SKILLS CAMPUS
Training & Certification Center

Framework Spring 4, développer des applications d'entreprise

-Réference: **MR-87** -Durée: **5 Jours (35 Heures)**

Les objectifs de la formation

- Mettre en place Spring sur les différentes couches d'une application n-tiers
- Mettre en place une façade Web REST et MVC
- Assurer la persistance de vos données
- Sécuriser votre application avec Spring
- SecurityCoupler Spring à d'autres technologies de l'écosystème Java JEE

A qui s'adesse cette formation ?

POUR QUI :

- Développeurs Java/Java EE ayant une réelle expérience en développement d'applications Java/JEE.

Programme

- **Introduction**
  - Les bonnes pratiques de conception : séparation des responsabilités, KISS, DRY, POJO.
  - Positionnement de Spring par rapport à la spécification Java EE et aux serveurs d'applications Java EE.
  - Les nouveautés par rapport à la version 3.
- **Présentation du conteneur Spring**
  - Les technologies " core " : cycle de vie des objets, injection de dépendances et interceptions.
  - L'accès aux données : JDBC, JPA, Hibernate.
  - La gestion des transactions.
  - Le Web : webmvc et REST.
  - L'intégration aux autres technologies.
  - Les stratégies de déploiement : servlet container, serveur d'applications, TC Server, Cloud.
  - Au-delà de Spring Framework : Spring Data, Spring Batch.

Programme

- **L'environnement de travail**
  - L'environnement Eclipse.
  - Le plug-in Spring Tool Suite (STS).
  - Les dépendances Maven.
  - Travaux pratiques Installation de l'IDE et du plug-in Spring Tool Suite.
- **Découpage n-tiers, injection de dépendances et interceptions**
  - Le découpage en couches, l'approche POJO.
  - La gestion de l'état : application stateful vs application stateless.
  - L'injection de dépendances par setter, par constructeur.
  - Les intercepteurs et la programmation orientée aspect, couplage avec AspectJ.
  - Les invocations asynchrones.
  - Les invocations planifiées (task et scheduling).
  - La configuration du conteneur (par annotations ou par xml), les profils.
  - Le test des beans managés : couplage avec JUnit, utilisation des Mocks.
  - Travaux pratiques Mise en place d'une application n-tiers, utilisation de Spring comme factory pour gérer nos objets (cycle de vie, injection de dépendance, interception).
- **Accès aux données et gestion des transactions**
  - Le couplage avec Spring JDBC et le JDBCTemplate.
  - Utilisation de bases de données embarquées : H2, HSQL, Derby.
  - Couplage avec l'API JPA et le framework Hibernate.
  - Le moniteur transactionnel de Spring, couplage avec JTA 1.
  - 2.
  - Le support des transactions au sein des tests.
  - Introduction à Spring Data pour les bases NoSQL.
  - Travaux pratiques Mise en place d'entités persistantes dans un projet Java.
  - Gestion par Spring du cycle de vie de l'entityManager JPA, utilisation du moniteur transactionnel de Spring et des annotations JTA.
- **IHM Web avec Spring MVC**
  - Rappel du pattern MVC.
  - Le 2 ways binding à base de POJO.
  - Les vues : JSP et JSTL, Freemarker, Velocity, JasperReports.
  - La validation de surface avec JSR 303.

Programme

- L'internationalisation.
- La gestion des exceptions.
- Travaux pratiques Mise en oeuvre de Spring MVC au sein d'une application Web qui expose la couche métier développée précédemment.

- **Mise en oeuvre de Web Services REST**
  - Qu'est-ce qu'une URI ? Rappels des bases du protocole http : les verbes, les codes de retours, les contentType, les en-têtes http.
  - Utilisation de Spring MVC pour mettre en place une application Web RESTFUL.
  - Cross Origin Shared Ressource.
  - Utilisation de Spring cache pour mettre en cache les réponses des endpoints REST.
  - Mise en place d'une API découvrable avec HATEOAS.
  - Travaux pratiques Développement d'une façade Web exposant les services développés précédemment, consommation des Web Services REST par un client html / javascript (utilisation d'angularJS).

- **Mise en oeuvre des websockets**
  - Les concepts du Web " real time ".
  - Présentation de STOMP et SockJS.
  - Le support par les navigateurs, les mécanismes de fallback.
  - La mise en place côté serveur.
  - La mise en place côté client (HTML, iOS, Android).
  - Exercice: Ajout d'une connexion websocket au client html/javascript développé précédemment.

- **Eléments de 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 et couplage avec Spring MVC.

- **Intégration de Spring à d'autres technologies**
  - L'enjeu posé par l'intégration entre Spring et des technologies tierces.
  - Le couplage entre Spring et JMX : managedResource et managedOperation.

Programme

- Activation de la console Web JMX de Sun (HtmlAdaptorServer).
- Exposition avec http Invoker.
- Travaux pratiques Supervision avec JMX, mise en place d'un endpoint avec http Invoker afin d'exposer les services développés précédemment.

(+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, 7éme étage N° 30
Casablanca 20340, Maroc