Hibernate的集合映射(Set、List、Array、Map、Bag)

2012-2-17 13:58| 发布者: [benben](http://space.189works.com/space-uid-10034751.html)| 查看: 428| 评论: 0

摘要: Hibernate集合映射POJOs如下：Customer类------customer表 Order类对应----------orders表 customer(1)---------------(n)orderpublic class Customer{ private String id; private String username; private Strin ...

|  |
| --- |
| **Hibernate集合映射**  POJOs如下：  Customer类------>customer表   Order类对应---------->orders表 customer(1)<--------------->(n)order  public class Customer {  private String id;  private String username;  private String password;  private Timestamp registerTime;  private int age;  private Set<Order> orders = new HashSet<Order>();   /\*setter and getter method\*/ }  public class Order {  private String id;  private String orderNumber;  private int balance;  private Customer customer;   /\*setter and getter method\*/ }    **Set集合映射：**  Hibernate为集合映射提供了专用的标签元素，Set集合映射，就使用<set>标签表示：  <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">    <hibernate-mapping>  <class name="com.suxiaolei.hibernate.pojos.Customer" table="customer">  <!-- 主键设置 -->  <id name="id" type="string">  <column name="id"></column>  <generator class="uuid"></generator>  </id>    <!-- 属性设置 -->  <property name="username" column="username" type="string"></property>  <property name="password" column="password" type="string"></property>  <property name="age" column="age" type="integer"></property>  <property name="registerTime" column="register\_time" type="timestamp"></property>    <set name="orders" inverse="true" cascade="all">  <key column="customer\_id"></key>  <one-to-many class="com.suxiaolei.hibernate.pojos.Order"/>  </set>    </class>  </hibernate-mapping>  　　<set>标签中的"name"属性表示customer对象中关系集合的属性名，"inverse"与"cascade"属性说明（参考这里）。在数据库中表示"一对多"的关系是通过外键关联的方式实现的，"多方"通过持有"一方"的主键值来确定关系，怎么持有"一方"的主键值？"多方"将使用一列来存储"一方"的主键值，然后将此列作为外键列参照"一方"的主键列。所以使用Hibernate开发时需要将两表的关系列（外键列）告诉Hibernate，<key column="customer\_id"></key>就是完成这个工作的，Hibernate就能根据 "customer\_id"列取出关联信息。例如：从customer表中取出一条记录后，Hibernate会根据该customer记录的主键值再从order表中查找"custom\_id"列，取出值相等的记录，然后组装到Customer对象中的set集合属性中，反之亦然。因为取出来的记录（只是一些零碎的值，还没有组装成对象）需要存放到Set集合中，所以要告诉Hibernate在Set集合里面能放什么类型的数据。<one-to-many>这个标签就是完成这个工作的，"class"属性是指定这个这个Set集合里面元素的类型。  <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">    <hibernate-mapping>  <class name="com.suxiaolei.hibernate.pojos.Order" table="orders">  <id name="id" type="string">  <column name="id"></column>  <generator class="uuid"></generator>  </id>    <property name="orderNumber" column="orderNumber" type="string"></property>  <property name="balance" column="balance" type="integer"></property>    <many-to-one name="customer" class="com.suxiaolei.hibernate.pojos.Customer">  <column name="customer\_id"></column>  </many-to-one>   </class>  </hibernate-mapping>  　　<many-to-one>标签是设置"一对多"关系中的"多方"的，name指定了哪一个属性是关系属性，"class"指定了关系属性的类型（也指定了与哪一个表关联）， "column"属性是指定这个关联属性是按照"customer\_id"列的值，在customer表中查询获得的。  测试：  tx = session.beginTransaction();    /\*  \* 创建Customer对象，并设置其属性值  \*/  Customer customer = new Customer();  customer.setUsername("zhangsan");  customer.setPassword("123456");  customer.setAge(22);  customer.setRegisterTime(new Timestamp(new Date().getTime()));    /\*  \* 创建Order对象order1，并设置其属性值  \*/  Order order1 = new Order();  order1.setOrderNumber("a1a2a3");  order1.setBalance(1000);  order1.setCustomer(customer);    /\*  \* 创建Order对象order2，并设置其属性值  \*/  Order order2 = new Order();  order2.setOrderNumber("d3d2d1");  order2.setBalance(670);  order2.setCustomer(customer);    customer.getOrders().add(order1);  customer.getOrders().add(order2);    session.saveOrUpdate(customer);    tx.commit();  查看数据库的数据：  customer表：  http://www.189works.com/data/attachment/portal/et2/201202/ET24255201202171338551.png  orders表  http://www.189works.com/data/attachment/portal/et2/201202/ET24255201202171338552.png  可以看到数据被成功的插入到数据库中了，并且"custom\_id"列（关系列）也正确赋值了。  **List映射：**  Hibernate为集合映射提供了专用的标签元素，List集合映射，就使用<list>标签表示：  <list name="orders" inverse="false" cascade="all">  <key column="customer\_id"></key>  <index column="customer\_index" type="integer"></index>  <one-to-many class="com.suxiaolei.hibernate.pojos.Order"/>  </list>   　　List集合是有顺序的，"index"标签，是用于记录顺序，List的顺序将表现在"customer\_index"列上，其余设置，与Set集合类似。**注意：**List映射中"inverse"中的值不能设置为"true"，因为List集合的顺序只有customer方知道，order方不知道List的存在。不然，"customer\_index"的列值将不会被赋值。  查看数据库：  customer表：  http://www.189works.com/data/attachment/portal/et2/201202/ET24255201202171338553.png  orders表：  http://www.189works.com/data/attachment/portal/et2/201202/ET24255201202171338554.png  可以看到记录正确的插入到数据库中了，而且"custom\_index"正确的表示出List的顺序。  **Array（数组）映射：标签使用<array>其他与List基本一致。**    **Map映射：**  Hibernate为集合映射提供了专用的标签元素，Map集合映射，就使用<map>标签表示：    <map name="orders" inverse="false" cascade="all">  <key column="customer\_id"></key>  <index column="order\_key" type="string"></index>  <one-to-many class="com.suxiaolei.hibernate.pojos.Order"/>  </map>  　　Map映射中<index>标签，表示Map集合中的key值，记录在"order\_key"列中，<one-to-many>表示Map集合中的vlaue。其他设置与上面一样。**注意**："inverse"不要设置成"true"因为key值是customer对象维护的，而order不知道key的存在。  **Bag映射：它是List与Set集合的结合，可以重复，但是无顺。使用List模拟Bag。设置类似Set，它也有专用标签<bag>。**    **总结**  　　在没有特殊要求下，最好使用Set集合，因为Set集合没有特殊信息需要"一方"自己维护，可以完全交给"多方"维护，能够提高性能，若需要记录数据的顺序可以使用List和Array映射，若需要key/value形式存储数据，可以使用Map映射。最后一点若集合放置的数据简单类型(原生类型、原生类型的包装类、String、Date之类的)在集合映射配置上稍有不同，<element>元素可以直接映射这些简单类型，其他配置与上述配置没什么不同。 |