



## Programme détaillé formation 'Architecture JAVA Niveau 1'- 5j

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

**Audience** : Développeurs expérimentés (4 ans +) .

**Prérequis** : pratique développement java/j2ee depuis au moins 4 ans.

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

**Durée** : 5 jours (total 35 h)

### Vous allez apprendre à

- ▶ concevoir un socle technique pour vos applications JEE
- ▶ mettre en oeuvre l'injection de dépendances : spécification CDI offre du marché : spring/Guice/Weld
- ▶ mettre en oeuvre la programmation aspects : spécification JEE 7 et offre SpringAOP / AspectJ
- ▶ choisir entre une couche service synchrone ou asynchrone (MOM)
- ▶ mettre en oeuvre une architecture synchrone : BEAN/RMI/WS/EJB/HESSIAN/BURLAP
- ▶ mettre en oeuvre une architecture asynchrone : Spring JMS / EJB3
- ▶ créer votre couche d'accès aux données (SGBDR/NoSQL) basée sur meilleures pratiques
- ▶ identifier et mettre en oeuvre des patterns de conception spécifiques JEE

### Programme détaillé

#### Architecture JEE 7

- ▶ Spécifications JEE 7 et livrables
- ▶ Architecture logique et physique
- ▶ Problématiques JEE et réponse des spécifications
- ▶ Choix frameworks : innovations ou spécifications ?
- ▶ Choix couche service : synchrone ou asynchrone ?
- ▶ Choix couche présentation : Java ou javascript ?
- ▶ Rôle de l'architecte MOE / MOA
- ▶ Gérer la gouvernance
- ▶ Le futur de JEE : JEE 8

#### Spécifications JEE

- ▶ Document clé : spécifications JEE 7 (JSR 342)
- ▶ Problématiques applications JEE
- ▶ Nouveautés JEE 7 par rapport à JEE6 / JEE5
- ▶ Implémentation de référence JEE 7 : GlassFish 4
- ▶ Repères clés : exemples, tutoriels et SDK JEE 7
- ▶ Service d'annuaire (JNDI)
- ▶ Service messages asynchrone (JMS)
- ▶ Service sécurité (JAAS)
- ▶ Service persistance objet/relationnel (JPA),
- ▶ Service transaction (JTA)
- ▶ Service Web services (JAX-WS/JAX-RS)

#### Injection de dépendances

## Objis, spécialiste formation Java

- ▶ Document clé N°1 : spécification CDI 1.0 (JSR 299) et 1.1 (JSR 346)
- ▶ Document clé N°2 : spécification DI (JSR 330)
- ▶ Problèmes du couplage fort
- ▶ CDI : 'Unifier' la notion de Bean
- ▶ Injection : Spring ou CDI/Weld ?
- ▶ @Inject / @Named / @Autowire
- ▶ Services CDI : Décorateurs, Intercepteur
- ▶ Services CDI : Event, Producers
- ▶ Démon / POC

### Programmation Aspects

- ▶ Principe et valeur ajoutée
- ▶ Différences avec la POO
- ▶ Tisseurs : Spring AOP ou AspectJ ?
- ▶ Tissage compilation, exécution, Chargement
- ▶ Tissage et serveur d'application
- ▶ Point de jonction, Coupe, Greffon
- ▶ POA dans JEE 7 : Intercepteurs
- ▶ POA et spécification CDI (JSR 299)
- ▶ Démon / POC

### Couche d'accès aux données (DAO)

- ▶ Document de référence : Spécifications JPA (JSR 338)
- ▶ Synthèse : 10 points clés couche accès aux données
- ▶ Interfaces générique et classes Abstraites
- ▶ Persistance relationnelle (sgbdr)
- ▶ Persistance non relationnelle (NoSQL)
- ▶ Les 4 types de bases NoSQL
- ▶ Spécification JPA et Implémentations
- ▶ Spring Data : CRUDRepository
- ▶ Bonne pratique : Spring Data jpa
- ▶ Quel framework ORM utiliser ? pour quoi ?
- ▶ Quel Cache utiliser pour Niveau 2 ?

### Couche Service synchrone

- ▶ Couche service synchrone avec POJO/RMI
- ▶ Couche service synchrone avec Hessian/Burlap
- ▶ Couche service avec EJB3 / Web Services
- ▶ Comprendre la différence WebService Soap/Rest
- ▶ Comprendre l'orchestration de Web Services
- ▶ Quel choix ? pourquoi ?

### Web Services

- ▶ SOAP avec jax-ws 2.2 (JSR 224)
- ▶ REST avec jax-rs 2.0 (JSR 339)
- ▶ Implémentations de référence
- ▶ Comparaison SOAP / REST
- ▶ Test de conformité d'un Web Service
- ▶ Frameworks : CXF, Metro ou Axis ?
- ▶ Gouvernance et qualité de service
- ▶ Spécifications WS-\* : gouvernance WS Soap
- ▶ Orchestration de WS avec BPEL
- ▶ Quel choix ? pourquoi ?
- ▶ Le futur des web services

### EJB 3

- ▶ spécification JSR 345 : EJB 3.2
- ▶ Services offerts par le conteneur
- ▶ Opposition EJB / SPRING
- ▶ Bonnes pratiques EJB session
- ▶ EJB et Web Services
- ▶ Nouveautés
- ▶ Quel futur pour les EJB ?

### Couche Service asynchrone

- ▶ Document clé : spécification JMS 2.0 (JSR 343)
- ▶ Pensez MOM !
- ▶ Fournisseurs de messages JMS : standalone ou dans serveur Jee ?
- ▶ Couche service asynchrone avec EJB Message
- ▶ Couche service asynchrone avec Spring JMS

## Objis, spécialiste formation Java

- ▶ Quel choix pour le fournisseur JMS ?
- ▶ Implémentation : ActiveMQ , OpenMQ , RabbitMQ ?
- ▶ Quel choix ? pourquoi ?
- ▶ Quel avenir pour MOM ?

### **Couche Présentation**

- ▶ Document de référence : spécification JSF 2.2 (JSR 344)
- ▶ Nouveautés JEE 7 avec HTML5 WebSocket (JSR 356)
- ▶ Clients web Java : lequel choisir : Spring MVC, JSF, Struts2, Wicket ?
- ▶ Client JavaScript : lequel choisir : ExtJS, GWT, jQuery ?
- ▶ Les clients Java lourd : Swing/JavaFx/EclipseRPC ?
- ▶ Problématiques déploiement d'un client riche.
- ▶ Technologie Java WebStart
- ▶ Quel avenir pour la couche présentation ?

### **Tests**

- ▶ Tests fonctionnel web avec SELENIUM
- ▶ Tests unitaires avec JUnit / TESTNG
- ▶ Tests d'intégration avec Spring
- ▶ Test de performance avec JMeter
- ▶ Quel avenir pour les Tests JEE ?