

ASSIGNMENT

Name: Kajal K J

Full Stack Development JAVA

24-07-2024

DAO (Data Access Object)

DAO stands for Data Access Object. DAO Design Pattern is used to separate the data persistence logic in a separate layer. This way, the service remains completely in dark about how the low-level operations to access the database is done. This is known as the principle of **Separation of Logic**.

Task1

Create a registration module with database connectivity to store data in a database

Task2

Create a login module with database connectivity to check authentication of user

Creating a Registration and Login System with Java (JDBC)

Step1: Create a New Java Project

1. Open NetBeans/Eclipses/Intelij.
2. Go to File -> New Project.
3. Select Java -> Java Application.
4. Click Next.
5. Enter the project name (e.g., RegistrationLogin) and location.
6. Click Finish.

Step2: Set Up the Project Structure

1. Right-click on the Source Packages directory in the Project Explorer.

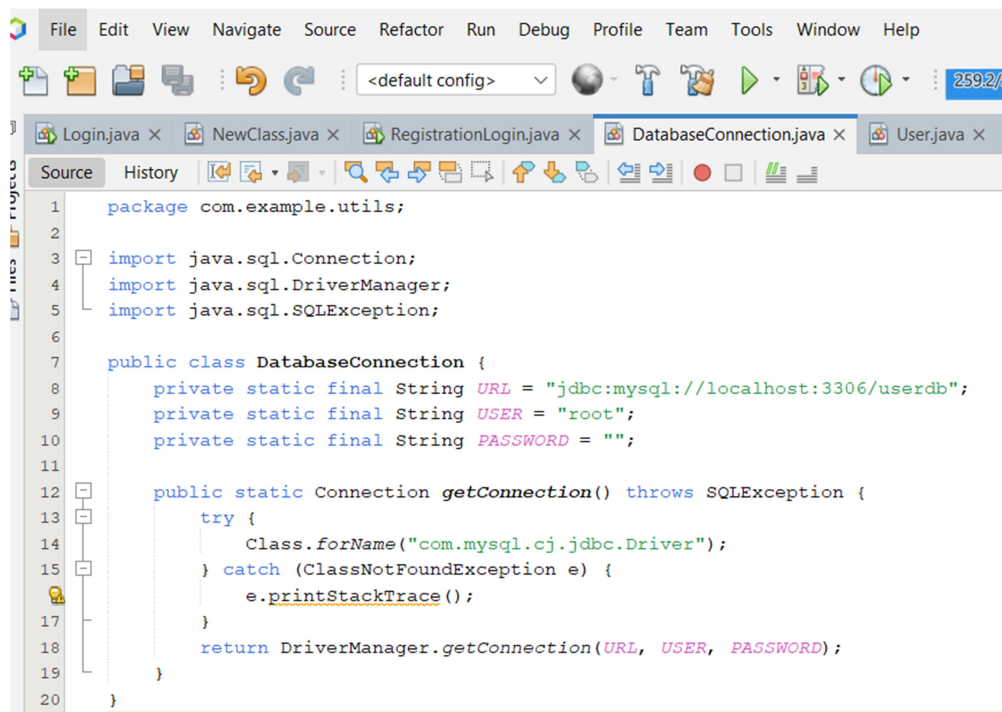
2. Select New -> Java Package.

Step3: Add MySQL Connector/J to the Project

1. Right-click on the project name in the Project Explorer.
2. Select Properties.
3. Go to Libraries -> Compile.
4. Click Add JAR/Folder and select the downloaded MySQL Connector/J JAR file.
5. Click Open and then OK.

Step4: Create Java Classes

1. DatabaseConnection.java



```
1 package com.example.utils;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DatabaseConnection {
8     private static final String URL = "jdbc:mysql://localhost:3306/userdb";
9     private static final String USER = "root";
10    private static final String PASSWORD = "";
11
12    public static Connection getConnection() throws SQLException {
13        try {
14            Class.forName("com.mysql.cj.jdbc.Driver");
15        } catch (ClassNotFoundException e) {
16            e.printStackTrace();
17        }
18        return DriverManager.getConnection(URL, USER, PASSWORD);
19    }
20 }
```

2.User.java



```
1 package com.example.model;
2
3 public class User {
4     private int id;
5     private String name;
6     private String email;
7     private String password;
8
9     // Getters and Setters
10    public int getId() {
11        return id;
12    }
13
14    public void setId(int id) {
15        this.id = id;
16    }
17
18    public String getName() {
19        return name;
20    }
21
22    public void setName(String name) {
23        this.name = name;
24    }
25
26    public String getEmail() {
27        return email;
28    }
29
30    public void setEmail(String email) {
31        this.email = email;
32    }
33
34    public String getPassword() {
35        return password;
36    }
37 }
```

2. UserDao.java

```
package com.example.dao;

import com.example.model.User;
import com.example.utils.DatabaseConnection;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
public class UserDao {  
    public boolean registerUser(User user) {  
        String query = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";  
        try (Connection conn = DatabaseConnection.getConnection();  
            PreparedStatement ps = conn.prepareStatement(query)) {  
            ps.setString(1, user.getName());  
            ps.setString(2, user.getEmail());  
            ps.setString(3, user.getPassword());  
            int result = ps.executeUpdate();  
            return result > 0;  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
        return false;  
    }  
  
    public User loginUser(String email, String password) {  
        String query = "SELECT * FROM users WHERE email = ? AND password = ?";  
        try (Connection conn = DatabaseConnection.getConnection();  
            PreparedStatement ps = conn.prepareStatement(query)) {  
            ps.setString(1, email);  
            ps.setString(2, password);  
            try (ResultSet rs = ps.executeQuery()) {  
                if (rs.next()) {  
                    User user = new User();  
                    user.setId(rs.getInt("id"));  
                    user.setName(rs.getString("name"));  
                    user.setEmail(rs.getString("email"));  
                    user.setPassword(rs.getString("password"));  
                    return user;  
                }  
            }  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
        return null;
    }
}
```

3. Main.java

```
package com.example.main;

import com.example.dao.UserDao;
import com.example.model.User;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        System.out.println("Starting the application...");

        UserDao userDao = new UserDao();

        Scanner scanner = new Scanner(System.in);

        while (true) {

            System.out.println("1. Register");
            System.out.println("2. Login");
            System.out.println("3. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            if (choice == 1) {

                System.out.println("Registering a new user...");
                System.out.print("Enter Name: ");
                String name = scanner.nextLine();
                System.out.print("Enter Email: ");
                String email = scanner.nextLine();
```

```
System.out.print("Enter Password: ");
String password = scanner.nextLine();

User user = new User();
user.setName(name);
user.setEmail(email);
user.setPassword(password);

boolean isRegistered = userDao.registerUser(user);
if (isRegistered) {
    System.out.println("User registered successfully.");
} else {
    System.out.println("User registration failed.");
}
} else if (choice == 2) {
    System.out.println("Logging in a user...");
    System.out.print("Enter Email: ");
    String email = scanner.nextLine();
    System.out.print("Enter Password: ");
    String password = scanner.nextLine();

    User user = userDao.loginUser(email, password);
    if (user != null) {
        System.out.println("Login successful. Welcome, " +
user.getName() + "!");
    } else {
        System.out.println("Invalid email or password.");
    }
} else if (choice == 3) {
```

```
        System.out.println("Exiting...");  
        break;  
    } else {  
        System.out.println("Invalid option. Please try again.");  
    }  
}  
scanner.close();  
}
```


OUTPUT

```

run:
Starting the application...
1. Register
2. Login
3. Exit
Choose an option: 1
Registering a new user...
Enter Name: kajal
Enter Email: kajal@gmail.com
Enter Password: 345
User registered successfully.
*****
1. Register
2. Login
3. Exit
Choose an option: 2
Logging in a user...
Enter Email: kajal@gmail.com
Enter Password: 345
Login successful. Welcome, kajal!
*****
1. Register
2. Login
3. Exit
Choose an option: ||

```

✓ Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)

`SELECT * FROM `users``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

		id	name	email	password		
<input type="checkbox"/>	Edit	Copy	Delete	6	Aparna	aparna@gmail.com	123
<input type="checkbox"/>	Edit	Copy	Delete	7	Aparna S	aparnas@gmail.com	789
<input type="checkbox"/>	Edit	Copy	Delete	8	kajal	kajal@gmail.com	345

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations