



Programme détaillé formation 'Architecture JAVA Niveau 2'- 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

- ▶ mettre en oeuvre JMX / MBeans pour la supervision d'applications Java
- ▶ sécuriser une application : JAAS , fichiers .policy, certificats
- ▶ Identifier les frameworks sécurité : Spring Security, Apache Shiro, JGuards
- ▶ comprendre les stratégies de monitoring et tuning performances
- ▶ comprendre les différentes briques d'une chaîne d'intégration continue
- ▶ faire le bon choix entre framework d'intégration (camel/spring integ) et ESB (Fuse/Mule)
- ▶ mettre en oeuvre une stratégie de haute disponibilité (horizontal/vertical) .
- ▶ Mettre en oeuvre OSGI coté serveur JEE

Programme détaillé

Osgi

- ▶ Document de référence : specs OSGI
- ▶ Services : modularité, cycle de vie
- ▶ Bundle OSGI, fichier MANIFEST.MF
- ▶ Framework OSGI : Felix, Equinox
- ▶ Moteur de services
- ▶ Création composant OSGI
- ▶ OSGI dans serveur Glassfish 4
- ▶ OSGI dans serveur Jboss 7
- ▶ Plateforme de service OSGI : Apache Karaf
- ▶ Outil développement OSGI : BndTools

Transactions

- ▶ Document de référence : JSR 907 (JTA)
- ▶ Transaction déclarative
- ▶ Transaction programmatique
- ▶ 5 propriétés d'une transaction
- ▶ propriété transaction : Timeout
- ▶ propriété transaction : Read-Only
- ▶ propriété transaction : Règles Rollback
- ▶ propriété transaction : Niveau d'Isolations
- ▶ propriété transaction : Propagation
- ▶ Annotation @Transactional
- ▶ Spring Transactions

▶ JOTM

Sécurité

- ▶ Document de référence 1 : JSR 115 (JACC)
- ▶ Document de référence 2 : JSR 196 (JASPIC)
- ▶ API JCE , JSSE
- ▶ 3 niveaux d'attaques
- ▶ fichier java.policy
- ▶ API JAAS : Realms
- ▶ Sécuriser WAR et EJB
- ▶ Single Sign On
- ▶ Apache Shiro
- ▶ JGuards
- ▶ Spring Security
- ▶ SAML / OpenAM
- ▶ Démonstrations / POC

Monitoring JMX

- ▶ Document de référence : JSR 160
- ▶ Spécification JMX
- ▶ Architecture JMX : 3 couches
- ▶ Instrumentation
- ▶ Agent / MBean Server
- ▶ Types de client JMX
- ▶ outil : jvisualvm
- ▶ Consoles d'administration JMX
- ▶ Démonstrations / POC

Performances JAVA

- ▶ Méthodologie de tuning
- ▶ Analyse performances
- ▶ Outils : jconsole, jvisualvm
- ▶ Fonctionnement mémoire Java
- ▶ Old Generation, Young generation
- ▶ PermSize, S0, S1
- ▶ Garbage Collector
- ▶ paramètres JAVA_OPTS
- ▶ Tests performance avec JMeter
- ▶ Démonstrations / POC

Haute disponibilité

- ▶ Cluster = LB + FO
- ▶ Load balancing avec mod_jk
- ▶ Reprise sur incident (Fail over)
- ▶ Réplication de session
- ▶ Affinité de session (sticky session)
- ▶ Cache niveau 2 des Entités JPA
- ▶ Multicast IP / Jgroups
- ▶ Démonstrations / POC

Intégration Java / SI

- ▶ Présentation SOA / ESB
- ▶ Critères de choix d'un ESB
- ▶ Frameworks d'intégration = ESB 'light'
- ▶ EIP Patterns : patterns d'intégrations
- ▶ Service clé : routage
- ▶ Service clé : transformation
- ▶ Service clé : médiation
- ▶ Service clé : Sécurité ,
- ▶ Service clé : Transactions
- ▶ Service clé : Haute disponibilité
- ▶ Esb léger : Camel ou Spring Intégration ?
- ▶ Esb lourd : Fuse ESB ou Mule ?
- ▶ Urbanisme et intégration
- ▶ Démonstrations / POC