JAV800 : Java Development with Spring

Code :	JAV800	**Duration :**	4 days	**Category :**	Frameworks

Audience :

Experienced Java developers who want to use the Spring Framework for Java Desktop, Web and Enterprise applications.

Prerequisites :

Experience with Java programming and object orientation is required to participate in this course. Basic knowledge of Web Applications and XML is beneficial for a proper understanding.

Realization :

The concepts are treated by means of presentation slides and demos. The theory is interspersed with exercises. The course material is in English.

Java Development with Spring

Contents :

This course covers the concepts, components and architecture of the Spring Framework. Ample attention is paid to the concept of Dependency Injection, also called Inversion of Control, that plays a central role. It is explained how a loose coupling between objects can be accomplished through an external configuration file that is read by the Spring bean factory of Spring application context. Different varieties of such dependency injection like setter injection and constructor injection are discussed as is the automation of this bean wiring process through autowiring. The other pillar of the Framework, Aspect Orientation, is also addressed. The concepts surrounding Aspect Orientation like Aspects, Joinpoints, Pointcuts, Advice and Weaving are explained. The main point of AOP is that crosscutting concerns in an application, such as logging or security, should be placed in separate parts of the code without polluting the main business logic. Next the different options to store data from Spring Java applications in databases are treated. Attention is paid to using JDBC with a JdbcTemplate, as well as to the use of Object Relational Mapping frameworks like Hibernate with a HibernateTemplate or JPA through annotations. In this respect also Transactions in Spring, the different Transaction Managers and the Transaction Callback API are discussed. Web Applications with the Spring MVC Framework are also part of the course schedule and the role of controllers, views, interceptors and viewresolvers is discussed. Finally, attention is paid to Security, Remoting, Spring and JMS and JMX.

Module 1 : Spring Introduction

What is Spring?
Addressing Layers
Characteristics
Framework Overview
Dependency Injection
Inversion of Control
Aspect Oriented Programming
Portable Service Abstractions
Spring Packages

Module 2 : Dependency Injection

Non-IoC or Dependency Injection
Benefits of Dependency Injection
Constructor Dependency Injection
Setter Dependency Injection
Bean Factory
XmlBeanFactory
Bean Configuration File
Injection Parameter Types
Bean Naming
Autowiring Properties
Application Context
Multiple configuration files
Working with interceptors
Externalizing constant values
Bean scopes

Module 3 : Application Configuration

Bean definition inheritance
Inner beans
p and util namespaces
Dependency injection of collections
Spring Expression Language
Autowiring and component scanning
Stereotype annotations
Java-based configuration
Mixing configuration styles
When to use XML, annotations, and Java configuration
Testing Applications

Module 4 : Aspect Orientation

Aspect Oriented Programming
The need for AOP
Crosscutting Concerns
Aspect
Joinpoints
Pointcuts
Advise
Weaving
Target
Introduction
Spring AOP
Static AOP
Dynamic AOP
Proxies
ProxyFactory

Module 5 : Spring Persistence

Spring and Persistence
Java Persistence
Traditional Persistence
Transparant Persistence
Shared Persistence Concepts
DAO Design Pattern
Before and after DAO
DAO Pattern JDBC
Integration with IoC
DAO Portability
Spring DAO Concepts
Transaction Management
Spring Exceptions
Exception Translation

Module 6 : Spring JDBC

Spring and JDBC
JDBC Characteristics
JDBC Architecture
Executing Statements
JDBC Drivers and URL's
Spring JDBC Data Access
DAO with JdbcTemplate
Data Source Injection
Querying using JdbcTemplate
RowMapper
Querying and Populating Objects
Updating with JdbcTemplate
ResultsetExtractor
Callbacks
SimpleJdbcTemplate
NamedParameterJdbcTemplate
JdbcDaoSupport

Module 7 : Spring ORM

Spring and Hibernate
Hibernate Integration
Mapping Classes
HibernateTemplate
Implementation HibernateTemplate
HibernateTemplate execute
Hibernate DAO Implementation
Hibernate Annotations
Spring and JPA
LocalEntityManagerFactoryBean
Using JPA API
Persistence Unit Configuration
LocalContainerEntityManagerFactoryBean
Persistence Configuration
PersistenceExceptionTranslationProcessor
Container Managed Transactions
Externalizing Database Properties
Entity Manager from JNDI
JpaTemplate and JpaDaoSupport
JPA Java Configuration

Module 8 : Transactions

Transaction Managers
Declaring Transaction Managers
Programmatic Transactions
Transaction Callback API
@Transactional annotation
Declarative Transactions
Isolation Levels
Read-Only Hint
Timeouts
Declaring a Transaction Manager
Configuring transaction propagation
Transactions and integration testing

Module 9 : Spring MVC

Spring in a Web application
What Spring MVC?
Request life-cycle
DispatcherServlet
URL Handler mapping
MVC Configuration
Loading Configuration Files
ContextLoaderListener
Controllers
Defining Views
Spring Tag Library
View Resolvers
Validation
Interceptors
MVC namespace
Exception Handling
Applying Themes
I18N support

Module 10 : Spring and Security

Spring Security Model
Process Behind Security Interceptors
Authentication Manager
Configuring authentication
Intercepting URLs
Security tag library for JSPs
Security at the method level
Customizing the Security filter chain
Access Decision Manager
Security Based on Roles
Security Based on Identity
Run-as Manager
Custom Login Pages
After Invocation Manager
XSD Extensions
Using Annotations

Module 13 : Spring and JMX

What is JMX?
JMX API
Managed Beans
MBean flavors
JMX Architecture
Naming MBeans
MBean Server
Registering Mbeans
Manipulating MBeans
Notifications
Configuring Spring to export MBeans automatically
Exporting a Spring bean as an MBean

Module 11 : Remoting

What is RMI?
Remote Interface
Dynamic Proxies
Marshalling and Serialization
Parameter Passing
Using Spring remoting over RMI
Using the HttpInvoker for remote access over HTTP

Module 12 : Spring and JMS

What is JMS?
Messaging Characteristics
JMS API
Messaging Models
JMS Architectural Components
JMS Message Interfacs
Configuring JMS resources with Spring
Using the JmsTemplate
Message listener containers