Hibernate One to Many Example(How to use List objects in Hibernate)

If the persistent class has list object that contains the entity reference, we need to use one-to-many association to map the list element.

Here, we are using the scenario of Forum where one question has multiple answers.

there can be many answers for a question and each answer may have its own informations that is why we have used **list** in the persistent class (containing the reference of Answer class) to represent a collection of answers.

Let's see the persistent class that has list objects (containing Answer class objects).

|  |
| --- |
| **import** java.util.List;    **public** **class** Question {  **private** **int** id;  **private** String qname;  **private** List<Answer> answers;  //getters and setters   } |

The Answer class has its own informations such as id, answername, postedBy etc.

|  |
| --- |
| **public** **class** Answer {  **private** **int** id;  **private** String answername;  **private** String postedBy;  //getters and setters    } |

Mapping class

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8'?>  <!DOCTYPE hibernate-mapping PUBLIC            "-//Hibernate/Hibernate Mapping DTD 5.3//EN"            "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">              <hibernate-mapping>            <**class** name="com.abc.Question" table="q501">            <id name="id">             <generator **class**="increment"></generator>            </id>            <property name="qname"></property>              <list name="answers" cascade="all">            <key column="qid"></key>            <index column="type"></index>            <one-to-many **class**="com.abc.Answer"/>            </list>              </**class**>              <**class** name="com.abc.Answer" table="ans501">            <id name="id">             <generator **class**="increment"></generator>            </id>            <property name="answername"></property>            <property name="postedBy"></property>            </**class**>              </hibernate-mapping> |

Config File as same as earlier one

Main.java

|  |
| --- |
| Configuration cfg = **new** Configuration();  cfg.configure("Question.cfg.xml");    SessionFactory sf = cfg.buildSessionFactory();    Session s = sf.openSession();  Transaction t = s.beginTransaction();  Answer ans1=**new** Answer();  ans1.setAnswername("Java is a programming language");  ans1.setPostedBy("Ravi Malik");    Answer ans2=**new** Answer();  ans2.setAnswername("Java is a platform");  ans2.setPostedBy("Sudhir Kumar");    Answer ans3=**new** Answer();  ans3.setAnswername("Servlet is an Interface");  ans3.setPostedBy("Jai Kumar");    Answer ans4=**new** Answer();  ans4.setAnswername("Servlet is an API");  ans4.setPostedBy("Arun");    ArrayList<Answer> list1=**new** ArrayList<Answer>();  list1.add(ans1);  list1.add(ans2);    ArrayList<Answer> list2=**new** ArrayList<Answer>();  list2.add(ans3);  list2.add(ans4);    Question question1=**new** Question();  question1.setQname("What is Java?");  question1.setAnswers(list1);    Question question2=**new** Question();  question2.setQname("What is Servlet?");  question2.setAnswers(list2);  s.persist(question1);  s.persist(question2);  t.commit();  s.close();  sf.close();  System.***out***.println(" Inserted....."); |

### **How to fetch the data of List**

Here, we have used HQL to fetch all the records of Question class including answers. In such case, it fetches the data from two tables that are functional dependent. Here, we are direct printing the object of answer class, but we have overridden the **toString()** method in the Answer class returning answername and poster name. So it prints the answer name and postername rather than reference id.

|  |
| --- |
| Configuration cfg = new Configuration();  cfg.configure("Question.cfg.xml");  SessionFactory sf = cfg.buildSessionFactory();  Session s = sf.openSession();  TypedQuery query=session.createQuery("from Question");      List<Question> list=query.getResultList();        Iterator<Question> itr=list.iterator();  **while**(itr.hasNext()){          Question q=itr.next();          System.out.println("Question Name: "+q.getQname());            //printing answers          List<Answer> list2=q.getAnswers();          Iterator<Answer> itr2=list2.iterator();  **while**(itr2.hasNext())         {          Answer a=itr2.next();              System.out.println(a.getAnswername()+":"+a.getPostedBy());          }      }      session.close();      System.out.println("success"); |