Hibernate数据过滤的使用

Hibernate3 提供了一种创新的方式来处理具有“显性(visibility)”规则的数据，那就是使用hibernate filter。 Hibernate filter是全局有效的、具有名字、可以带参数的过滤器， 对于某个特定的Hibernate session您可以选择是否启用（或禁用）某个过滤器。

一旦启用了数据过滤器，则不管是数据查询，还是数据加载，该过滤器将自动作用于所有数据，只有满足过滤条件的记录才会被选出来。

过滤器条件相当于定义一个 非常类似于类和各种集合上的“where”属性的约束子句，但是过滤器条件可以带参数。 应用程序可以在运行时决定是否启用给定的过滤器，以及使用什么样的参数值。 过滤器的用法很像数据库视图，只不过是在应用程序中确定使用什么样的参数的。

过滤器分以下几步：

## 定义过滤器

要使用过滤器，必须首先在相应的映射节点中定义。而定义一个过滤器，要用到位于<hibernate-mapping/> 节点之内的<filter-def/>节点：

|  |
| --- |
| <filter-def name="myFilter">  <filter-param name="myFilterParam" type="string"/>  </filter-def> |

## 使用过滤器。

定义好之后，就可以在某个类中使用这个过滤器。通过<filter.../>元素将指定的过滤器应用到指定的持久化类

|  |
| --- |
| <class name="myClass" ...>  <filter name="myFilter" condition=":myFilterParam =MY\_FILTERED\_COLUMN"/>  </class>  Condition属性值是一个SQL风格的where子句，因此condition属性所指定的过滤条件应该根据表明、列名进行过滤。也可以在某个集合使用它：  <set ...>  <filter name="myFilter" condition=":myFilterParam = MY\_FILTERED\_COLUMN"/>  </set>  可以在多个类或集合中使用某个过滤器；某个类或者集合中也可以使用多个过滤器。 |

## 在程序中通过Session启用过滤器

Session对象中会用到的方法有：enableFilter(String filterName), getEnabledFilter(String filterName), 和 disableFilter(String filterName). Session中默认是不启用过滤器的，必须通过Session.enabledFilter()方法显式的启用。 该方法返回被启用的Filter的实例。以上文定义的过滤器为例：

session.enableFilter("myFilter").setParameter("myFilterParam", "some-value");

## 完整的实例

|  |
| --- |
| 过滤器的配置文件  <hibernate-mapping package="com.hibernate.domain">  <!-- 映射Person持久化类 -->  <class name="Person" table="person">  <!-- 映射标识属性 -->  <id name="id" column="person\_id">  <!-- 指定identity的主键生成策略 -->  <generator class="identity"/>  </id>  <!-- 映射普通属性 -->  <property name="name" type="string"/>  <property name="age" type="int"/>    <!-- 使用数据过滤 -->  <filter name="myFilter" condition=":filterparam = age" />  </class>    <!--定义Filter -->  <filter-def name="myFilter">  <filter-param name="filterparam" type="int"/>  </filter-def>  </hibernate-mapping> |
| 示例程序  public class PersonManager {  public static void main(String[] args) {  Session session = HibernateUtil.getSession();  Transaction tx = session.beginTransaction();    //启动myFilter过滤器，并且设置参数    session.enableFilter("myFilter").setParameter("filterparam", 30);  //查询person实体，不加任何筛选条件  List list = session.createQuery("from Person as p").list();  for (Iterator iterator=list.iterator();iterator.hasNext();) {  Person person = (Person) iterator.next();  System.out.println(person.getName());  }  }  }  一般来说如果某个筛选条件使用的很频繁，那么我们可以将该筛选条件设置为过滤器；如果是临时的数据筛选，还是使用常规的查询比较好。 |