ComponentSoft
open cloud trainings and consultation

JVA-117A. Spring-MVC Web Applications

Version 4.2

This course enables the experienced Java developer to use the Spring application framework to manage objects in a lightweight, inversion-of-control container, and to build sophisticated web applications using the model/view/controller or MVC framework.

Spring's core module gives the developer declarative control over object creation and assembly; this is useful for any tier of any Java application, so we study it in some depth to begin the course. Then students build web applications that use Spring MVC to organize their designs into coherent request/response cycles. They use Spring command objects to manage HTML forms and their data, and connect these to the validation framework. We connect our applications to persistent stores and study the DAO and ORM modules, to better understand JDBC and JPA persistence models and declarative transaction control. The course concludes with a chapter on Spring's testing framework, including the mock-MVC utilities for web controllers.

Prerequisites

- Java programming – Course 103 is excellent preparation.
- Basic knowledge of XML
- Web development with servlets and JSP – Course 111.
- For the final chapter some undersatnding of JUnit is required

Learning Objectives

- Understand the scope, purpose, and architecture of Spring
- Use Spring 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
- Build a web application as a Spring DispatcherServlet and associated application context
- Use Spring MVC annotations to develop web controllers, mapping request URLs and other criteria to Java methods and binding request data to method parameters
- Build and manage HTML forms with Spring command objects and custom tags
- Customize input binding, validation, and error handling
- Use Spring interceptors to implement horizontal features in the web application
- Connect business objects to persistent stores using Spring's DAO and ORM modules
- Simplify JDBC code using Spring templates
- Integrate JPA entities and DAOs into Spring applications
- Control transactions using Spring, either programmatically or declaratively
- Develop effective unit tests using Spring's test framework and the MockMvc environment for web controllers

IDE Support: Eclipse Luna or higher

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Luna or higher. Students can code, build, deploy, and test all exercises from within Eclipse, and

take advantage of Eclipse's built-in editors and wizards for web applications, XML files, JSPs, and more.

Course Length: 5 days

Course Outline

Chapter 1. Overview of Spring

- Java EE: The Good, The Bad, and the Ugly
- Enter the Framework
- Spring Value Proposition
- The Spring Container
- Web Applications
- Persistence Support
- Aspect-Oriented Programming
- The Java EE Module(s)
- Integrating Other Frameworks

Chapter 2. The Container

- JavaBeans, Reconsidered
- The Factory Pattern
- Inversion of Control
- XML View: Declaring Beans
- Java View: Using Beans
- Singletons and Prototypes

Chapter 3. Instantiation and Configuration

- Configuring Through Properties
- Configuration Namespaces
- The p: Notation
- Bean (Configuration) Inheritance
- Configuring Through Constructors
- Bean Post-Processors
- Lifecycle Hooks
- Integrating Existing Factory Code
- Awareness Interfaces

Chapter 4. Dependency Injection

- Assembling Object Graphs
- Dependency Injection
- Single and Multiple Relationships
- The Utility Schema
- Using Spring Expression Language (SpEL)
- Inner Beans

- Autowiring
- @Component, @Service, & Company
- @Autowired Properties
- Best Practices with Spring Annotations
- Java Classes as @Configurations
- AnnotationConfigApplicationContext
- Capabilities and Limitations
- Mixing and Importing XML and Java Configurations

Chapter 5. Assembling Object Models

- Collections and Maps
- Support for Generics
- The Spring Utility Schema (util:)
- Autowiring to Multiple Beans
- Order of Instantiation
- Bean Factory vs. Application Context

Chapter 6. The Web Module

- Servlets and JSPs: What'sMissing
- The MVC Pattern
- The Front Controller Pattern
- Dispatcher Servlet
- A Request/Response Cycle
- The Strategy Pattern
- Web Application Contexts
- Annotation-Based Handler Mappings
- @Controller and @RequestMapping
- "Creating" a Model
- Views and View Resolvers

Chapter 7. Handling Requests

- Matching URLs
- Identifying Views
- Request Parameters
- Injectable Parameters
- Command Objects
- Return Types
- HTTP Methods
- Path Variables
- Scope and Granularity of Command Objects
- Headers and Cookies
- RESTful Web Services

Chapter 8. Working with Forms

- Form Processing in Spring MVC
- Command Objects in Request Methods
- Spring Custom Tags
- <form:form> and Friends
- Text Fields, Check Boxes, and Buttons
- Radio Buttons and Select/Option Lists
- Command objects at Session Scope
- Limitations of @SessionAttributes

Chapter 9. Data Binding

- A Consolidated Process
- Property Editors
- Custom Property Editors
- DataBinder and @InitBinder Methods
- Converters and Formatters
- Using <mvc:annotation-driven/>
- Custom Formatters

Chapter 10. Validation

- Validating Form Input
- Spring Validators
- Deriving a Validator Reference
- Applying a Validator
- <form:errors>
- Bean Validation a/k/a JSR 303,
- Configuring JSR-303 Support
- Automatic Support with @Valid

Chapter 11. Configuring Spring MVC

- Configuring Message Sources
- Resolving Error Codes
- HandlerExceptionResolver
- @ExceptionHandler
- @ControllerAdvice for Global Error Handling

Chapter 12. Interceptors

- Interceptors
- Configuring Interceptors
- Filters in the Request-Handling Cycle

Chapter 13. Persistence with JDBC

- Reducing Code Complexity
- The DataAccessException Hierarchy
- JdbcTemplate

- RowMapper<T> and ResultSetExtractor<T>
- The DaoSupport Hierarchy
- Capturing Generated Keys
- Transaction Control
- TransactionTemplate
- Isolation Levels
- Transaction Propagation

Chapter 14. Persistence with JPA

- Object/Relational Mapping
- The Java Persistence API
- JpaDaoSupport and JpaTemplate
- @PersistenceUnit and @PersistenceContext
- Shared Entity Managers
- Using <tx:annotation-driven>
- The @Transaction Annotation
- Isolation and Propagation
- A Limitation of @Transactional
- Understanding Entity States
- Bean Validation in JPA
- Optimistic Locking

Chapter 15. Testing

- Testability of Spring Applications
- Dependency Injection
- Mocking
- SpringJUnit4ClassRunner
- TestContext
- @ContextConfiguration
- Preserving Test Isolation
- @DirtiesContext
- Mocking Spring MVC
- Building Requests
- Checking the ModelAndView
- Profiles
- Testing Persistence Components