CRUD Operations

Discuss the basic CRUD (Create, Read, Update, Delete) operations in Hibernate. Provide examples of each operation using Hibernate APIs (Session or EntityManager). Include saving an entity (session.save()), retrieving an entity by ID (session.get()), updating an entity (session.update()), and deleting an entity (session.delete()).

The basic CRUD (Create, Read, Update, Delete) operations in Hibernate using Hibernate APIs such as Session.

1. Create Operation (session.save())

The save method is used to save an entity instance to the database. When you call save, the entity instance is associated with the Hibernate Session and a new row is inserted into the database.

Example

import org.hibernate.Session;

import org.hibernate.Transaction;

public class HibernateCRUD {

public static void main(String[] args) {

Session session = HibernateUtil.getSessionFactory().openSession();

Transaction transaction = null;

try {

transaction = session.beginTransaction();

Employee employee = new Employee("John Doe", "IT", 5000.00);

session.save(employee);

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

session.close();

}

}

}

2. Read Operation (session.get())

The get method is used to retrieve an entity by its primary key. It returns the entity instance if it exists in the database; otherwise, it returns null.

Example

import org.hibernate.Session;

public class HibernateCRUD {

public static void main(String[] args) {

Session session = HibernateUtil.getSessionFactory().openSession();

try {

// Retrieve an Employee by its primary key (id)

Long employeeId = 1L;

Employee employee = session.get(Employee.class, employeeId);

if (employee != null) {

System.out.println("Employee found: " + employee.getName());

} else {

System.out.println("Employee not found");

}

} catch (Exception e) {

e.printStackTrace();

} finally {

session.close();

}

}

}

3. Update Operation (session.update())

The update method is used to update an existing entity in the database. The entity instance must be associated with the Hibernate Session for the changes to be persisted.

Example

import org.hibernate.Session;

import org.hibernate.Transaction;

public class HibernateCRUD {

public static void main(String[] args) {

Session session = HibernateUtil.getSessionFactory().openSession();

Transaction transaction = null;

try {

transaction = session.beginTransaction();

Long employeeId = 1L;

Employee employee = session.get(Employee.class, employeeId);

if (employee != null) {

// Update the employee's department

employee.setDepartment("HR");

employee.setSalary(6000.00);

session.update(employee);

transaction.commit();

} else {

System.out.println("Employee not found");

}

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

session.close();

}

}

}

4. Delete Operation (session.delete())

The delete method is used to remove an entity from the database. The entity instance must be associated with the Hibernate Session for the deletion to occur.

import org.hibernate.Session;

import org.hibernate.Transaction;

public class HibernateCRUD {

public static void main(String[] args) {

Session session = HibernateUtil.getSessionFactory().openSession();

Transaction transaction = null;

try {

transaction = session.beginTransaction();

Long employeeId = 1L;

Employee employee = session.get(Employee.class, employeeId);

if (employee != null) {

// Delete the employee from the database

session.delete(employee);

transaction.commit();

} else {

System.out.println("Employee not found");

}

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

session.close();

}

}

}

**Utility Class**

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory sessionFactory = buildSessionFactory();

private static SessionFactory buildSessionFactory() {

try {

// Create the SessionFactory from hibernate.cfg.xml

return new Configuration().configure().buildSessionFactory();

} catch (Throwable ex) {

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

public static void shutdown() {

// Close caches and connection pools

getSessionFactory().close();

}

}

1. Create Operation: Use session.save(entity) to save a new entity instance to the database.

2. Read Operation: Use session.get(EntityClass.class, id) to retrieve an entity by its primary key.

3. Update Operation: Use session.update(entity) to update an existing entity in the database.

4. Delete Operation: Use session.delete(entity) to remove an entity from the database.