



Nous allons vous faire aimer Java

# Formation Architecture JEE (5j)

**Destiné à des développeurs Java expérimentés (4 à 6 ans exp) , cette formation fournit les connaissances théoriques et pratiques nécessaires pour pouvoir prétendre à un rôle de d'architecte ou leader technique dans un projet Java / JEE.** Identifiez les différentes problématiques (sécurité, persistance, transactions,haute disponibilité...) et solutions d'architecture technique associées (avantages et inconvénients).

**Audience :** développeurs , intégrateurs, architectes

**Durée :** 5 jours

## Vous allez apprendre à :

- ▶ Valider vos acquis en développement Jee (jsp,servlets,jstl,el,ejb)
- ▶ Comprendre technologies JMX / MBeans
- ▶ Mettre en oeuvre programmation orientée aspects.
- ▶ Mettre en oeuvre des patterns de conception spécifiques JEE
- ▶ Exposer vos services métier à travers à RMI, Corba et Web Services
- ▶ Sécuriser une application architecture Java d'entreprise avec JAAS
- ▶ Évaluer les frameworks de persistance (JDBC, Hibernate, JPA/EJB3...)
- ▶ Évaluer l'impact des EJB (Enterprise JavaBeans)
- ▶ Mettre en oeuvre une architecture asynchrone via JMS / MOM
- ▶ Comprendre les services fournis par les EJB (EJB2/EJB3)
- ▶ Comprendre les stratégies de monitoring et tuning performances

## Programme détaillé

### Programmation Aspects

- ▶ Principes et valeur ajoutée
- ▶ Différences avec la POO
- ▶ Tisseurs : Spring AOP, AspectJ, Jboss AOP
- ▶ Tissage à la compilation
- ▶ Tissage à l'exécution
- ▶ Tissage et serveur d'application
- ▶ Mise en oeuvre Audit

### Injection de dépendances

- ▶ Présentation du concept
- ▶ Pattern IoC
- ▶ Conteneur Léger
- ▶ Mise en oeuvre avec Spring
- ▶ Plexus, Google Guice

### JMX

- ▶ Spécification
- ▶ Instrumentation
- ▶ Agent JMX
- ▶ Client JMX
- ▶ jconsole
- ▶ Mis en oeuvre

### Architecture en couche

- ▶ Valeur ajoutée : conception, maintenance
- ▶ Interfaces entre les couches
- ▶ couche Service : bonnes pratiques
- ▶ couche DAO : bonnes pratiques
- ▶ Mise en oeuvre avec Spring

## **Couche Persistance**

- ▶ Spécification JPA
- ▶ Frameworks : Hibernate, EclipseLink
- ▶ Aspects statiques et dynamiques
- ▶ Objets détachés
- ▶ Paramétrages
- ▶ Transaction

## **Couche Service**

- ▶ Couche service avec POJO/RMI
- ▶ Couche service avec Web Services
- ▶ Couche service avec EJB3

## **Couche Client**

- ▶ Clients Java : lequel choisir ?
- ▶ Client JavaScript : lequel choisir ?
- ▶ Frameworks Ajax : ExtJS, GWT, Flex, SilverLight
- ▶ Les API de présentation (AWT, SWING, SWT)
- ▶ Problématiques déploiement d'un client riche
- ▶ Java WebStart ou Eclipse RCP

## **Web Services**

- ▶ Web Service SOAP
- ▶ Web Services REST
- ▶ Analyse de trames
- ▶ Comparaison SOAP / REST
- ▶ Test de conformité d'un Web Service
- ▶ Mise en oeuvre framework Apache CXF
- ▶ Gouvernance et qualité de service

## **Les services techniques Java EE**

- ▶ Annuaire JNDI,
- ▶ Middleware MOM avec JMS / ActiveMQ
- ▶ Sécurité JAAS,
- ▶ Persistance JPA,
- ▶ Transaction JTA
- ▶ Transactions distribuées

## **EJB et EJB 3**

- ▶ Services offerts par le conteneur
- ▶ Opposition EJB / SPRING
- ▶ Bonnes pratiques EJB session
- ▶ EJB et Web Services
- ▶ EJB Lite

## **XML**

- ▶ Définition et utilisations
- ▶ Complémentarité avec Java
- ▶ Les API standards comme JAXP
- ▶ Comprendre JAXB, JAXP
- ▶ SAX, DOM, STAX
- ▶ Intégration basée sur flux XML

## **Sécurité**

- ▶ Modèle de sécurité de Java
- ▶ fichier java.policy
- ▶ API JAAS : Realms
- ▶ Sécuriser les composants EJB
- ▶ Sécuriser les applications Web

## **Performances JAVA**

- ▶ Méthodologie de tuning
- ▶ Analyse performances
- ▶ Outils : jconsole, jvisualvm
- ▶ Fonctionnement mémoire Java
- ▶ Garbage Collector
- ▶ paramètres JAVA\_OPTS
- ▶ Tests performance avec JMeter

### **Intégration**

- ▶ Intégration du système d'information
- ▶ Intégration java avec Apache Camel
- ▶ L'EAI et ses produits
- ▶ Urbanisme et intégration
- ▶ SOA et Web Services
- ▶ Communication Java-C/C++ via JNI

### **Tests**

- ▶ Tests unitaires avec JUnit
- ▶ Tests fonctionnel avec SELENIUM

### **Haute disponibilité**

- ▶ Load balancing
- ▶ Fail over
- ▶ apache mod\_jk
- ▶ réplication de session
- ▶ Affinité de session
- ▶ Cache distribué