

# 90.05.04 Spring

## Duration: 5 days

Tailored Spring Web Development course

This course teaches you exactly what you need to start being productive on Spring projects

### Who can benefit?

This is an intermediate to advanced level Spring course, designed for developers who wish understand Spring Web development features.

The student should be an experienced Java/Java EE professionals, with practical development experience using Servlets/JSP/JSF. Students should be familiar with basic Spring tasks. Students should also be familiar with unit testing and JUnit 4.x

### Detailed description

This course teaches you the fundamental skills to work with Spring. Specialized modules are left out to use the short time for this course as good as possible. The only exception is Spring's data-access capabilities. We consider this a core aspect of Spring as well.

The course concentrates on

- Understanding configuring a Spring application
- Understand Spring's schema based configuration
- Use Spring 2.5 annotations for configuration
- Extending the IoC
- Use stereotype annotations (controller, component etc)
- Spring's AOP support
- Using services (transactions etc)
- Spring's DAO support (JDBC, Transactions etc)
- Spring's ORM support (JPA, Hibernate etc)
- Spring MVC
- Spring MVC JSF integration
- Spring`s unit testing framework

### Workshop overview

This is an overview of the course. For a detailed description with objectives per topic, we refer to the detailed workshop overview, listed further below in this document.

#### Review of the Spring Framework

The Spring Framework and IoC  
The Spring IoC container  
Advanced Spring Container Features

#### Advanced Spring configuration

XML Schema based configuration  
Annotations and Container Extension Points

## Spring AOP Framework

Introduction to Aspect-Oriented Programming  
Spring AOP

## Spring Data Access Support

Overview  
Spring JDBC  
Introduction to Hibernate  
Spring Hibernate

## Web Frameworks

Spring MVC  
Spring and JSF (Optional)

## Java Unit testing techniques

Spring TestContext framework

## Detailed overview

## Review of the Spring Framework<sub>(rev:)</sub>

### The Spring Framework and IoC

- Understand the value of Spring
- Explore IoC/DI
- Configuring collaborators
- Understand built-in property editors

### The Spring IoC container

- Instantiate the IoC container
- Introduce the spring configuration file
- Configure beans within the spring configuration file
- Configure bean properties using the Spring configuration file
- Configure collaborators
- Understand built-in property editors

### Advanced Spring Container Features

- Introduce some of Spring's callback interfaces
- Provide an overview of Spring's factory beans
- Re-use bean definitions in the configuration file
- Create custom property editors
- introduce the `PropertyPlaceholderConfigurer`
- Use placeholders in the configuration file
- explain the use of factory beans
- Add internationalization using `MessageSources`

## Advanced Spring configuration<sub>(rev:)</sub>

### XML Schema based configuration

- Explain XML-Schema based configuration
- Introduce the `util` namespace
- Introduce the `context` namespace



## Annotations and Container Extension Points

- explain the use of `BeanPostProcessors`
- use annotations to define bean dependencies
- become familiar with the `@Autowired` annotation
- use `@Qualifier` and bean meta data
- develop your own qualifier annotation
- understand the use of the JSR-250 `@Resource` annotation
- add lifecycle annotations to the bean
- use stereotype annotations within the bean
- explain the use of the `@Repository`, `@Service` and `@Controller` annotations

## Spring AOP Framework<sub>(rev:)</sub>

### Introduction to Aspect-Oriented Programming

- explain the basics of AOP
- introduce AOP support within Spring

### Spring AOP

- Understanding Spring AOP
- introduce the AOP alliance API
- Explain how to develop interceptors (aspects)
- become familiar with the AspectJ language to define pointcuts
- understand how XML-Schema based AOP definitions are added to the Spring configuration file.

## Spring Data Access Support<sub>(rev:4263)</sub>

### Overview

- Understand the Spring DAO
- Understand Transaction Demarcation within Spring

### Spring JDBC

- Overview of Spring JDBC support
- Defining `DataSources`
- JDBC Exceptions and the `SQLExceptionTranslator` within Spring
- Creating JDBC DAOs
- Using the `JdbcTemplate` class
- Mapping rows to Objects using the `JdbcTemplate`
- Mapping data to Objects using SQL Helper Objects (Operation Classes)

### Introduction to Hibernate

- Define Object to Relational (O/R) Mapping (ORM)
- Define the Hibernate Architecture
- Illustrate the benefits of using Hibernate
- Understand the goals of Hibernate
- Understand key functionalities of Hibernate
- Define the different Hibernate types (*Entities* and *values*)
- Understand different Persistent representations (POJOs, Maps and XML)

### **Spring Hibernate**

- Overview of Spring Hibernate support
- Session and connection management
- Creating Hibernate DAOs
- Using the `hibernateTemplate` class

## **Web Frameworks**(rev:1161)

### **Spring MVC**

- Introduction to SpringMVC
- Understand the fundamentals of SpringMVC
- SpringMVC components
- SpringMVC annotations
- Map URLs to handlers
- Understand scopes
- Discuss the different handlers
- Understand views

### **Spring and JSF (Optional)**

- Brief overview of JSF
- Understand the integration between JSF and Spring
- Understand bean resolution

## **Java Unit testing techniques**(rev:4589)

### **Spring TestContext framework**

- Introduce the Spring TestContext framework
- Understand the design of this framework
- Understand how to use spring's test framework with different test environments
- Understand in particular use in JUnit 4.x
- Understand how to use DI for your fixtures
- Understand transactional tests
- Understand how to perform unit and integration tests
- List the TestContext annotations
- Extend the TestContext framework
- Integrate DBUnit in JUnit 4.x using Spring
- Extend the framework (automatic support for DBUnit)

