

I. Introduction

- Programmation par composants
- Intégré à J2EE
- Communique en RMI avec des clients Java ou avec des servlets J2EE / JSP

II. Types d'EJB

- Entité : représente un objet du modèle (persistance gérée par EJB ou conteneur)
- Session : extension du client sur le serveur / logique métier / interfaçable par client
- Message : producteur ou consommateur (utilise JMS)

III. Conteneur EJB

- Environnement d'exécution
- Fourni la persistance, transactions, cache, sécurité, gestion d'erreurs, portabilité.

IV. Concepts des EJB

- Contrat : répartition des responsabilités entre client, conteneur, EJB, persistance
- Services orthogonaux fournis par le conteneur au développeur
- Interposition du conteneur entre client et EJB

V. Utilisateurs

- Les clients doivent posséder un compte utilisateur sur le serveur JBOSS et le renseigner dans le fichier de propriétés de l'application client.

VI. Annotation EJB

- Interface `@Remote` : Accessible à distance
- Interface `@Local` : Accessible en local
- Classe `@Stateless` : Implémentation sans état (implements les interfaces Local et Remote)
 - Méthode `@PostConstruct` `postConstruct()`
 - Méthode `@PreDestroy` `preDestroy()`
- Classe `@Statefull` : Implémentation avec état (implements les interfaces Local et Remote)
 - Méthode `@PostConstruct` `postConstruct()`
 - Méthode `@Init` `init()`
 - Méthode `@PrePassivate` `prePassivate()`
 - Méthode `@PostActivate` `postActivate()`
 - Méthode `@Remove` `remove()`
 - Méthode `@PreDestroy` `preDestroy()`
- **Persistance :**
 - géré par un `@PersistenceContext` `EntityManager` `em` ou application `@PersistenceUnit`
 - Classe EJB entité `@Entity` ou `@Table`
 - constructeur par défaut, attributs privés et accesseurs, coder `hashCode` et `equals`
 - `@Id` : clé primaire / `@GeneratedValue` pour générer sa valeur
 - `@Column`