

Spring

Mettre en œuvre les bases du frameworks Spring

Spring est un framework d'intégration qui permet de gérer les technologies de présentation, de persistance ou services tout en limitant fortement l'adhérence aux technologies d'implémentation.

La clé en est l'utilisation du pattern IoC (Inversion of Control).
Ce type de solution est également dénommé « conteneur léger ».

Il s'agit également d'un outil open source plébiscité et très actif.
La nouvelle spécification EJB3 de reprend en grande partie les « best practices » développés par Spring.

Détails

- Code : JF-SPR
- Durée : 3 jours (21 heures)

Public

- Architectes
- Chefs de projets
- Ingénieurs

Pré-requis

- Bonne pratique de Java

Objectifs

- Etre capable de développer une application Java EE en utilisant Spring
- Connaître et assimiler les concepts d'un conteneur « léger »
- Connaître les apports de la Programmation Orientée Aspect (AOP)
- Savoir utiliser les bonnes pratiques de développement Spring

Programme

Introduction

- Concepts de conteneur léger
- Vue d'ensemble et exemples d'utilisation
- Pattern 'Inversion de Contrôle (IoC) par Injection de dépendance'

Mise en oeuvre : le conteneur léger de Spring

- Vue d'ensemble du framework
- Les Beans, BeanFactory et ApplicationContext
- Modes singleton ou normal
- Méthodes d'injection de dépendance
- Configuration de beans spécifiques à Spring, cycle de vie
- Définition de bean abstrait et héritage
- Injections automatiques
- APIs d'accès aux ressources
- Techniques de validation (PropertyEditors, DataBinder, BeanWrapper)

Framework MVC Web spécifique à Spring

- Orientations MVC2, rappels
- Configuration du serveur
- Les contrôleurs
- Les gestionnaires de vues
- Les mappings
- Spring WebFlow
- Intégration avec JSF et Struts

Pattern DAO, Accès aux données avec JDBC

- Classes abstraites pour le support DAO

- Utilisation de JDBC via les APIs d'encapsulation Spring

Persistance des composants

- Gestion des ressources
- Configuration de SessionFactory dans le contexte d'application Spring
- IoC : HibernateTemplate et HibernateCallback
- Implémentation DAO avec les APIs Hibernate
- Démarcation de transactions par programmation et déclaration

Programmation Orientée Aspect (AOP)

- Les concepts
- La terminologie utilisée : Aspect, Joinpoint, Advice, Pointcut, Introduction, Target object, AOP Proxy, Weaving
- Les différents types « advice » : Around, Before, Throws, After
- Les objectifs et fonctionnalités AOP de Spring
- Intégration avec le framework AspectJ

Gestion des transactions

- La couche d'abstraction Spring
- Les stratégies de transaction : par programmation ou par aspects
- Mise en oeuvre d'une gestion transactionnelle sans programmation et par les aspects

Intégration de technologies

- RMI

- EJB
- JMS
- Web Services
- Services REST

Spring Test

- Spring et le TDD
- Annotations de test

Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages