

Technologies Java



Houssem MAHMOUDI
houssem.mahmoudi@ensicaen.fr

Chapitre 3 : Architecture Java EE

1. Définition
2. Modèle MVC
3. Architecture Java EE
4. Composants Java EE
 - a. Composants Web
 - b. Composants EJB
 - c. Composants JPA
 - d. Services Web
 - e. Autres composants

I. Définition

- **Java SE** « Java Standard Edition » : Est une plateforme de développement des applications à base Java **coté client**.
- **Java EE** « Java Entreprise Edition » Est une plateforme (ensemble des APIs) pour développer des applications **cotés serveurs** (web application, web service, EJB...).
- Domaines d'application de Java EE :
 - Banque et finance : Transactions bancaire
 - E-commerce : Gestion de commandes et des paiements
 - Santé : Gestion des dossiers
 - ...



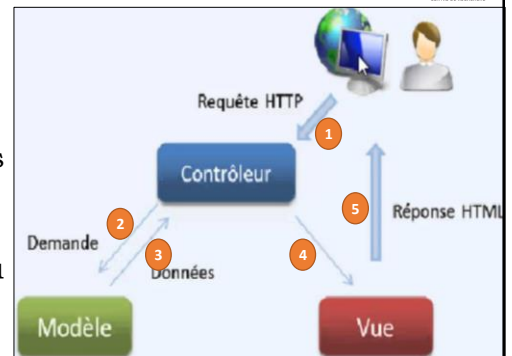
II. Modèle MVC

II. Modèle MVC

- Le modèle MVC (Modèle-Vue-Contrôleur) est une architecture logicielle utilisée pour structurer le code d'une application en trois composants.
 - ❖ **Modèle** : Gère l'accès et la manipulation des données. Il représente la logique métier (business logic) pour définir comment une application doit fonctionner.
 - ❖ **Vue** : Responsable de l'interface utilisateur et de l'affichage des données dans une page HTML.
 - ❖ **Contrôleur** : Intermédiaire entre le modèle et la vue. Le contrôleur reçoit les requêtes de l'utilisateur, interagit avec le modèle pour traiter les données, puis transmet les résultats à la vue pour l'affichage.

II. Modèle MVC

0. L'utilisateur interagit avec l'interface (vue).
1. Le contrôleur reçoit la requête et l'analyse.
2. Il demande au modèle d'effectuer les traitements nécessaires.
3. Le modèle transmet les données mises à jour au contrôleur.
4. Le contrôleur transmet les données à la vue.
5. La vue affiche les données mises à jour à l'utilisateur (Réponse HTML).



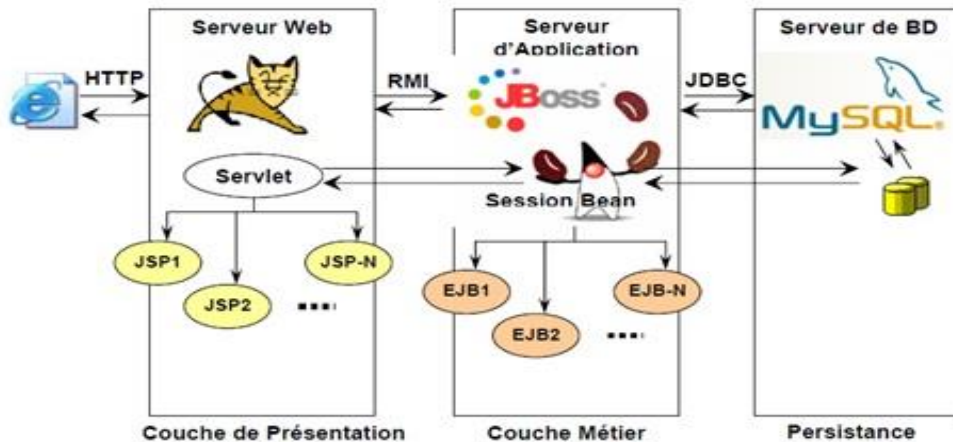
II. Modèle MVC – Avantages

- Séparation des préoccupations : L'application est divisée en trois composants distincts, chacun ayant une responsabilité.
- Réutilisabilité du code : Composants du modèle MVC sont indépendants les uns des autres.
- Clarté et maintenance Facile: Corriger des bugs ou ajouter des fonctionnalités.
- Adaptabilité aux Frameworks modernes : Nombreux Frameworks utilisent le modèle MVC (Spring MVC, Struts, Laravell, Djingo,...).
- Sécurité améliorée : La séparation des composants permet de mieux contrôler les accès aux données et aux fonctionnalités.

III. Architecture Java EE

III. Architecture Java EE

- Java EE utilise une architecture basée sur les composants, qui sont exécutés dans des conteneurs spécifiques : **Conteneur Web** et **Conteneur EJB**



III. Architecture Java EE

- Conteneur Web** : Héberge les servlets, les pages JSP et les composants JavaServer Faces.
- Conteneur EJB** : Gère les Enterprise JavaBeans, qui encapsulent la logique métier.
- Dans l'architecture Java EE, un serveur d'application permet d'héberger, gérer et fonctionner les deux conteneurs Web & EJB.
- Exemple des serveurs d'applications :
 - GlassFish
 - Jboss
 - WebLogic Server
 - Apache Tomcat

IV. Composants Java EE

IV. Composants Java EE



1. Composants Web

- **Servlet** : Classes Java qui traitent les requêtes **HTTP** et génèrent des réponses dynamiques. Ils forment la base des applications web Java.
- **JavaServer Pages (JSP)** : Permettent de créer des pages web dynamiques en intégrant du code Java dans une page HTML.
- **JavaServer Faces (JSF)** : Framework web pour la création d'interfaces utilisateur riches basées sur des composants.

IV. Composants Java EE

2. Composant Métier

- **Enterprise JavaBeans (EJB)** : Composants côté serveur qui encapsulent la logique métier. Ils gèrent les transactions, la sécurité et la concurrence.
 - Stateless Session Bean : Pas de conservation d'état entre les appels.
 - Stateful Session Bean : Conserve l'état du client pendant une session.

IV. Composants Java EE

3. Composants de Persistance

- **Java Persistence API (JPA)** : API standard pour la gestion de la persistance des données et le mapping objet-relationnel.
- **Mapping objet-relationnel** est une technique de programmation qui permet de faire le lien entre un modèle de données orienté objet et une base de données relationnelle. Mapping des classes aux tables :
 - Une classe est mappée à une table.
 - Les attributs de la classe sont mappés aux colonnes de la table

IV. Composants Java EE

4. Services Web

- **JAX-WS (Java API for XML Web Services)** : API pour créer des services web basés sur le protocole **SOAP**.
- **JAX-RS (Java API for RESTful Web Services)** : API pour créer des services web **RESTful** qui utilisent HTTP comme protocole de communication.

IV. Composants Java EE

5. Autres Composants

- Java Message Service (**JMS**) : API pour la messagerie asynchrone entre composants.
- Java Transaction API (**JTA**) : Gère les transactions distribuées.
- Java Naming and Directory Interface (**JNDI**) : API pour la connexion à des services d'annuaire.
- Java Database Connectivity (**JDBC**) : API pour la connexion aux bases de données.
- JavaMail : API pour la gestion des emails.