



Programme détaillé formation 'SPRING + JPA + JSF'- 5j

Objectifs : fournir connaissances théoriques et pratiques permettant d'être un leader technique Architecte Java

Audience : Développeurs, chefs de projets .

Prérequis : pratique développement java/j2ee .

Moyens pédagogiques : 1 ordinateur/stagiaire. Supports cours. Travaux pratiques. Vidéoprojecteur. Tests

Durée : 5 jours (total 35 h)

Vous allez apprendre à

- ▶ Créer une architecture en couches avec JSF , Spring , hibernate.
- ▶ Mettre en oeuvre l'injection de dépendances avec Spring
- ▶ Optimiser le paramétrage du conteneur léger de Spring
- ▶ Mettre en oeuvre la programmation Aspect (Spring AOP et AspectJ)
- ▶ Accéder à vos données avec Spring JDBC et Spring Data
- ▶ Comprendre et mettre en oeuvre les transactions avec Spring
- ▶ Créer des écrans web avec Spring MVC 3
- ▶ Tester plus facilement vos applications Spring
- ▶ Sécuriser vos applications avec Spring Security

Cours détaillé formation SPRING

Spring 3 : comprendre le conteneur léger

- ▶ **Notion clé : injection de dépendance**
- ▶ Conteneur 'léger' spring / conteneur 'lourd' j2ee
- ▶ Fichier ApplicationContext.xml
- ▶ Cycle de vie d'un bean Spring
- ▶ scope singleton, prototype
- ▶ Interfaces de callback
- ▶ Les post-processors
- ▶ Injection par setter
- ▶ Injection par constructeur

Spring 3 : architecture

- ▶ Approche modulaire des librairies Spring 3 (jars)
- ▶ Noyau : Spring Core au coeur de l'injection
- ▶ Accès données : Spring DAO, Spring ORM
- ▶ Applications web : Spring Web, Spring MVC
- ▶ Nouveauté Spring 3 : Spring Expression Language (spEL)
- ▶ Tests avec Spring

Spring 3 : paramétrage conteneur

- ▶ Propriétés simples (String, int,...)

- ▶ Propriétés de type collection (,)
- ▶ Méthodes d'injection (setter, constructeur)
- ▶ Auto-câblage (Autowiring) byName, byType
- ▶ Optimisation : autowiring annotation (context:annotation-config)
- ▶ Optimisation : auto-découverte (context:component-scan)
- ▶ Annotations @Component, @Service, @Controller
- ▶ Classe abstraite (abstract=true)
- ▶ Héritage (parent=true)

Spring 3 : Programmation orientée aspect

- ▶ Notion clé : Interception
- ▶ Pointcut, Advice, JoinPoint
- ▶ Notion de Tissage, Tisseur
- ▶ Présentation framework AOP AspectJ
- ▶ Comparaison tisseurs Spring AOP / AspectJ

Spring 3 : accès données JDBC

- ▶ Les classes de la couche d'abstraction
- ▶ Mise en oeuvre design pattern DAO
- ▶ Classes fournies : JdbcTemplate, JdbcDaoSupport
- ▶ Gestion des exceptions : gamme d'exceptions Spring
- ▶ Exécuter des requêtes de lecture, écriture
- ▶ Annotation @Repository

Spring 3 : Gestion des transactions

- ▶ Concept de transaction
- ▶ Gérer les transactions avec Spring
- ▶ Transactions programmatiques
- ▶ Transactions déclaratives
- ▶ Annotation @TransactionManagement

Spring 3 : création d'écrans web avec Spring MVC

- ▶ Cinématique d'une requête Spring MVC : 6 étapes clés
- ▶ Configuration Spring MVC : 4 étapes clés
- ▶ Comprendre le DispatcherServlet
- ▶ Création de contrôleurs Spring MVC 3
- ▶ Annotations @Controller
- ▶ Formulaire Spring MVC : balises clés
- ▶ Formulaire Spring MVC : validation (@Valid, jsr 303)
- ▶ Formulaire Spring MVC : conversion
- ▶ Formulaire Spring MVC : internationalisation
- ▶ Intégration Spring avec templates Tiles

Spring 3 : intégration avec JSF

- ▶ Rappels JSF 2 : cycle de vie, facadelets, @notations
- ▶ Faire d'un Managed bean JSF un bean Spring
- ▶ ContextLoaderListener
- ▶ Injection bean service dans le contrôleur Spring

Spring 3 : intégration Spring/JPA Hibernate

- ▶ Accéder aux bases de données relationnelles
- ▶ Accéder aux bases de NoSQL
- ▶ Lacunes des Daoa basés sur Template / DaoSupport.
- ▶ Bonne pratique : Spring data JPA
- ▶ Spring data : pagination , tri
- ▶ Spring Data : CRUDRepository
- ▶ Gérer les exceptions

Tests avec Spring

- ▶ Tests unitaires : lien entre JUnit 4 et Spring
- ▶ Tests d'intégration : chargement du conteneur !

Objis, spécialiste formation Java

- ▶ Trest transactionnels
- ▶ Mock Tests avec Mockito

Sécurité j2ee avec Spring Security 3

- ▶ Spécification JAAS : Authentification & Autorisation
- ▶ Sécurité déclarative / programmatic
- ▶ Royaume de de sécurité (Realms)
- ▶ 4 questions : QUOI, COMMENT, QUI, OU
- ▶ fichier web.xml : balise 'security-constraint'
- ▶ fichiers context.xml, jboss-web.xml, sun-web.xml
- ▶ Comprendre les filtres Spring Security
- ▶ Personnalisation et Paramétrage de filtres (ex : formulaire, cookies, session)