

Hibernate 数据检索



Hibernate

Hibernate查询语言——HQL

讲师：李兴华

JAVA培训报名热线：010-51283346

.IT.电子教育门户.

© MLDN INC.

更多学习资源尽在

www.MLDN.cn

本章内容

- ◆ 了解Criteria Query的使用
- ◆ 掌握Hibernate查询语言（HQL）的使用



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

Criteria Query 范例

```
Criteria criteria = session.createCriteria(Person.class);  
Criteria.add(Expression.eq("name","lxh"));  
Criteria.add(Expression.eq("sex",new Integer(1))) ;
```

这里的criteria实例本质上是对SQL “Select * from t_user where name = 'lxh' and sex=1”的封装。

Hibernate在运行期会自动根据Criteria中指定的查询条件（也就是上面代码中通过criteria.add方法添加的查询表达式）生成相应的SQL语句。

这种方式的特点是比较符合JAVA程序员的编码习惯，并且具备清晰的可读性。



Criteria查询表达式

- ◆ Criteria本身只是一个查询容器，具体的查询条件需要通过Criteria.add方法添加到Criteria实例中。
- ◆ 如前面所示，Expression对象具体描述了查询条件。针对SQL语法，Expression提供了对应的查询限定机制，见下表。



JAVA培训报名热线：010-51283346

更多学习资源尽在 **www.MLDN.cn**

Criteria查询表达式 (1)

方法	描述
Expression.eq	对应SQL“field=value”表达式 如Expression.eq("name","lxh")
Expression.allEq	参数唯一MAP对象，其中包含了多个属性-值对应关系，相当于多个Expression.eq关系的叠加
Expression.gt	对应SQL “field > value”表达式
Expression.ge	对应SQL “field >= value”表达式



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

查找姓名等于张三的用户的全部信息

```
package org.lxx.oper;

import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxx.pojo.Person;

public class HibernateDemo {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
    }
}
```

```
Criteria cri = session.createCriteria(Person.class);
cri.add(Expression.eq("name", "张三"));
List all = cri.list();
if (all.size() > 0) {
    Person per = (Person) all.get(0);
    System.out.println(per.getName() + " --> " + per.getSex());
}
}
```

检索出年龄大于 25 岁的全部用户

```
package org.lxx.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxx.pojo.Person;

public class HibernateDemo02 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        Criteria cri = session.createCriteria(Person.class);
        cri.add(Expression.gt("age", 25));
        List all = cri.list();
        Iterator iter = all.iterator();
        while (iter.hasNext()) {
            Person per = (Person) iter.next();
            System.out.println(per.getName() + " --> " + per.getSex() + " --> "
                + per.getAge());
        }
    }
}
```

小于 25 岁的全部用户

```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxh.pojo.Person;

public class HibernateDemo02 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        Criteria cri = session.createCriteria(Person.class);
        cri.add(Expression.gt("age", 25));
        List all = cri.list();
        Iterator iter = all.iterator();
        while (iter.hasNext()) {
            Person per = (Person) iter.next() ;
            System.out.println(per.getName() + " --> " + per.getSex() + " --> "
                + per.getAge());
        }
    }
}
```

Criteria查询表达式（2）

方法	描述
Expression.lt	对应SQL “field < value”表达式
Expression.le	对应SQL “field <= value”表达式
Expression.between	对应SQL “between”表达式 如下面的表达式表示年龄（age）位于13到50区间内。 Expression.between(“age”,new Integer(13),new Integer(50))
Expression.like	对应SQL “field like value”表达式



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

lt: <
gt: >
le: <=
gt: >=

查询年龄在 25~35 岁之间的全部用户

```
package org.lxx.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxx.pojo.Person;

public class HibernateDemo04 {
```

```
/**
 * @param args
 */
public static void main(String[] args) {
    Session session = new Configuration().configure().buildSessionFactory()
        .openSession();
    Criteria cri = session.createCriteria(Person.class);
    cri.add(Expression.between("age", 25, 35));
    List all = cri.list();
    Iterator iter = all.iterator();
    while (iter.hasNext()) {
        Person per = (Person) iter.next();
        System.out.println(per.getName() + " --> " + per.getSex() + " --> "
            + per.getAge());
    }
}
```

要求对数据进行模糊查询，查找出姓名中带有“三”的全部用户信息。

```
package org.lxx.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxx.pojo.Person;

public class HibernateDemo05 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        Criteria cri = session.createCriteria(Person.class);
        cri.add(Expression.like("name", "%三%"));
        List all = cri.list();
        Iterator iter = all.iterator();
        while (iter.hasNext()) {
```

```

        Person per = (Person) iter.next();
        System.out.println(per.getName() + " --> " + per.getSex() + " --> "
            + per.getAge());
    }
}

```

Criteria查询表达式 (3)

方法	描述
Expression.in	对应SQL "field in ..."表达式
Expression.eqProperty	用于比较两个属性之间的值，对应SQL中的 "field=field"。如： Expression.eqProperty("Person.groupID","TGroup.id")；
Expression.geProperty	用于比较两个属性之间的值，对应SQL "field > field"
Expression.gtProperty	用于比较两个属性之间的值，对应SQL "field>=field"



JAVA培训报名热线：010-51283346

更多学习资源尽在  www.MLDN.cn

in 操作

```

package org.lxh.oper;

import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.TreeSet;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;

```

```
import org.lxx.pojo.Person;

public class HibernateDemo06 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        Criteria cri = session.createCriteria(Person.class);
        Set<Integer> conds = new TreeSet<Integer>();
        conds.add(new Integer(1));
        conds.add(new Integer(2));
        cri.add(Expression.in("id", conds));
        List all = cri.list();
        Iterator iter = all.iterator();
        while (iter.hasNext()) {
            Person per = (Person) iter.next();
            System.out.println(per.getName() + " --> " + per.getSex() + " --> "
                + per.getAge());
        }
    }
}
```

Criteria查询表达式（4）

方法	描述
Expression.ltProperty	用于比较两个属性之间的值，对应SQL "field<field"
Expression.leProperty	用于比较两个属性之间的值，对应SQL "field<=field"
Expression.and	and关系组合。如： Expression.and(Expression.eq("name","LXH"),Expression.eq("sex",new Integer(1)))；



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

要查询姓名为张三，且性别为男的用户

```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.TreeSet;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Expression;
import org.lxh.pojo.Person;

public class HibernateDemo07 {

    /**
     * @param args
     */
}
```

```
public static void main(String[] args) {
    Session session = new Configuration().configure().buildSessionFactory()
        .openSession();
    Criteria cri = session.createCriteria(Person.class);
    cri.add(Expression.and(Expression.eq("name", "张三"), Expression.eq(
        "sex", "男")));
    List all = cri.list();
    Iterator iter = all.iterator();
    while (iter.hasNext()) {
        Person per = (Person) iter.next();
        System.out.println(per.getName() + " --> " + per.getSex() + " --> "
            + per.getAge());
    }
}
```

Criteria查询表达式 (5)

方法	描述
Expression.or	or关系组合。如： Expression.or(Expression.eq("name","LXH"),Expression.eq("name","LFF"))；
Expression.sql	作为补充，本方法提供了原生SQL语法的支持。可以通过这个方法直接通过SQL语句限定查询条件下面的代码返回所有名称以“lxh”起始的记录： Expression.sql("lower({alias}.name) like lower(?)","lxh%",Hibernate.STRING); 其中“{alias}”将由Hibernate在运行期使用当前关联的POJO别名替换。



JAVA培训报名热线：010-51283346

更多学习资源尽在
www.MLDN.cn

Example类

- ◆ Example类实现了Criterion接口，它也可以用作Criteria的查询条件。Example的作用是：根据已有对象，查找属性与之相符的其他对象。



```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Example;
import org.lxh.pojo.Person;

public class HibernateDemo08 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
    }
}
```

```
Criteria cri = session.createCriteria(Person.class);
Person person = new Person() ;
person.setName("张三") ;
person.setSex("男") ;
person.setAge(20) ;
cri.add(Example.create(person));
List all = cri.list();
Iterator iter = all.iterator();
while (iter.hasNext()) {
    Person per = (Person) iter.next();
    System.out.println(per.getName() + " --> " + per.getSex() + " --> "
        + per.getAge());
}
}
```

HQL

- ◆ Hibernate查询语言（HQL）
- ◆ 这是 Hibernate 官方所推荐的查询语言，接近 SQL 的语法，并提供更多的特性与封装。



JAVA培训报名热线：010-51283346

更多学习资源尽在

www.MLDN.cn

HQL语法结构

◇ [select/update/delete ...]
[from ...]
[where ...]
[group by ... [having ...]]
[order by ...]

其中，update/delete子句为Hibernate3 引入的新特性。



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

update 与 delete 操作是在 Hibernate 3 中引入的。

要查询 Person 表中所拥有的全部性别

```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Example;
import org.lxh.pojo.Person;

public class HibernateDemo09 {

    /**
     * @param args
     */
}
```

```
public static void main(String[] args) {
    Session session = new Configuration().configure().buildSessionFactory()
        .openSession();
    String hql = "SELECT distinct p.sex FROM Person AS p" ;
    Query q = session.createQuery(hql) ;
    List all = q.list();
    Iterator iter = all.iterator();
    while (iter.hasNext()) {
//        Person per = (Person) iter.next();
//        System.out.println(per.getName() + " --> " + per.getSex() + " --> "
//            + per.getAge());
        System.out.println(iter.next()) ;
    }
}
}
```

求出全部的记录数

```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;

public class HibernateDemo10 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        String hql = "SELECT COUNT(p.id) FROM Person AS p" ;
        Query q = session.createQuery(hql) ;
        List all = q.list();
        Iterator iter = all.iterator();
        while (iter.hasNext()) {
            System.out.println(iter.next()) ;
        }
    }
}
```

HQL中使用where子句

- ◇ 可以使用where子句来限定查询的条件，除了 = 运算之外，还有 >、>=、<、<=、!= 或 <>等比较运算，例如：
 - ◇ Session session = sessionFactory.openSession();
 - ◇ Query query = session.createQuery("from Person p where p.name='lxh'");



JAVA培训报名热线：010-51283346

更多学习资源尽在
www.MLDN.cn

通过 HQL 查找姓名等于“张三”的全部用户信息。

```
package org.lxh.oper;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;
import org.lxh.pojo.Person;

public class HibernateDemo11 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
```

```
String hql = "FROM Person AS p WHERE p.name=?" ;
Query q = session.createQuery(hql) ;
q.setString(0, "张三") ;
List all = q.list();
Iterator iter = all.iterator();
while (iter.hasNext()) {
    System.out.println(((Person)iter.next()).getAge()) ;
}
}
```

HQL中使用where子句

- ◆ 可以在where子句上进行表达式：
 - ◆ from Person p where (p.age / 10 = 3)
- ◆ 可以在where子句上使用and、or：
 - ◆ from Person p where (p.age > 20) and (p.name = 'lxh')
- ◆ is not null与is null则可以测试字段值是否为空值
 - ◆ from Person p where p.name is not null



JAVA培训报名热线：010-51283346

更多学习资源尽在
www.mldn.cn

HQL中使用where子句

- ◆ between可以测试字段值是否在指定的范围之内：
 - ◆ from Person p where p.age between 20 and 30
- ◆ 使用in或not in来测试字段值是否在指定的集合中：
 - ◆ from Person p where p.name in('lxh', 'laf')
- ◆ like或not like可以进行模糊条件搜寻：
 - ◆ from Person p where p.name like 'l%'



JAVA培训报名热线：010-51283346

更多学习资源尽在

www.MLDN.cn

HQL中使用where子句

- ◆ 对查询结果使用order by进行排序:
 - ◆ from Person p order by p.age
- ◆ 使用desc反排序:
 - ◆ from Person p order by p.age desc
- ◆ 同时指定两个以上的排序方式，例如先按照“age”反序排列，如果“age”相同，则按照“name”顺序排列:
 - ◆ from Person p order by p.age desc, p.name



HQL中使用where子句

- ◆ 配合GROUP BY子句，自动将指定的字段依相同的内容群组：
 - ◆ `select p.sex, avg(p.age) from Person p group by p.sex`
- ◆ 可以结合having子句，例如只将平均大于20的数据分组显示出来：
 - ◆ `select p.sex, avg(p.age) from Person p group by p.sex having avg(p.age) > 20`



更新、删除

- ◇ 更新：
 - ◇ update User set name='lxh' where name='lll'
- ◇ 删除：
 - ◇ delete Person where name='lxh'



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

将张三的性别修改成“女”

```
package org.lxh.oper;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.cfg.Configuration;

public class HibernateDemo12 {

    /**
     * @param args
     */
    public static void main(String[] args) {
        Session session = new Configuration().configure().buildSessionFactory()
            .openSession();
        String hql = "UPDATE Person SET sex=? WHERE id=?";
        Query q = session.createQuery(hql);
        q.setString(0, "女");
        q.setInteger(1, 1);
    }
}
```

```
        q.executeUpdate() ;  
        session.beginTransaction().commit() ;  
    }  
}
```

Hibernate分页

- ◆ 在Hibernate的Criteria Query和Query接口中提供了以下两个方法，用于完成分页功能：
 - ◆ `setFirstResult(100) ;`
 - ◆ `setMaxResults(20) ;`



JAVA培训报名热线：010-51283346

更多学习资源尽在
www.MLDN.cn

```
package org.lxx.oper;  
  
import java.util.Iterator;  
import java.util.List;  
  
import org.hibernate.Query;  
import org.hibernate.Session;  
import org.hibernate.cfg.Configuration;  
import org.lxx.pojo.Person;  
  
public class HibernateDemo13 {  
  
    /**  
     * @param args  
     */  
}
```

```
public static void main(String[] args) {  
    Session session = new Configuration().configure().buildSessionFactory()  
        .openSession();  
    String hql = "FROM Person AS p" ;  
    Query q = session.createQuery(hql) ;  
    q.setFirstResult(2) ;  
    q.setMaxResults(2);  
    List all = q.list();  
    Iterator iter = all.iterator();  
    while (iter.hasNext()) {  
        Person per = (Person) iter.next();  
        System.out.println(per.getName() + " --> " + per.getSex() + " --> "  
            + per.getAge());  
    }  
}
```

参考的另外一个分页案例 → Struts + DAO + Hibernate

本章重点

- ◆ 掌握HQL的基本使用
- ◆ 掌握Hibernate分页技术



JAVA培训报名热线：010-51283346

更多学习资源尽在  www.MLDN.cn

