

Structure du Projet de Gestion des Employés

Fatim ezzahrae AKNIZ

9 décembre 2024

Structure du Dossier

Le dossier se compose de cinq sections principales : **Model**, **DAO**, **Controller**, **View** et **Main**.

1. Model

Cette section contient la classe **Employe**, qui représente la structure des données relatives à un employé. Les principales caractéristiques de cette classe sont :

- Les **constructeurs**, permettant de créer des objets **Employe** avec différents ensembles de paramètres.
- Les **énumérations** pour définir les rôles et les postes, garantissant que seules des valeurs valides peuvent être utilisées.
- Les **getters** pour accéder aux propriétés de la classe, conformément aux principes de l'encapsulation.

2. DAO (Data Access Object)

Cette section contient les classes et l'interface nécessaires pour interagir avec la base de données. Elle est composée des éléments suivants :

2.1 Classe DBConnection

Cette classe contient la méthode qui établit une connexion avec la base de données. Elle utilise JDBC pour assurer une connexion fiable et sécurisée. Les paramètres de connexion, tels que l'URL, le nom d'utilisateur et le mot de passe, sont configurés dans cette classe.

2.2 Interface EmployeeDAOI

Cette interface définit les signatures des méthodes nécessaires pour les opérations CRUD (Create, Read, Update, Delete) et autres actions. Les méthodes incluent :

- **Add()** : Ajouter un employé.
- **Update()** : Mettre à jour les informations d'un employé.
- **Delete()** : Supprimer un employé.
- **findAll()** : Récupérer une liste de tous les employés.
- **FindById()** : Trouver un employé en fonction de son ID.

2.3 Classe EmployeeDAOImpl

Cette classe implémente l'interface **EmployeeDAOI**. Elle contient le corps des méthodes déclarées dans l'interface, ainsi que d'autres méthodes pour interagir avec la base de données, telles que :

- **getEmployees()** : Récupère tous les employés avec leurs informations.
- **getRoles()** : Récupère la liste des rôles disponibles.
- **getPostes()** : Récupère la liste des postes disponibles.

3. View

Cette section correspond à l'interface utilisateur. Elle inclut :

- Les **champs de saisie** pour entrer les informations relatives aux employés.
- Une **table des employés** affichant les données sous forme tabulaire.
- Les **boutons** pour les actions telles que Ajouter, Modifier, Supprimer, Afficher et Trouver.
- Une méthode pour afficher un panneau secondaire demandant l'ID d'un employé lors de l'utilisation du bouton "Trouver".

De plus, cette classe contient des getters pour accéder aux boutons et champs de texte, facilitant ainsi l'interaction avec le contrôleur.

4. Controller

Cette section agit comme un pont entre le DAO et la vue. Elle inclut les fonctionnalités suivantes :

- Ajout de **listeners d'action** (`ActionListener`) aux boutons de l'interface utilisateur.
- Récupération des données saisies par l'utilisateur dans les champs et leur transmission en tant que paramètres aux méthodes appropriées du DAO.
- Mise à jour de l'interface utilisateur pour refléter les modifications après chaque opération.
- Affichage de **messages d'information** ou d'erreur à l'utilisateur en fonction du résultat de son action.

5. Main

Le point d'entrée de l'application. Il initialise la connexion avec la base de données, crée les instances nécessaires des classes DAO, View et Controller, et démarre l'application.

Conclusion

Cette structure modulaire permet une séparation claire des responsabilités, facilitant la maintenance, l'ajout de nouvelles fonctionnalités et le test de l'application. Grâce à cette organisation, l'application garantit une expérience utilisateur fluide et une interaction efficace avec la base de données.