Fast Track to Spring 4

Over the years, Spring has added many new capabilities, in line with the increasing capabilities of the Java platform. As well as being current with Spring 4, this course introduces techniques for using these powerful capabilities. It includes complete coverage of the three main configuration styles: Java-based (@Configuration), annotation-based (@Component), and the traditional XML-based configuration that may still play an important role in existing and new projects. It also provides guidelines for when and how to use each one.

This 3-day course starts with in-depth coverage on using the powerful capabilities of Spring's Core module to reduce coupling and increase the flexibility, ease of maintenance, and testing of your applications. It goes on to cover many of the most important capabilities of Spring, including integrating persistence layers (e.g. Hibernate/JPA) with Spring, using Spring's powerful Aspect Oriented Programming (AOP) to program cross-cutting concerns in a safe and maintainable way, and using Spring's declarative transaction capabilities. It also covers integration of Spring with Java EE Web applications. This course is hands on with labs to reinforce all the important concepts. It will enable you to build working Spring applications and give you an understanding of the important concepts and technology in a very short time.

Course Objectives:

- Understand the core principles of Spring, and of Dependency Injection (DI)/Inversion of Control.
- Use the Spring Core module and DI to configure and wire application objects (beans) together.
- Know the different types of metadata (XML, annotations/@Component, and Java Configuration/@Configuration), and how and when to use them.
- Understand and use the complete capabilities of the Core module, such as lifecycle events, bean scopes, and the Spring API.
- Work with the ORM (Object-Relational Mapping) module to integrate Spring with technologies such as Hibernate or JPA.
- Understand and use Spring's powerful AOP capabilities for programming cross-cutting concerns across multiple points in an application.
- Learn safe and maintainable techniques for programming with AOP.
- Understand and use Spring’s transaction support, including the easy-to-use Java annotation support as well as the tx/aop XML configuration elements.
- Integrate Spring with Java EE Web applications.

Audience: Java developers who need to work with Spring based applications.

Prerequisites: A good working knowledge of basic Java, JDBC, and Servlets/JSP.

Number of Days: 3 days

1 Introduction to Spring

Overview of Spring Technology
Spring vs. JEE
What is Spring?
The Spring Modules
The Spring Jars
Spring Introduction
Managing Beans

The JavaTunes Online Store
XML Bean Definitions
Declaring Beans
Spring’s XML Schemas
The Spring Container
Working with Spring
Why Bother?
ApplicationContext Interface
Common Application Context
Specifying Configuration Files
A Word About Junit
Dependencies and Dependency Injection
Dependencies Between Objects
Dependency Inversion Principal
Dependency Inversion Illustrated
Dependency Injection Configuration
DI Hides Dependencies
Dependency Injection Reduces Coupling

2 Configuration in Depth

Annotation-based Configuration
Annotations for Spring Configuration
Declaring Beans and DI with Annotations
Complete Declarations Using Annotations
Using @Inject/@Autowired
Enabling Annotations/ Detecting Beans
Wiring Strategies So Far
Java-based Configuration
Using Java-based Configuration
Dependency Injection
How does it Work?
Dependencies in Configuration Classes
Other @Bean Capabilities
Java-based Configuration – Pro/Con
Integrating Configuration Types
Choosing a Configuration Style
Importing between XML/@Configuration
Scanning for @Configuration Classes
Bean Scope and Lifecycle
Specifying Bean Scope
Bean Creation Lifecycle – Advanced Details

3 Wiring in Depth

Value Injection
Configuring Value Properties, Property Conversions
Externalizing Values in Properties Files
Constructor Injection
Configuration - @Configuration and XML
P: and C: namespaces for XML configuration
Qualifiers/Domain Specific Language (DSL)
Limitations of Autowiring
Qualifiers and DSL
Creating and Using an Annotation-Based DSL for Bean Configuration
Benefits of Qualifiers for Bean Configuration
Profiles
Configuring Profiles (XML & @Configuration)
Activating Profiles
Overview of SpEL

4 Database Access with Spring

Overview of Spring database support
Configuring a DataSource
Using Spring with Hibernate
High Level Hibernate Overview
SessionFactory configuration
Contextual Sessions and Spring Integration
Using Spring with JPA
Managing the EntityManager (EM)
JEE and JNDI Lookup of the EM
Configuration and Vendor Adaptors
Creating a JPA Repository/DAO Bean

5 Aspect Oriented Programming

Overview of AOP
Crosscutting Concerns
AOP Basics, Aspect, Joinpoint, Advice, Pointcut
Spring AOP Introduction
Configuration - XML and @AspectJ
Defining an Aspect, Pointcut, and Advice
Pointcut Expressions and Advice
The execution() Designator
Kinds of Advice

Marker Annotations (Rubber Stamp AOP)
Issue with AOP Configuration
Defining an AOP Marker / Rubber Stamp
Advantages of Marker Annotations
@AspectJ Based AOP Support
@AspectJ Annotations Overview
Defining an Aspect, Pointcut, and Advice
Spring AOP Proxies and Self-Invocation Issues
Load-Time Weaving
Caveats of AOP

6 Spring Transaction (TX)
Intro to Spring Transaction Management
Spring Transaction Managers
Spring TX Scope and Propagation
Spring TX Attributes
XML Configuration of Transactions
Specifying Advice, TX Attributes, and Methods
Linking Advice with Pointcuts
Benefits of XML Configuration of TX Behavior

7 Web Applications with Spring
Integrating Spring with Java EE Web Apps
ContextLoaderListener
WebApplicationContext
Using Spring beans in Web app controller logic

8 XML Specific Configuration
Collection Valued Properties – Configuring and using lists, sets, etc.
Additional Capabilities
Factory Classes and Factory Methods
Definition Inheritance (Parent Beans)
AutoWiring with XML
Inner Beans, Compound Names

9 Appendix: Maven and Spring