

Programmation JEE

Ali ZAINOUL

for DITESCO
October 31, 2023



Table des matières

- 1** Introduction à JEE
 - Présentation de JEE
 - Modèle MVC dans JEE
- 2** Environnement de développement JEE
 - Installation de l'environnement de développement
 - Windows
 - Linux
 - MacOS
- 3** Développement d'une Application de Gestion de Jeux Vidéo en JEE
 - Modèle de l'Application
 - Syntaxes Utilisées
 - APIs Utilisées

Find me

Find me @Github.
This course was made with ♥.

Qu'est-ce que JEE ?

- JEE (**Java Platform, Enterprise Edition**) est une plateforme de développement d'applications d'entreprise en Java.
- Il propose un modèle de développement basé sur le **modèle MVC** (Modèle-Vue-Contrôleur).
- JEE offre des composants d'entreprise tels que les **Servlets**, les **JSP (JavaServer Pages)** et les **EJB (Enterprise JavaBeans)**.

Caractéristiques de JEE

- Il fournit des services essentiels pour les applications d'entreprise, comme la gestion des transactions et la sécurité.
- JEE favorise l'interopérabilité en utilisant des protocoles standard comme **HTTP (Hypertext Transfer Protocol)** et **SOAP (Simple Object Access Protocol)**.
- Il permet aux applications d'être hautement évolutives et disponibles.

Modèle MVC dans JEE

- Le **Modèle-Vue-Contrôleur (MVC)** est un modèle de conception couramment utilisé dans le développement d'applications JEE pour organiser le code de manière modulaire. Il divise l'application en trois composants clés :
 - Le **Modèle** représente les données et la logique métier de l'application. Il gère l'accès aux données, leur manipulation et la prise de décisions.
 - La **Vue** est responsable de l'interface utilisateur et de l'affichage des données au client. Elle rend les données du modèle de manière lisible pour l'utilisateur.
 - Le **Contrôleur** gère les interactions entre l'utilisateur et l'application. Il reçoit les demandes de l'utilisateur, traite ces demandes, met à jour le modèle et choisit la vue appropriée à afficher.

Illustration du MVC

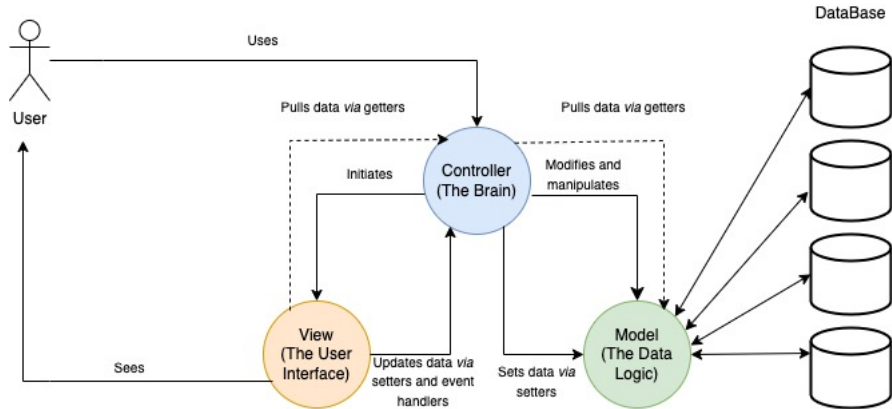


Figure: Illustration of the MVC design pattern

Installation de Java - Windows

1. Téléchargement de Java :

- Rendez-vous sur le site officiel d'Oracle ou AdoptOpenJDK pour télécharger la version de Java JDK appropriée pour Windows.

2. Installation de Java :

- Exécutez le programme d'installation téléchargé.
- Suivez les instructions de l'installateur pour terminer l'installation.
- Vérifiez que l'installation a réussi en ouvrant un terminal et en exécutant `java -version`.

3. Configuration des variables d'environnement :

- Configurez la variable d'environnement `JAVA_HOME` pour pointer vers le répertoire d'installation de Java.
- Ajoutez `JAVA_HOME\bin` au chemin système.

Installation de Tomcat - Windows

1. Téléchargement de Tomcat :

- Rendez-vous sur le site officiel Apache Tomcat et téléchargez la version de Tomcat qui correspond à vos besoins.

2. Installation de Tomcat :

- Décompressez l'archive Tomcat dans un répertoire de votre choix (par exemple, C:\Tomcat).

3. Configuration de Tomcat :

- Configurez les variables d'environnement CATALINA_HOME et CATALINA_BASE pour pointer vers le répertoire d'installation de Tomcat.
- Modifiez le fichier `server.xml` dans le répertoire `conf` de Tomcat pour configurer les ports et d'autres paramètres en fonction de vos besoins.

Installation de Java - Linux

1. **Installation de Java :**

- Utilisez le gestionnaire de packages de votre distribution Linux (par exemple, apt, yum, dnf) pour installer Java JDK.

2. **Configuration des variables d'environnement :**

- Configurez la variable d'environnement JAVA_HOME pour pointer vers le répertoire d'installation de Java.
- Mettez à jour le chemin en utilisant export dans votre fichier de profil, par exemple ~/.bashrc.

Installation de Tomcat - Linux

1. Téléchargement de Tomcat :

- Rendez-vous sur le site officiel Apache Tomcat et téléchargez la version de Tomcat qui correspond à vos besoins.

2. Installation de Tomcat :

- Décompressez l'archive Tomcat dans un répertoire de votre choix.

3. Configuration de Tomcat :

- Configurez les variables d'environnement CATALINA_HOME et CATALINA_BASE.
- Modifiez le fichier `server.xml` dans le répertoire `conf` de Tomcat.

Installation de Java - macOS

1. **Installation de Java :**

- Utilisez le gestionnaire de packages Homebrew pour installer Java en utilisant la commande `brew cask install adoptopenjdk`.

2. **Configuration des variables d'environnement :**

- Configurez la variable d'environnement `JAVA_HOME` pour pointer vers le répertoire d'installation de Java.
- Mettez à jour le chemin en utilisant `export` dans votre fichier de profil, par exemple `~/.bash_profile`.

Installation de Tomcat - macOS

1. Téléchargement de Tomcat :

- Rendez-vous sur le site officiel Apache Tomcat et téléchargez la version de Tomcat qui correspond à vos besoins.

2. Installation de Tomcat :

- Décompressez l'archive Tomcat dans un répertoire de votre choix.

3. Configuration de Tomcat :

- Configurez les variables d'environnement CATALINA_HOME et CATALINA_BASE.
- Modifiez le fichier `server.xml` dans le répertoire `conf` de Tomcat.

Modèle de l'Application

L'application de gestion de jeux vidéo suit un modèle de données solide pour gérer les jeux, les adhérents et les locations. Voici quelques éléments clés du modèle :

Modèle de l'Application (suite)

- **Jeux** : Chaque jeu est associé à des détails tels que le titre, la plateforme, le genre, etc. Le modèle gère également l'état de chaque jeu (disponible, loué, etc.).
- **Adhérents** : Les adhérents de la bibliothèque sont enregistrés avec leurs informations personnelles. Chaque adhérent peut emprunter et retourner des jeux.
- **Locations** : Les locations sont des enregistrements de jeux empruntés par les adhérents. Chaque location est associée à un adhérent, un jeu et des dates de début et de fin.

Syntaxes Utilisées

L'application utilise les technologies et les syntaxes Java EE pour créer des Servlets et des JSP qui gèrent l'interaction avec les utilisateurs. Voici quelques éléments importants de la syntaxe utilisée :

Syntaxes Utilisées (suite)

- **Servlets** : Les Servlets sont utilisés pour gérer les requêtes HTTP, traiter les formulaires et interagir avec la base de données. Les Servlets suivent une structure de classe Java EE et utilisent des annotations pour définir leurs URL de correspondance.
- **JSP** : Les JavaServer Pages (JSP) sont utilisées pour générer des pages Web dynamiques. Les JSP incluent du code Java, des balises JSTL et des expressions EL pour afficher et manipuler les données.
- **JavaBeans** : Les JavaBeans sont utilisés pour représenter les objets du modèle, tels que les jeux, les adhérents et les locations. Ils sont manipulés dans les Servlets et les JSP pour effectuer des opérations sur les données.

APIs Utilisées

L'application utilise diverses API Java EE pour gérer les fonctionnalités principales. Voici quelques-unes des API clés utilisées :

APIs Utilisées (suite)

- **Servlet API** : L'API Servlet est utilisée pour créer des Servlets qui gèrent les requêtes HTTP. Elle fournit des classes et des méthodes pour gérer les cycles de vie des Servlets.
- **JSTL (JavaServer Pages Standard Tag Library)** : JSTL est une bibliothèque de balises standard utilisée dans les JSP pour simplifier la génération de pages Web dynamiques. Elle offre des balises pour les boucles, les conditions, l'accès aux données et d'autres opérations courantes.
- **JDBC (Java Database Connectivity)** : JDBC est utilisé pour se connecter à la base de données et effectuer des opérations de lecture et d'écriture. Il fournit des classes et des méthodes pour interagir avec la base de données.