Projet : Plantes versus Zombies

Objectif :

Une application frontend vous sera fournie. Cette application vous permettra de jouer au jeu, ainsi que de personnaliser vos zombies, vos plantes et vos terrains. Cependant, pour jouer au jeu… vous devrez réaliser l'application backend qui permettra à votre jeu de fonctionner !

Développer une API RESTful backend complète permettant la gestion des données des plantes, des zombies et de la carte via une interface CRUD (Create, Read, Update, Delete). Ce projet mettra en pratique les concepts de Spring MVC, l'architecture en couches, l'utilisation de JdbcTemplate pour l'accès à la base de données MySQL, ainsi que l'intégration de Jakarta.

Technologies requises :

- **Langage :** Java 21
- **Framework :** Spring MVC
- **Gestion des dépendances :** Maven
- **Serveur d'application :** Tomcat
- **Base de données :** MySQL
- **Outil d'accès aux données :** JdbcTemplate
- **API Servlet :** Jakarta Servlet API
- **Format de données :** JSON

Environnement de développement :

- IDE : IntelliJ IDEA ou Visual Studio Code

Installations et configurations :

N'installez que les logiciels que vous n'avez pas déjà installés, bien entendu.

Il existe certainement des extensions dans vos différents IDE pour ces technologies, libre à vous de les utiliser. Cependant c'est aussi vous exposer à des comportements différents.

1. **Java :**
    - Téléchargement des sources :

      Sur le site d'Oracle, téléchargez les sources de Java 21.

      https://www.oracle.com/fr/java/technologies/downloads/#jdk21-windows

      Ensuite, extrayez les sources à l'emplacement de votre choix.

    - Configuration des variables d'environnement :

Pour que votre système reconnaisse Java, vous allez devoir configurer deux variables d'environnement JAVA_HOME et PATH.

JAVA_HOME : Définissez JAVA_HOME à l'endroit où vous avez placé vos sources java. Par exemple, si vous avez placé java à C:\Program Files\Java, définissez JAVA_HOME à C:\Program Files\Java\jdk-21.0.6.

PATH : Normalement, la variable PATH est déjà définie. Donc trouvez-la et ajoutez-lui une valeur. Il faut lui ajouter le chemin vers les fichiers binaires de Java. Donc %JAVA_HOME%\bin. %JAVA_HOME% fait référence à la variable JAVA_HOME.

- Vérification de l'installation :

  $> java -version

  Normalement, vous verrez la version de Java que vous avez installée.

2. **MySQL**

   - Téléchargement des sources :

     Rendez-vous sur le site officiel de MySQL : MySQL Downloads.

     Téléchargez l'installeur MySQL pour Windows et lancez-le. Il faut installer MySQL server. Normalement, vous n'aurez pas besoin de changer les valeurs d'installation par défaut.

   - Configuration des variables d'environnement :

     Pour pouvoir lancer MySQL depuis votre terminal, vous allez devoir configurer la variable d'environnement PATH. Normalement, vous verrez qu'elle existe déjà dans la liste de vos variables d'environnement. Vous devrez la définir sur le chemin de répertoire bin de l'endroit où vous avez installé MySQL. Donc, si vous avez laissé le chemin par défaut pendant l'installation :\Program Files\MySQL\MySQL Server 8.0\bin.

   - Vérification de l'installation :

     Tapez la commande suivante pour vérifier que MySQL est correctement installé et reconnu par votre système :

     $> mysql --version

     Vous devriez voir une sortie indiquant la version de MySQL installée.

   - Connexion à MySQL :

     Utilisez la commande suivante pour vous connecter à MySQL en tant qu'utilisateur root :

$> mysql -u root -p

Entrez le mot de passe root que vous avez défini lors de l'installation. Vous devriez entrer dans la console de MySQL.

- Créer la base de données :

  CREATE DATABASE pvz;

  Pour vérifier que la base de données est bien créée :

  SHOW DATABASES;

  Si votre base de données apparaît bien dans la liste, c’est bon.

  Maintenant, à chaque fois que vous vous connecterez à MySQL pour utiliser votre base de données, vous ferez :

  USE pvz;

- Création d’un nouvel utilisateur :

  Il n’est pas recommandé d’utiliser l’utilisateur root au quotidien, ses droits sont trop importants : une erreur de manipulation et votre installation MySQL serait cassée.

  Pour créer l’utilisateur, faites :

  CREATE USER epf@'localhost' IDENTIFIED BY 'mot_de_passe';

  Un utilisateur ‘epf’ sera créé.

  Maintenant, pour lui donner les droits de travailler sur votre bdd pvz :

  GRANT ALL PRIVILEGES ON pvz.* TO epf@'localhost';

  Pour que les modifications prennent effet :

  FLUSH PRIVILEGES;

  Pour vérifier que l’utilisateur à bien été créé :

  SELECT User, Host FROM mysql.user;

  Il devrait apparaître dans la liste.

  Pour vous connecter à cet utilisateur, vous devez d’abord vous sortir de votre commande MySQL :

  quit;

  Et pour vous connecter :

  $> mysql -u epf -p

- Remplir la base de données :

  On vous fournit deux scripts pour générer la base de données.

  init.sql : va s'occuper de créer toutes les tables.

  values.sql : va remplir la base de données.

  Pour lancer les scripts, entrez dans votre console MySQL et lancez :

  $> SOURCE ./chemin/vers/le/script

3. **Maven**

   - Téléchargement des sources :

     Rendez-vous sur le site officiel de Maven : [Apache Maven Download]

     Téléchargez l'archive binaire.

     Extrayez l'archive à l'emplacement de votre choix, par exemple C:\Program Files\Apache\maven.

   - Configuration des variables d'environnement :

     Pour que votre système reconnaisse Maven, vous allez devoir configurer deux variables d'environnement : MAVEN_HOME et PATH.

     MAVEN_HOME : Définissez MAVEN_HOME à l'endroit où vous avez extrait Maven. Par exemple, si vous avez extrait Maven dans C:\Program Files\Apache, définissez MAVEN_HOME à C:\Program Files\Apache\maven.

     PATH : Normalement, la variable PATH est déjà définie. Trouvez-la et ajoutez-lui une valeur. Il faut lui ajouter le chemin vers les fichiers binaires de Maven. Donc %MAVEN_HOME%\bin. %MAVEN_HOME% fait référence à la variable MAVEN_HOME.

   - Vérification de l'installation :

     Ouvrez un terminal et tapez la commande suivante pour vérifier que Maven est correctement installé et reconnu par votre système :

     $> mvn -version

     Normalement, vous verrez la version de Maven que vous avez installée, ainsi que la version de Java utilisée.

   - Compiler le projet :

     $> mvn clean install

Cela va générer un répertoire target et, dans ce répertoire, vous trouverez un fichier .war qui sera votre projet compilé.

4. **Tomcat**
    - Téléchargement des sources :

      Rendez-vous sur le site officiel de Tomcat : Apache Tomcat Download

      Téléchargez l'archive binaire (par exemple, `apache-tomcat-10.1.35.zip`).

      Extrayez l'archive à l'emplacement de votre choix, par exemple `C:\Program Files\Apache\tomcat`.

    - Configuration des variables d'environnement :

      Pour que votre système reconnaisse Tomcat, vous allez devoir configurer deux variables d'environnement : CATALINA_HOME et PATH.

      CATALINA_HOME : Définissez CATALINA_HOME à l'endroit où vous avez extrait Tomcat. Par exemple, si vous avez extrait Tomcat dans C:\Program Files\Apache, définissez CATALINA_HOME à C:\Program Files\Apache\tomcat.

      PATH : Normalement, la variable PATH est déjà définie. Trouvez-la et ajoutez-lui une valeur. Il faut lui ajouter le chemin vers les fichiers binaires de Tomcat. Donc %CATALINA_HOME%\bin. %CATALINA_HOME% fait référence à la variable CATALINA_HOME.

    - Vérification de l'installation :

      Ouvrez un terminal et tapez la commande suivante pour vérifier que Tomcat est correctement installé et reconnu par votre système :

      $> %CATALINA_HOME%/bin/version.bat

      Normalement, vous verrez la version de Tomcat que vous avez installée.

    - Démarrage et arrêt de Tomcat :

      Démarrer Tomcat :

      $> %CATALINA_HOME%/bin/startup.bat

      Arrêter Tomcat :

      $> %CATALINA_HOME%/bin/shutdown.bat

      Pour vérifier que votre Tomcat a bien démarré http://localhost:8080/.

      Si une page apparaît, Tomcat est bien lancé.

- Démarrer Tomcat avec votre projet :

  Récupérez le .war généré par Maven et placez-le dans le dossier webapps de Tomcat. Ensuite, quand vous démarrez Tomcat, il va générer un dossier dans webapps du même nom que votre .war.

Consignes du Projet :

Configuration de l'environnement et du projet

1. **Création de la structure du projet (architecture en couches) :**

   La première étape en commençant un projet Java est de réfléchir à l'architecture de ce projet. Ici, on vous demandera de faire une architecture en couches. Pensez aussi à séparer vos configurations du reste de votre code.

   Tous vos packages devront partir de com.oxyl.coursepfback.

Connexion à la Base de Données et Implémentation des Repositories

Pensez bien à écrire des **tests** pour **chacune** des classes que vous allez créer !

1. **Configuration de la connexion à la base de données (JdbcTemplate) :**

   Normalement, vous avez configuré votre base de données MySQL et l'avez peuplée. Maintenant, il faut relier votre application à cette base de données.

   Pour cela, créez une classe dédiée à votre configuration de base de données. Dans cette classe, vous devrez définir deux Beans, un pour votre DataSource et un pour votre JdbcTemplate.

2. **Création des Classes de Model :**

   Il est maintenant temps de réfléchir aux objets que votre application va manipuler. Dans notre cas, il y a trois objets : les zombies, les maps et les plantes. Ces objets devront correspondre aux objets que vous récupérez dans votre base de données.

3. **Création des Interfaces de DAO :**

   Il est toujours une bonne pratique de définir des interfaces pour les DAO. Le principal avantage est que si jamais votre projet venait à changer de base de données, une nouvelle implémentation de votre DAO respecterait toujours votre contrat. De façon plus globale, dans chaque couche du projet, il est fortement conseillé de faire des interfaces pour la maintenabilité et la flexibilité du code.

   Déclarez les méthodes CRUD pour chaque entité que vous récupérez de la base de données.

   Si vous avez regardé les champs présents dans les différentes tables de la base de données, vous aurez remarqué que la table des zombies est liée à la table des

maps. Vous devrez aussi déclarer une méthode qui va récupérer les zombies correspondant à une carte.

4. **Implémentation des DAO avec JdbcTemplate :**

   Maintenant que vous avez votre contrat dans vos interfaces, définissez les méthodes dans des classes de DAO.

   Pour faire vos requêtes, utilisez JdbcTemplate et faites attention aux risques d'injection SQL qui pourraient arriver.

Implémentation des Services et des Controllers

1. **Création des Interfaces de Service :**

   Créez des interfaces services pour chacun des DAO que vous avez créés. Déclarez les méthodes qui feront appel aux méthodes des DAO.

Implémentation des Services :

Maintenant, faites les implémentations de vos interfaces. Ce projet ne contient pas beaucoup de logique métier, les implémentations seront donc très simples, mais en général, les logiques métiers sont cruciales dans un projet back.

Gardez bien à l'esprit qu'un service prend en argument un modèle et renvoie un modèle.

Création des Controllers Spring MVC :

Nous sommes arrivés à notre dernière couche ! Vous devrez ici définir tous les endpoints nécessaires pour que l'application front fonctionne correctement. C'est dans cette couche qu'on envoie les bons codes HTTP et le bon verbe HTTP en plus des données qu'on doit lui transmettre.

Un controller ne renvoie jamais un model, on passe toujours par des DTO, pour maîtriser les informations qui passent du back au front.

Pour aller plus loin (optionnel) :

- **Gestion des exceptions :** Implémenter une gestion centralisée des exceptions pour gérer les erreurs de manière cohérente dans toute l'application.
- **Validation des données :** Spring propose des comportements avancés de validations de données, mettez les en place dans ce projet.
- **Documentation :** Documenter l'API RESTful en utilisant Swagger ou OpenAPI.
- **Sécurité :** Implémenter une authentification de base pour sécuriser l'API.
- **Logging :** Utiliser un framework de logging (par exemple, Logback) pour enregistrer les événements importants de l'application.

Critères d'évaluation :

- Respect des exigences techniques et fonctionnelles.
- Qualité du code (lisibilité, maintenabilité, documentation).

- Couverture des tests unitaires.
- Respect des principes de l'architecture en couches.
- Utilisation correcte des technologies requises (Spring MVC, JdbcTemplate, MySQL, Maven, Tomcat, Jakarta).
- Bonus pour les points optionnels ou toute autre proposition enrichissante pour le projet.