

导航

BlogJava

[首页](#)
[新随笔](#)
[联系](#)
[XML 聚合](#)
[管理](#)

< 2011年5月 >						
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

公告

Sparta- 紫杉

PMP（项目管理专业人士）；
 历任Java程序开发、系统分析、部门经理之职；
 目前专注于J2EE相关技术研究及应用、SSH2框架研究与应用。

请通过QQ22086526 或
 SpartaYew@163.com与我联系！

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)

留言簿

[给我留言](#)
[查看公开留言](#)
[查看私人留言](#)

随笔分类

[Database\(rss\)](#)
[Java\(5\)\(rss\)](#)
[JavaScript\(2\)\(rss\)](#)

Hibernate3查询返回Map探秘

—源于我对它的一个误会

sparta- 紫杉 2010-4-14 9:02

开发环境： eclipse3.4.2 + weblogic10.3 + jdk1.6.0_13 + Struts2.1.8 + Spring3.0.1 + Hiberante3.3.2

初识Hibernate3的查询结果返回Map的功能是非常高兴的，因为我第一印象认为它能够解决类似以下的问题：

从角色表SysRoles中，得到以Role_Id（角色ID）为Key，以Role_Desc（角色描述）为Value的Map。

要知道在以前，我通常采用如下的传统方式解决此类问题：

```

public HashMap getRolesMap(){
    try {

        List<SysRoles> results = getHibernateTemplate().find("from Sys
Roles");

        Map hashMap = new HashMap(0);

        for( SysRoles role : results ){
            hashMap.put( role.getRoleId(), role.getRoleDesc());
        }

        return hashMap;
    } catch (RuntimeException re) {

        throw re;
    }
}
    
```

代码虽也不多，短短5行代码，但在代码里面需要经过一次提取，一次加工的过程也有点繁琐，程序员嘛，要本着一个“代码简洁、清晰”的原则。

有无更好的方法呢？那么在知道Hibernate3能够从查询结果中返回Map的功能后，我很自然的写下如下的代码，以为能节省一个加工的过程：

```

public HashMap getPermissionMap(){
    try {
    
```

Manager(3)(rss)
 software engineering(rss)
 SSH2(9)(rss)
 心湖揽翠(2)(rss)

随笔档案

2011年5月 (21)

搜索

最新评论 XML

1. re: 春天的故事－Spring Security3十五日研究
 @林峰

朋友，已发，请查收。
 --sparta-紫杉
 2. re: 春天的故事－Spring Security3十五日研究
 感谢楼主，能不能发一份完整的源码给我
 E-mail:bflcf@126.com
 --林峰

3. re: 春天的故事－Spring Security3十五日研究[未登录]
 1716030936@qq.com，感谢楼主
 --guyue

4. re: 春天的故事－Spring Security3十五日研究
 @guyue

盆友，不妨给我个QQ邮箱地址，发你一份配置参考下吧！
 --sparta-紫杉
 5. re: 春天的故事－Spring Security3十五日研究[未登录]
 按照楼主的配置，程序跑到AbstractUserDetailsAuthenticationProvider类中时，userCache总是NullUserCache对象呢。已经在配置了用户缓存了...
 --guyue

阅读排行榜

1. 春天的故事－Spring Security3十五日研究(10510)
 2. Struts2中以非迭代方式提取Map中的值(1770)
 3. hibernate中提倡持久类实现equals()和hashCode()的原因分析(1503)

```

        HashMap hashMap = (HashMap)getSession()
            .createSQLQuery("select role_Id, role_Desc from Sys_Role
s").setResultTransformer(Transformers.ALIAS_TO_ENTITY_MAP);

        return hashMap;
    } catch (RuntimeException re) {

        throw re;
    }
}
```

哈哈，只有两行代码便能解决这种问题，看起来似乎是理想的选择，赶快试试吧。

运行后，在jsp偌大的空白页面上，一行异常代码分外刺眼：
 org.hibernate.impl.SQLQueryImpl cannot be cast to java.util.HashMap.

看来setResultTransformer是不能直接返回Map的，它是返回一个封装了所有Map的List。

看来我还真是太草率了，对上面语句中的“Transformers.ALIAS_TO_ENTITY_MAP”迷惑了双眼。

看来事情并不那么简单，还需要探索，既然已经开始了，索性探索个明白吧！在经过对代码、对查询方法进行几次修订后，成为如下代码：

```

public HashMap getPermissionMap(){
    try {

        List list = getHibernateTemplate()
            .find("select new Map(sysRoles.roleId as roleId, sysRoles.roleDesc as roleDesc) from SysRoles sysRoles");

        for( Object map : list){
            hashMap.putAll( (Map) map );
        }

        return (HashMap) hashMap;
    } catch (RuntimeException re) {

        throw re;
    }
}
```

修改完成之后，我不禁哑然失笑，之前坚持的所谓的“代码简洁、清晰”原则已经荡然无存。但反观代码，似乎有些道理，同时，对java的使用也有了

4. 365天，730个吻(1417)
 5. Struts2的dojo使用与/temp
 late/ajax/head.ftl not found.
 (827)

评论排行榜

1. 春天的故事－Spring Security3十五日研究(53)
2. 365天，730个吻(5)
3. Struts2中以非迭代方式提取Map中的值(2)
4. Birt + SSH2 完整实践(1)
5. 管理的原理之三－工具理论(1)

更进一步的理解，对浪费的时光也不再那么怜惜了。

可惜事情总不能如你所愿，通过对代码的执行发现，数据库里面有多条数据，但是那个返回的hashMap的size却始终是一条。

我通过下面代码对返回的hashMap进行循环打印，发现仅能打印一条，并且内容始终是一样：

```

for( Map.Entry map : hashMap.entrySet() ){
    System.out.println("#### map1.getKey() = " + map1.getKey() + ", map1
.getValue()=" + map1.getValue());
}

```

唉，还是找找相关的“从Hibernate的查询结果中返回Map”的原理吧，看来我们多少得为我们的草率付出点代价。

下面是我从Hibernate的教程中得到的相关信息，一个说得比较清楚的是javaEye(现在为itEye)的蔡华江，请原谅使用了你的文章：

////////////////////////////////// 摘抄开始 //////////////////////////////////

在问答里和论坛中，经常看到有人问，怎样将使用本地SQL查询出来的结果映射为值对象的问题，这里就Hibernate中提供的方法做个结论。
 前提，这里没有使用属性的延迟加载技术。

假设有个值对像，如下：

```

package test;

public class Person {
    private Long id;
    private String name;
    private Long age;
    private Long phone;
    private String address;

    public Person(Long id, String name, Long age, Long phone, String address)
    {
        this.id = id;
        this.name = name;
        this.age = age;
        this.phone = phone;
        this.address = address;
    }

    public Long getId() {
        return id;
    }
}

```

```
public void setId(Long id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public Long getAge() {
    return age;
}

public void setAge(Long age) {
    this.age = age;
}

public Long getPhone() {
    return phone;
}

public void setPhone(Long phone) {
    this.phone = phone;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
}
```

如果查询全部五列记录的话，那么只要

```
List list = getHibernateTemplate().loadAll(Person.class);
```

如果只需要查询id,name,age三列记录呢？那么就要新增构造方法了，

```
public Person(Long id, String name, Long age) {
    this.id = id;
    this.name = name;
    this.age = age;
}
```

然后呢，就可以通过HQL来进行查询。

```
List list = getHibernateTemplate().find("select new test.Person(id,name,age) from Person");
```

这个方法通常可以满足需要了，只是如果，只需要查询id,name,phone三列记录的话，还新增构造方法？不行了，会出现构造方法冲突了。有个办法：

```
List list = getSession().createQuery("select id,name,phone from person")
    .addScalar("id",Hibernate.Long).addScalar("name").addScalar("phone",Hibernate.Long)
    .addEntity(Person.class);
```

但是，这个只适用于存在Person实体的，如果Hibernate中没有进行Person映射的呢，系统中只存在一个JavaBean。

```
List list = getSession().createSQLQuery("select id \"id\",name \"name\",phone \"phone\" from person")
    .addScalar("id",Hibernate.Long).addScalar("name").addScalar("phone",Hibernate.Long)
    .setResultTransformer(Transformers.aliasToBean(Person.class));
```

那么Hibernate就会自动将查出来的三列内容组装到VO对象中去，只是代码量有点大，而且名称都需要重新定义为小写的，在Oracle中查出来的列都默认为大写的（不知道其它数据库怎么样），这个办法就不依赖于构造方法了，只需要定义私有变量，设置getter/setter方法就行了。

不过如果更猛点的，根本就没有JavaBean对象可以填充怎么办，Hibernate可以将查出来的列组装到集合类中去。如Map。

```
List list = getSession().createSQLQuery("select * from person")
    .setResultTransformer(Transformers.ALIAS_TO_ENTITY_MAP);
```

//sparta-紫杉 注：这个ALIAS_TO_ENTITY_MAP的含义是指以数据库里的字段为key，以字段内容为Value的Map。有几个字段就会生成几个Map，然后将所有的Map组装到List中。

除了这个Transformers还可以将列转化为List。

```
List list = getSession().createSQLQuery("select * from person")
    .setResultTransformer(Transformers.TO_LIST);
```

到此，还可以通过继承Transformers将结果映射为其它对象，不累述了，基本功能够用了。

//////////////////// 摘抄结束 //////////////////////////////////////

哈哈，各位达人，看完上面这篇文章之后，豁然开朗了吧？

请注意在上面文章中笔者注释的一段文字就会明白：在hibernate从查询结果返回Map中，是以字段为key，以字段的内容为Value的Map，而不是我想象的假设我在查询中仅提供两个字段，那么Hibernate就会自然地以我的第一个字段的内容为key，以第二个字段的内容为Value生成Map列表。

那么我也终于明白为什么在我的执行结果中仅有一条数据的原因了，因为在我的代码中返回的list里面，虽然有很多条数据，但是我们应该明白一点：当我使用Map的putAll时，由于在Map中，均是以字段名称作为key的，那么在数据库中无论有多少条的记录，字段名始终是相同的，也就是说，在Map中重复的key会被反复覆盖，仅保留最后的一条。它是返回一个封装了所有Map的List，在使用时，只有迭代该List，提取各Map进行使用才不会导致重复覆盖。

什么样的应用场景赋予我们什么样的想象力，误会有时源于无知，但更多源于我们的无限想象力，有时异想天开的灵感确实能为我们提供打开另一扇未知知识大门的钥匙。

难道不是吗？本次误会让我对Hibernate的理解又深入了一层，无论是原理还是应用场景。再者，不妨将牛吹大一点吧：希望Hibernate团队能够从我的这个误会中吸取点灵感，开发出一个能够返回以前后两个字段的内容分别为key和value的Map的方法来，以方便各位使用Hibernate的程序达人，似乎也未尝不可吧？哈哈！

— 东营 sparta-紫杉 原创，转载请注明出处：)

<http://www.blogjava.net/SpartaYew/>

SpartaYew@163.com

QQ:22086526

posted on 2011-05-19 08:26 sparta-紫杉 阅读(270) 评论(1) 编辑 收藏 所属分类: SSH2

评论

re: Hibernate3查询返回Map探秘 回复 更多评论
不错

2011-10-17 15:20 | 段凯歌

[新用户注册](#) [刷新评论列表](#)

找优秀程序员，就在博客园

IT新闻：

- 2012：苹果的命运分水岭
- 消息称聚划算投放电视广告 13日起登陆央视平台
- Eyeboard：低成本的开源眼动跟踪解决方案
- 苹果iTunes商店登陆巴西等16个拉美国家
- 电子产品：影响夫妻关系的双刃剑

[博客园](#)
[博问](#)
[IT新闻](#)
[Java程序员招聘](#)

标题

姓名

主页

验证码

*

9833

内容(请不要发表任何与政治相关的内容)

Remember Me?

登录

[使用Ctrl+Enter键可以直接提交]



推荐职位：

- 中高级.NET程序员(沪江网)
- 知识库技术编辑(博客园)
- .NET 高级软件开发工程师 (美国新蛋网)

- 北京 **SQL**数据库开发工程师(圣特尔科技)
- 北京**C#**开发工程师 **C/S**方向(圣特尔科技)
- 南京 **.NET**软件开发工程师(江苏百年软件)

博客园首页随笔:

- 最简约**JS**日历控件
- 轻松实现**WCF**服务的构造函数依赖注入
- 将**html**内容生成为**word**文档实现思路
- **javascript1.7**新语法
- 为**ASP.NET MVC**创建一个基于**Unity**的**ControllerFactory**

知识库:

- 如何防止代码腐烂
- **HTTP Caching** 优化网站
- 走向**DBA**[**MSSQL**篇] - 从**SQL**语句的角度提高数据库的访问性能
- **Twitter**首席工程师: 如何“打败”**CAP**定理
- **HTML5**标签使用的常见误区

网站导航:

博客园 IT新闻 知识库 **C++**博客 程序员招聘 管理

相关文章:

春天的故事—**Spring Security3**十五日研究

Birt + SSH2 完整实践

Struts2的**dojo**使用与/template/ajax/head.ftl not found.

Struts2中以非迭代方式提取**Map**中的值

Struts2标签之**Checkbox**详解

hibernate中提倡持久类实现**equals()**和**hashCode()**的原因分析

hibernate3关联映射表的级联关系维护

Spring3中**XmlBeanDefinitionReader**类的玩笑

Powered by:

BlogJava

Copyright © sparta-紫杉