



zenika

www.zenika.com

Durée
4 jours (28h)

Répartition
40% théorie
60% pratique

Pré-requis
Connaissance de Java

Public
Architecte
Développeur
Chef de projet

Tarif (déjeuners inclus)
2295 € (HT)
Tarif (2 semaines avant)
2095 € (HT)

Lieu
Paris 9ème

Sessions 2008
01 au 04 juillet
16 au 19 septembre
21 au 24 octobre
25 au 28 novembre
15 au 18 décembre

Intra-entreprise sur demande

Renseignements et inscription
+33(0)1.45.26.19.15
training@zenika.com
www.zenika.com

Spring

La formation officielle



Objectifs

- Utiliser Spring pour développer tous les composants d'une application JEE
- Comprendre le principe des containers légers et le pattern Inversion of Control
- Découvrir les enjeux de la programmation orientée aspects (AOP)
- Comprendre comment Spring facilite l'écriture de tests et l'amélioration du code

Contenu

Les bases du framework Spring

- Introduction à l'architecture du container léger et à l'ApplicationContext de Spring
- L'injection de dépendance et l'approche TDD (Test Driven Development) pour structurer, configurer et tester rapidement une application
- Organisation des fichiers de configuration
- Utilisation de SpringIDE pour Eclipse : autocomplétion, représentation graphique des beans Spring...
- Personnalisation du cycle de vie des beans Spring
- Les espaces de nommage XML pour une meilleure lisibilité de la configuration Spring
- Les annotations comme alternative à la configuration XML

Approche Test Driven Design (TDD) avec Spring

- Qu'est-ce que le développement piloté par les tests ?
- Tests unitaires avec JUnit, deux approches possibles : les Stubs et les Mocks
- Tests d'intégration avec les classes utilitaires Spring

La programmation orientée aspect

- Présentation des concepts de l'AOP
- Les différents types d'Advice supportés par Spring
- Visualisation des beans Spring cibles d'un aspect avec SpringIDE
- Les différentes façons de créer des proxies AOP avec Spring, comment choisir parmi ces méthodes ?
- Identification des problématiques transverses et des cibles potentielles de l'approche AOP
- Les aspects fournis par Spring
- Ecrire ses propres aspects

Persistence et transactions

- Comment Spring uniformise l'accès aux données ainsi que la gestion des exceptions
- "Best practice" Spring pour construire une couche de persistance
- Analyses et compromis sur les stratégies de persistance (Hibernate, JDBC, JPA)
- Stratégie de gestion des transactions avec Spring au niveau de la couche de services
- Implémenter l'accès aux données de manière fiable et performante en peu de lignes de code
- La démarcation transactionnelle déclarative, les règles d'utilisation de la propagation et du rollback

Architecture et conception Web

- Illustration des apports et possibilités de Spring MVC
- DispatcherServlet, Handler Mapping et ViewResolver : le cycle de vie d'une requête Spring MVC
- Configuration et chargement du contexte Spring dans un environnement web
- Les annotations MVC disponibles depuis Spring 2.5
- Les différentes stratégies de mapping
- Gestion des exceptions
- Gestion des formulaires
- Intégrer les différentes technologies de vue (JSP, PDF, Excel, Velocity et Freemarker)
- Implémenter une conversation avec état dans une application web grâce à Spring Web Flow
- Intégration de Spring avec Struts (actions spécifiques ou actions Struts standards)
- Intégration de Spring avec JSF

Intégration aux services d'entreprise

- Configuration de la distribution des services gérés par Spring sans écriture de code
- Conception de web services type "contract-first" avec Spring Web Services
- Exporter un service transactionnel à des clients distribués avec Spring remoting
- Configurer le même service pour un faire un Message-Driven POJO avec Spring JMS
- Monitoring et gestion d'une application grâce à Spring JMX

La sécurité

- Gestion de la sécurité avec Spring security (ACEGI)
- Les nouveautés de Spring Security 2.0
- Gestion de l'authentification
- Protection des URLs
- Protection de l'accès à certaines méthodes avec le protect-pointcut
- Mécanismes avancés