflows. We conclude the course with a look at lifecycle and context features, such as the FlowExecutionListener.

Objectives
At the end of this course, students will be able to:
- Understand the scope, purpose, and architecture of Spring
- Use Spring's bean factories and application contexts to declare application components, rather than hard-coding their states and lifecycles
- Use dependency injection to further control object relationships from outside the Java code base
- Use annotations to take advantage of Spring post-processors for automated bean instantiation and wiring
- Create validators for business objects, and associate them for application-level and unit-testing uses
- Build a simple Web Flow application, with declarative control over page flows.
- Bind model beans to pages, and control when binding should and shouldn't occur.
- Invoke Java methods as actions on controllers, service objects, and other JavaBeans placed in Web Flow or Spring singleton scopes.
- Install custom data-binding logic.
- Develop validation logic for individual model beans and view states.
- Organize more complex applications into master flows and subflows.
- Observe the progress of a flow using a flow execution listener, and implement interceptors over one or more lifecycle hooks.

Topics
- Overview of Spring
- The Spring Web Module
- Spring Web Flow

Audience
This course is designed for Java Developers.

Prerequisites
Before attending this course students should have experience with the following:

- Java programming–
- Basic knowledge of XML
- Servlets programming
- JSP

Duration
Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Spring Web Flow
Course Outline

I. Introduction to Spring

A. Overview of Spring
1. Java EE: The Good, The Bad, and the Ugly
2. Enter the Framework
3. Spring Value Proposition
4. The Spring Container
5. Web Applications
6. Persistence Support
7. Aspect-Oriented Programming
8. The Java EE Module(s)
9. Integrating Other Frameworks

B. The Container
1. JavaBeans, Reconsidered
2. The Factory Pattern
3. Inversion of Control
4. XML View: Declaring Beans
5. Java View: Using Beans
6. Singletons and Prototypes

C. Instantiation and Configuration
1. Configuring Through Properties
2. Configuration Namespaces
3. The p: Notation
4. Bean (Configuration) Inheritance
5. Configuring Through Constructors
6. Bean Post-Processors
7. Lifecycle Hooks
8. Integrating Existing Factory Code

II. Dependency Injection

A. Complex Systems
1. Assembling Object Graphs
2. Dependency Injection
3. Single and Multiple Relationships
4. The Utility Schema
5. Bean Aliases
6. Inner Beans
7. Autowiring
8. Auto-Detecting Beans
9. @Autowired Properties
10. Best Practices with Spring 2.5 Annotations

B. Assembling Object Models
1. Collections and Maps
2. Support for Generics
3. The Spring Utility Schema (util:)
4. Autowiring to Multiple Beans
5. Order of Instantiation
6. Bean Factory vs. Application Context

C. Validation
1. Validators
2. The Errors Object
3. ValidationUtils
4. Error Messages and Localization
5. Nested Property Paths

III. The Spring Web Module

A. The Web Module
1. Servlets and JSPs: What's Missing
2. The MVC Pattern
3. The Front Controller Pattern
4. DispatcherServlet
5. A Request/Response Cycle
6. The Strategy Pattern
7. JavaBeans as Web Components
8. Web Application Contexts
9. Handler Mappings
10. "Creating" a Model
11. View Resolvers

B. Customizing Control Flow
1. HandlerMapping Options
2. ViewResolver Options
3. Chaining View Resolvers
4. Triggering Redirects

C. Spring Web Flow
1. Introducing Spring Web Flow
2. Spring Web Flow
3. Relationship to Spring
4. How It Works: Development Time
5. How It Works: Request Time
6. Totally Inverted Control
7. The Web Flow Schema
8. Flows, States, and Transitions
9. Model Beans

D. States and Transitions
1. Flows
2. View States
3. Transitions
4. Lifecycle Hooks
5. Encoding Commands in HTML Views
6. Global Transitions
7. Web Flow URLs

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Spring Web Flow
Course Outline (cont'd)

E. Actions
   1. Variables
   2. Scopes
   3. Actions
   4. Expression Language
   5. Implicit Objects
   6. Using Spring Beans
   7. Action and Decision States
   8. Sequence of Events

F. Beans and Scopes
   1. Model Beans and Binding
   2. Web Flow Scopes
   3. Flow Scope
   4. View Scope
   5. POST-REDIRECT-GET
   6. Flash Scope
   7. Spring Singletons vs. Web Flow Scopes
   8. @Autowired Dependencies

G. Binding
   1. Binding to a Model
   2. The MessageContext
   3. Message Bundles
   4. Reporting Error Messages
   5. Converters
   6. The ConversionService
   7. A Declarative Approach

H. Validation
   1. Validation
   2. The ValidationContext
   3. Registering a Validator
   4. Adapting Spring Validators
   5. The MessageBuilder

I. Subflows
   1. Designing with Web Flow
   2. Flow Input and Output
   3. Subflow States
   4. Conversation Scope
   5. Subflows as Factories
   6. Dynamic Transitions
   7. Action States as Join Points

J. Lifecycle
   1. Observing Flow Execution
   2. The FlowExecutionListener
   3. The RequestContext
   4. The FlowSession
   5. Java View of Flow Definitions
   6. Implementing Interceptors

IV. Appendix A. Spring IDE
   A. Installing Spring IDE
   B. Navigation Features
   C. Auto-Completes
   D. Validation
   E. Support for Spring Web Flow

V. Appendix B. Learning Resources

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically