

Acceso a Datos

Reto Hibernate – Eneko Olabarria

Cabecera con todos los imports y constante static Scanner para leer el input del usuario por consola. A continuación se encuentra el metodo main con las llamadas a SessionFactory y Session para comenzar a usar Hibernate para recorrer la base de datos.

```
6 package com.eneko;
7
8 import com.eneko.hibernate.HibernateUtil;
9 import com.eneko.hibernate.Libro;
10 import java.util.Iterator;
11 import java.util.List;
12 import java.util.Scanner;
13 import org.hibernate.Query;
14 import org.hibernate.Session;
15 import org.hibernate.SessionFactory;
16
17 /**
18  *
19  * @author eneko
20  */
21
22 public class Main {
23
24     private final static Scanner SC = new Scanner(System.in);
25
26     public static void main(String[] args) {
27         SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
28         Session session = sessionFactory.openSession();
29     }
```

Primera query que recoge todos los datos de la tabla Libro, la guarda en una lista que se convierte a Iterator y se recorre con un bucle while para ir mostrando por consola el titulo, isbn y nombre del autor de todos los elementos.

```
30     String hql = "FROM Libro";
31     Query query = session.createQuery(hql);
32
33     List<Libro> lista = query.list();
34     Iterator<Libro> iterator = lista.iterator();
35
36     while(iterator.hasNext()) {
37         Libro libro = (Libro) iterator.next();
38         System.out.printf(
39             "Titulo: %s\n\tISBN: %s\n\tAutor: %s\n",
40             libro.getTitulo(), libro.getIsbn(), libro.getAutor().getNombreAutor()
41         );
42     }
```

Segunda query donde primero se recogen los foreign keys que se emplearan en la query, tras esto se encuentra la query con los where que emplean las variables antes recogidas y usando el metodo setParameter se aplican, tras esto mismo procedimiento que en el primer caso creando una lista, convirtiendola en Iterator y recorriendo el mismo para mostrar cada elemento, en esta ocasión mostrando todos los campos de cada entrada.

Para finalizar se cierran los recursos empleados: Scanner, SessionFactory y Session comprobando si esta se ha quedado abierta previamente.

```
44      System.out.println("Id del autor:");
45      int autorId = SC.nextInt();
46      System.out.println("Id de la editorial:");
47      int editorialId = SC.nextInt();
48      System.out.println("Id del tema:");
49      int temaId = SC.nextInt();
50
51      hql = "SELECT l FROM Libro l " +
52           "JOIN l.autor a " +
53           "JOIN l.editorial e " +
54           "JOIN l.tema t " +
55           "WHERE a.idAutor = :nombreAutor " +
56           "AND e.idEditorial = :nombreEditorial " +
57           "AND t.idTema = :nombreTema";
58
59      query = session.createQuery(hql);
60      query.setParameter("nombreAutor", autorId);
61      query.setParameter("nombreEditorial", editorialId);
62      query.setParameter("nombreTema", temaId);
63
64      List<Libro> lista2 = query.list();
65      Iterator<Libro> iterator2 = lista2.iterator();
66
67      while(iterator2.hasNext()) {
68          Libro libro = (Libro) iterator2.next();
69          System.out.printf(
70              "Titulo: %s\n\t"
71              + "ISBN: %s\n\t"
72              + "Numero de Ejemplares: %d\n\t"
73              + "Autor: %s\n\t"
74              + "Editorial: %s\n\t"
75              + "Tema: %s\n",
76              libro.getTitulo(),
77              libro.getIsbn(),
78              libro.getNumeroEjemplares(),
79              libro.getAutor().getNombreAutor(),
80              libro.getEditorial().getNombreEditorial(),
81              libro.getTema().getNombreTema()
82          );
83      }
84      SC.close();
85      sessionFactory.close();
86      if (session.isConnected()) session.close();
```