

# Framework Spring 3, développer des applications d'entreprise

-Référence: **MR-179**

-Durée: **4 Jours (28 Heures)**

## Les objectifs de la formation

- Cette formation vous permettra de concevoir des applications web en utilisant un conteneur léger, d'appliquer l'injection de dépendance et la programmation par aspect dans votre application, d'intégrer les principaux modules Spring tels que Spring Core, Spring MVC, et Spring Security, et d'implémenter une couche d'accès aux données

## A qui s'adresse cette formation ?

### POUR QUI :

- Développeurs Java/Java EE.

### PRÉREQUIS :

- Bonnes connaissances en développement Java.
- Expérience requise en développement d'applications d'entreprise.

## Programme

- **Introduction**
  - Présentation de l'écosystème Java EE.
  - La place de Spring dans cet écosystème, notamment par rapport aux EJB.
  - L'état de l'art des bonnes pratiques de conception d'une application.
  - Le rôle d'un conteneur léger.
- **Conteneur léger et inversion de contrôle**
  - La prise en charge des objets par Spring.
  - La gestion du cycle de vie des objets confiés à Spring (pattern Singleton, Prototype).
  - La programmation par interface et le couplage faible (pattern Strategy).
  - L'injection de dépendance.
  - Présentation de l'injection par setter et par constructeur : quel type d'injection dans quel cas ? L'accès programmatique au conteneur.
  - La configuration par annotations.
  - Travaux pratiques Installation de Spring et de son environnement.

- Mise en pratique de l'injection de dépendance au sein d'une application.
- **La Programmation Orientée Aspect**
  - Présentation du concept de Programmation Orientée Aspect (AOP).
  - Vocabulaire (aspect, join point, pointcut, advice.
  - ).
  - Spring AOP versus AspectJ.
  - Les proxies (ProxyFactory.
  - ).
  - Les types d'advice (
    - Before,
    - AfterReturning.
    - ).
  - Déclaration et implémentation d'un aspect.
  - Les différents formats d'expression de Pointcut.
  - Configuration XML versus annotations.
  - Travaux pratiques Mise en pratique de la programmation par aspect au sein d'une application.
- **Tests unitaires**
  - Amélioration de la qualité par les tests unitaires avec JUnit et les mocks.
  - L'apport de l'IOC.
  - Intégration de JUnit à Spring.
  - Travaux pratiques Mise en place de tests unitaires au sein de Spring.
- **Mise en oeuvre de l'accès aux données**
  - Couplage Spring/JDBC (JdbcTemplate).
  - Présentation des enjeux du mapping relationnel.
  - JPA (et JPA 2) comme API de normalisation de la persistance en Java.
  - Hibernate comme implémentation de JPA.
  - Couplage Spring / JPA.
  - Mise en oeuvre des transactions : par aspect, par annotation, programmatique.
  - Travaux pratiques Implémentation d'une couche de persistance des données avec JPA.
- **Le Web**
  - Le modèle MVC.

- La mise en oeuvre du pattern MVC avec Spring Web-MVC.
- Utilisation des annotations dans la mise en oeuvre du modèle MVC.
- Couplage Spring - JSR 303 (Bean Validation) pour la validation de la saisie.
- Mise en oeuvre d'Ajax avec Spring MVC.
- Travaux pratiques Implémentation de la couche de présentation avec Spring-MVC.
  
- **La sécurité**
  - Présentation de Spring Security pour la gestion de l'authentification et des autorisations.
  - Création, chargement et récupération d'un contexte de sécurité.
  - Activation d'un filtre.
  - Création d'un formulaire de login.
  - Sécurisation des méthodes.
  - Application d'une stratégie de sécurité par aspects, par annotations.
  - Travaux pratiques Mise en place de la sécurité avec Spring Security.
  
- **Spring Remoting**
  - Que fait Spring Remoting ? Les différents types d'accès distants (RMI, JAX-WS, JAX-RPC.
  - ).
  - Rappel sur RMI et la sérialisation.
  - Utilisation de RMI avec Spring.
  - Configuration du client et du serveur.
  - Rappel sur les Web Services REST.
  - Intégration de REST et de Spring MVC.
  - Paramétrer la requête, transformer la réponse.
  - Travaux pratiques Intégration RMI et des Web Services REST dans l'application.
  
- **Spring et JMS**
  - Notion de communication synchrone ou asynchrone.
  - Rappels JMS.
  - Notion de producteur et de consommateur.
  - Type de destination possible : Queue, Topic.
  - Les implémentations.
  - Intégration et configuration de JMS dans Spring.
  - Utilisation du mode "Point to Point" et "Publish/Subscribe".

## Programme

- Définition d'un JmsTemplate.
  - Réception synchrone et asynchrone de messages.
  - Envoi de messages.
  - Travaux pratiques Produire et consommer des messages JMS déposés dans un Topic depuis Spring.
- **Spring et JMX**
    - Présentation de l'architecture et des concepts de JMX (Managed Beans, agents JMX, MBean Server).
    - Bean Spring.
    - Les annotations (
      - ManagedResource,
      - ManagedAttribute,
      - ManagedOperation.
    - ).
    - Configuration Spring.
    - Création d'un MBean Server.
    - Outil de supervision et d'administration de MBeans : la JConsole.
    - Travaux pratiques Administration de Beans Spring via la JConsole.



(+212) 5 22 27 99 01



(+212) 6 60 10 42 56



Contact@skills-group.com

Nous sommes à votre disposition :  
De Lun - Ven 09h00-18h00 et Sam 09H00 – 13H00

Angle bd Abdelmoumen et rue Soumaya, Résidence Shehrazade 3, 7ème étage N° 30  
Casablanca 20340, Maroc