**HibernateUtil Class:**  
  
package com.examly.hibernate.util;

import com.examly.hibernate.model.Employee;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static SessionFactory sessionFactory;

static {

try {

sessionFactory = new Configuration()

.configure("hibernate.cfg.xml") // Load settings from hibernate.cfg.xml

.addAnnotatedClass(Employee.class) // Register the Employee class

.buildSessionFactory();

} catch (Throwable ex) {

System.err.println("Initial SessionFactory creation failed." + ex);

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

public static void setSessionFactory(SessionFactory sessionFactory) {

HibernateUtil.sessionFactory = sessionFactory;

}

public static void shutdown() {

getSessionFactory().close();

}

}

**EmployeeDAO Class**

package com.examly.hibernate.dao;

import com.examly.hibernate.model.Employee;

import com.examly.hibernate.util.HibernateUtil;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.query.Query;

import java.util.Collections;

import java.util.List;

public class EmployeeDAO {

// Save Employee

public Employee saveEmployee(Employee employee) {

Transaction transaction = null;

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

transaction = session.beginTransaction();

session.save(employee);

transaction.commit();

return employee;

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

return null;

}

}

// Get Employee by ID

public Employee getEmployeeById(int id) {

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

return session.get(Employee.class, id);

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

// Update Employee

public Employee updateEmployee(Employee employee) {

Transaction transaction = null;

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

transaction = session.beginTransaction();

session.update(employee);

transaction.commit();

return employee;

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

return null;

}

}

// Delete Employee by ID

public String deleteEmployee(Employee employee) {

Transaction transaction = null;

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

transaction = session.beginTransaction();

session.delete(employee);

transaction.commit();

return "Employee deleted successfully";

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

return "Failed to delete Employee";

}

}

// Get All Employees

@SuppressWarnings("unchecked")

public List<Employee> getAllEmployee() {

try (Session session = HibernateUtil.getSessionFactory().openSession()) {

Query<Employee> query = session.createQuery("FROM Employee");

return query.getResultList();

} catch (Exception e) {

e.printStackTrace();

return Collections.emptyList();

}

}

}