

Compte Rendu du TP1

Idriss Chrigui

December 12, 2024

Table des Matières

1	Objectif	3
2	Résumé des Travaux	3
3	Implémentation du Code	3
3.1	Classe Principale	3
3.2	DAO	4
3.2.1	Interface DAO	4
3.2.2	Implémentation DAO	4
3.3	Modèle (Model)	5
3.4	Contrôleur (Controller)	5
3.5	Vue (View)	6
4	Résultats	6
5	Défis et Solutions	6
6	Conclusion	6

1 Objectif

L'objectif de cette session pratique était de concevoir et d'implémenter un système de gestion des employés en Java, basé sur une architecture à plusieurs couches, incluant des modèles, des vues, des contrôleurs et des interactions avec une base de données.

2 Résumé des Travaux

Le travail assigné comprenait la création des composants suivants :

- **Modèle (Model)** : Une représentation des données des employés comprenant des attributs tels que le nom, l'email, le téléphone, le salaire, le poste et le rôle.
- **DAO (Data Access Object)** : Une couche permettant d'interagir avec la base de données PostgreSQL à l'aide de requêtes SQL.
- **Contrôleur (Controller)** : Une couche gérant la logique pour ajouter, supprimer, mettre à jour et afficher les employés.
- **Vue (View)** : Une interface graphique (GUI) permettant les interactions utilisateur.

3 Implémentation du Code

3.1 Classe Principale

```
1 import Controllers.EmployeeController;
2 import DAO.EmployeeDAOImpl;
3 import Views.EmployeeView;
4
5 public class Main {
6     public static void main(String[] args) {
7         // Init the database connection
8         EmployeeDAOImpl dao = new EmployeeDAOImpl();
9
10        // Render the View
11        EmployeeView ev = new EmployeeView();
12
13        // Add controller for the view
14        EmployeeController ec = new EmployeeController();
15    }
16 }
```

3.2 DAO

3.2.1 Interface DAO

```
1 package DAO;
2
3 interface EmployeeDAOI {
4     // Credentials
5     public String url = "jdbc:mysql://localhost:3306/java_db";
6     public String dbuser = "root";    // Database user
7     public String dbpw = "";          // Database password
8
9     // Abstract methods
10    public boolean addEmployee(Employee em);
11    public boolean deleteEmployee(int id);
12    public boolean updateEmployee(int id, Employee em);
13 }
```

3.2.2 Implémentation DAO

```
1 package DAO;
2
3 public class EmployeeDAOImpl implements EmployeeDAOI {
4     // Constructor
5     public EmployeeDAOImpl();
6
7     // Methods to be override here
8 }
```

3.3 Modèle (Model)

```
1 package Models;
2
3 public class Employee {
4     // Constructor
5     public Employee(ResultSet rs);
6
7     // Getters
8     public int getId();
9     public String getLname();
10    public String getFname();
11    public String getEmail();
12    public double getSalary();
13    public String getPhone();
14    public String getPost();
15    public String getRole();
16
17    // Setters
18    public void setId(int id);
19    public void setLname(String lname);
20    public void setFname(String fname);
21    public void setEmail(String email);
22    public void setSalary(double salary);
23    public void setPhone(String phone);
24    public void setPost(String post);
25    public void setRole(String role);
26
27    // methods for Controller interactions
28    public boolean addEmployee();
29    public static boolean deleteEmployee(int id);
30    public boolean updateEmployee(int id);
31
32    public String toString();
33 }
```

3.4 Contrôleur (Controller)

```
1 package Controllers;
2
3 public class EmployeeController {
4     // Constructor
5     public EmployeeController();
6
7     // Event listeners initialization methods
8     private void initAddEvent();
9     private void initDeleteEvent();
10    private void initUpdateEvent();
11    private void initShowEvent();
12
13    // Useful View handling methods
14    public static void populateTable();
15    public static void emptyFields();
16 }
```

3.5 Vue (View)

```
1 package Views;
2
3 public class EmployeeView extends JFrame {
4     // Constructor
5     public EmployeeView();
6 }
```

4 Résultats

L'application a été testée sous Ubuntu avec PostgreSQL comme système de gestion de base de données.

L'implémentation a permis de démontrer les fonctionnalités suivantes :

- Établir une connexion à une base de données PostgreSQL.
- Ajouter, supprimer et mettre à jour les enregistrements des employés via des interactions avec l'interface graphique.
- Afficher les données des employés dans un tableau au sein de l'interface graphique.

5 Défis et Solutions

- **Défi** : Gestion des exceptions SQL lors des opérations sur la base de données.
- **Solution** : Ajout de blocs try-catch et de messages d'erreur appropriés.
- **Défi** : Synchronisation des mises à jour de l'interface avec les modifications dans la base de données.
- **Solution** : Implémentation d'une méthode pour actualiser dynamiquement le tableau après chaque opération.

6 Conclusion

Cette session a permis d'acquérir une expérience pratique dans la création d'une application Java complète avec une interface graphique et une intégration à une base de données. Elle a renforcé la compréhension de l'architecture MVC et des interactions avec une base de données SQL.