SKILLS CAMPUS
Training & Certification Center

Notre métier, développer le vôtre

# Spring 3 framework, develop enterprise applications

★★★★★

MR-179 4 Days (28 Hours)

## Description

This training will allow you to learn how to use the Spring framework to develop more efficient and robust Java applications. You will see how to facilitate the configuration of the different components of your application, carry out tests more easily and reduce development efforts.

## Who is this training for ?

### For whom

Java/Java EE developers.

### Prerequisites

Good knowledge of Java development. Experience in enterprise application development required.

## Training objectives

- This training will allow you to design web applications using a lightweight container, apply dependency injection and aspect-based programming in your application, integrate the main Spring modules such as Spring Core, Spring MVC, and Spring Security, and implement a data access layer

## Training program

### Introduction

- Presentation of the Java EE ecosystem.
- The place of Spring in this ecosystem, particularly in relation to EJBs.
- The state of the art of best practices for designing an application.
- The role of a lightweight container.

### Conteneur léger et inversion de contrôle

- Object support by Spring.
- Management of the life cycle of objects entrusted to Spring (Singleton pattern, Prototype).
- Programming by interface and weak coupling (pattern Strategy).
- Dependency injection.
- Presentation of setter and constructor injection: what type of injection in which case? Programmatic access to the container.
- Configuration by annotations.
- Practical work Installation of Spring and its environment.
- Upgrading in practice of dependency injection within an application.

## La Programmation Orientée Aspect

- Presentation of the concept of Aspect Oriented Programming (AOP).
- Vocabulary (aspect, join point, pointcut, advice.
- ).
- Spring AOP versus AspectJ.
- Proxies (ProxyFactory.
- ).
- Advice types (
- Before,
- AfterReturning.
- ).
- Declaration and implementation of an aspect.
- The different expression formats of Pointcut.
- XML configuration versus annotations.
- Practical work Putting aspect-based programming into practice within an application.

## Tests unitaires

- Quality improvement through unit testing with JUnit and mocks.
- The contribution of the IOC.
- Integration of JUnit into Spring.
- Practical work Setting up unit tests within Spring.

## Mise en oeuvre de l'accès aux données

- Spring/JDBC coupling (JdbcTemplate).
- Presentation of the challenges of relational mapping.
- JPA (and JPA 2) as an API for standardizing persistence in Java.
- Hibernate as an implementation of JPA.
- Spring / JPA coupling.
- Transactions implementation: by aspect, by annotation, programmatic.
- Practical work Implementation of a data persistence layer with JPA.

## Le Web

- The MVC model.
- Implementing the MVC pattern with Spring Web-MVC.
- Using annotations in implementing the MVC model.
- Spring - JSR 303 (Bean Validation) coupling for input validation.
- Implementation of Ajax with Spring MVC.
- Practical work Implementing the presentation layer with Spring-MVC.

## La sécurité

- Introduction to Spring Security for managing authentication and authorization.
- Creating, loading, and retrieving a security context.
- Enabling a filter.
- Creation of a login form.
- Securing methods.
- Application of a security strategy by aspects, by annotations.
- Practical work Setting up security with Spring Security.

## Spring Remoting

- What does Spring Remoting do? The different types of remote access (RMI, JAX-WS, JAX-RPC.
- ).
- Reminder about RMI and serialization.
- Using RMI with Spring.
- Configuring the client and server.
- Reminder about REST Web Services.
- Integration of REST and Spring MVC.
- Configure the request, transform the response.
- Practical work Integration of RMI and REST Web Services in the application.

## Spring et JMS

- Concept of synchronous or asynchronous communication.
- JMS callbacks.
- Concept of producer and consumer.
- Possible destination type: Queue , Topic.
- Implementations.
- Integration and configuration of JMS in Spring.
- Using "Point to Point" and "Publish" mode /Subscribe".
- Defining a JmsTemplate.
- Synchronous and asynchronous reception of messages.
- Sending messages.
- Practical work Produce and consume JMS messages posted in a Topic from Spring.

## Spring et JMX

- Presentation of the architecture and concepts of JMX (Managed Beans, JMX agents, MBean Server).
- Bean Spring.
- Annotations (
- ManagedResource,
- ManagedAttribute,
- ManagedOperation.
- ).
- Configuration Spring.
- Creation of an MBean Server.
- MBeans supervision and administration tool: JConsole.
- Practical work Administrating Spring Beans via JConsole.