

# 利用 Myeclipse 简化基于 Hibernate 的数据库访问

窦东阳, 赵英凯, 许文稼

(南京工业大学 自动化学院, 江苏 南京 210009)

**摘 要** :Hibernate 作为当今新兴的、最受人瞩目的数据库访问技术,提供了一种很好的数据持久层解决方案。Myeclipse 集成环境功能非常强大,其中之一就是简化基于 Hibernate 的数据库访问。使用 Myeclipse 开发基于 Hibernate 的数据库应用,使得开发人员真正从繁重的代码编写工作中解脱出来,极大地提高了开发效率和项目质量。这里将结合一个实例来介绍这种强大的技术。

**关键词** :数据持久化; 数据库应用; Hibernate; Eclipse; Myeclipse; J2EE

**中图法分类号** :TP311 **文献标识码** :A **文章编号** :1000-7024(2007)04-0773-02

## Simplifying realization of database access based on hibernate using myeclipse

DOU Dong-yang, ZHAO Ying-kai, XU Wen-jia

(College of Automation, Nanjing University of Technology, Nanjing 210009, China)

**Abstract** : As the most rising and attractive technology of database access, Hibernate offers a good resolution of data persistence. Myeclipse IDE is a powerful tool, which can simplify the realization of database access based on Hibernate. Developing Hibernate application with Myeclipse makes the developers free out of the hard code writing. The efficiency and quality of our project is increased greatly. This advanced technology is introduced by an instance.

**Key words** : data persistence; database application; hibernate; eclipse; myeclipse; J2EE

## 0 引 言

在今天的企业环境中,直接调用 JDBC 访问数据库,是非常复杂、浪费时间的。如何提高数据库访问层的开发效率,让开发者专注于手中的业务而无需操心复杂烦琐的数据库连接,成为近年来争相讨论的话题。Hibernate 正是这一时期的产物。而 Myeclipse 使得 Hibernate 的实现高效而简单,成为目前炙手可热的工具之一。

## 1 Hibernate

### 1.1 Hibernate 简述

为了方便企业应用与数据库之间的频繁交互,开发人员在两者之间设计了一个“持久层”,它是专注于实现数据库持久化的一个相对独立的领域。作为一个持久层框架实现,Hibernate 对 JDBC 进行了轻量级的对象封装,提供了强大而高效的对象到关系型数据库的持久化服务。利用 Hibernate,我们可以按照 Java 的基本语义进行持久层开发,随心所欲地使用对象编程的思想操作数据库,把大部分精力集中在业务层次上,完全抛开一些冗长无味的代码编写工作。同时,Hibernate 也提供了面向对象的查询语言 HQL,在 POJO 和关系型数据库之间构建了一条快速、高效、便捷的沟通渠道。

### 1.2 Hibernate 体系结构与原理

Hibernate 技术实质上是一个为企业应用提供各种数据库服务的中间件,体系结构如图 1 所示。

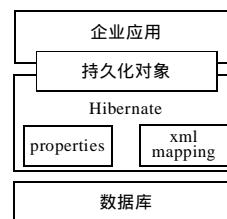


图 1 Hibernate 体系结构

企业应用通过 Hibernate 与数据库联系,对数据进行操作。而 Hibernate 通过 properties 文件和 xml mapping 文件将 POJO 类映射到数据库表的对应行,从而,企业应用可以通过持久化的对象类直接访问数据库,而不必使用 JDBC 和 SQL 连接数据库和操作数据。这样一来,对数据的操作变得跟操作对象一样简单。例如,使用 Hibernate 的查询,可以直接返回包含某个值对象的列表,而不必像传统的 JDBC 访问方式一样把结果集的数据逐个装载到值对象中,大大简化了开发工作。

收稿日期:2006-01-16 **E-mail** :doudy41@sina.com

**作者简介**:窦东阳(1983-),男,江苏扬州人,硕士研究生,研究方向为数据库技术、智能控制;赵英凯(1943-),男,教授,博士生导师,研究方向为智能控制、神经网络、过程控制等;许文稼(1983-),女,硕士研究生,研究方向为智能控制。

## 2 Myeclipse

### 2.1 Myeclipse 简述

Eclipse 是一个基于 Java 的、开源的、可扩展 IDE。就其本身而言,它只是一个框架和一组服务,它的强大体现在可以通过下载丰富的插件来搭建不同的开发环境以满足各种开发要求。另外,Eclipse 是一个开放源代码的项目,任何人都可以免费下载 Eclipse 的源代码,然后根据自己的需要在此基础上开发自己的功能插件。

Myeclipse 就是这样一个 Eclipse 支持下的 J2EE 开发插件,功能十分完善,具备了完备的编码、调试、测试和发布等功能并且完整支持 HTML、Struts、JSF、CSS、JSP、SQL、Hibernate、Spring 等技术。Myeclipse 插件基本上涵盖了众多 Eclipse 插件的功能,能满足我们的各种应用需求,基本上安装了它就无须其它任何插件。甚至可以这样理解,Myeclipse 就是一个比较完备的、功能强大的开发环境。

### 2.2 Myeclipse 简化 Hibernate 原理

如上所述,Myeclipse 提供了对 Hibernate 很好的支持,它使得 Hibernate 的实现极其简单,大部分基础代码都可以由它帮你完成,极大地提高了开发效率。

Hibernate 基础代码包括 POJO 类和 Hibernate 映射文件。以往,在最原始的配置文件编写方式——手工编写时,从业务逻辑出发,我们不但需要考虑到实体类以及数据库表的对应关系,还要遵循 Hibernate 的相关规范,一行一行地编写,难免顾此失彼,既耗时也容易出错。此时,应用 Hibernate 至少在代码编写上较之传统的数据库访问方式并没有显著改善。

Myeclipse 的出现改变了这种情况,它帮我们完成了大部分烦琐重复的工作。虽然现在这类工具很多,但无疑它的功能是最完备的。现在,通过 Myeclipse,我们可以根据数据库表结构轻松地导出对应的映射文件和 POJO 类。这样操作极大地减少了编码的工作量并且在最大程度上保证了映射文件、POJO 类与实际数据库结构的一致性、同步性,提高了编码的质量,而且不容易出错,是目前国外非常流行的开发方式之一。

## 3 开发应用

也许光有上面的阐述会让人觉得是“纸上谈兵”,下面我们结合一个简单但完整体现上述思想的实例,来体会 Myeclipse 给我们工作带来的改善。

这里,我们需要安装了 Myeclipse 插件的 Eclipse 3.0 开发环境、Mysql 数据库,以及相应的 JDBC 驱动程序。

#### (1) 创建一个示例数据库

这个示例数据库非常简单,仅有一张表格,表格名 `t_user`,它包含两个字段:递增主键“id”和用户名“name”。

#### (2) 创建 Hibernate 配置文件

启动 Myeclipse,新建一个 Java Project,利用 Myeclipse 添加“Hibernate capabilities”,这时,Myeclipse 会把使用 Hibernate 时所需的各个库添加进来,并生成 `hibernate.cfg.xml`,它是 Hibernate 的配置文件,主要用于配置数据库连接以及其它各种属性。

现在,我们所要做的只是输入一些数据库连接的属性并且将 JDBC 驱动包添加进来,Myeclipse 会自动帮我们生成这个文件,不需要我们手工编写。

```
<hibernate-configuration>
<session-factory>
<property name="myeclipse.connection.profile">hitt_mysql
</property>
<property name="connection.url">jdbc:mysql://localhost:3306/
test</property>
<property name="connection.username">root</property>
<property name="connection.password">123456</property>
<property name="connection.driver_class">com.mysql.jdbc.
Driver</property>
<property name="dialect">org.hibernate.dialect.MySQL Dia-
lect</property><mapping resource="TUser.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

#### (3) 创建 POJO 类和 Hibernate 映射文件

打开 Myeclipse Database Explorer,找到我们的表 `t_user`。Myeclipse 可以根据选择的表的结构,自动完成对象类到数据库的映射,避免了手工编写的缺陷。这里,项目中生成了 `TUser.java` 和 `Tuser.hbm.xml`,前者是数据库表对应的 POJO 类,后者则是两者之间的映射关系文件,这些也都是由 Myeclipse 帮我们写好的,不需要花费太大精力。

`TUser.java` 文件:

```
public class TUser extends AbstractTUser implements Seriali-
zable{
    public TUser(){
    }
    public TUser(java.lang.Integer id){
        super(id);
    }
}
```

`Tuser.hbm.xml` 文件:

```
<hibernate-mapping>
<class name="TUser" table="t_user">
    <id name="id" column="id" type="integer">
        <generator class="native"/>
    </id>
    <property name="name" column="name" type="string"
not-null="true" />
</class>
</hibernate-mapping>
```

#### (4) 持久化对象的操作

上面的步骤已经完成了 Hibernate 的基础代码和配置文件的编写,还需要编写一个简单的持久化代码来测试这个实例。这里,我们以插入一个名为“Doudy”的记录为例。

```
Configuration config=new Configuration().configure();
SessionFactory sessionFactory=config.buildSessionFactory();
(下转第 788 页)
```

较少时,与 Artifact 和 Post 方式相比,Pass 方式并不占优势;但是,随着资源访问数目的增加,Pass 方式还是显示出了明显的优势。这是由于最差状态下,用户与每一个初次访问的站点间都要进行一次信息交互,当用户与域内每个站点都进行过信息交互后,再次访问时就不再进行交互。

安全信息保护方面,引入 Pass 的方式中,用户全局标识对用户本身和本地浏览器都是透明的,它只是在需要进行信任转移时在站间传递,而用户的本地标识则是保留在本地浏览器中。这种信息保存机制在确保用户调用效率的同时保证了信息的安全性。信息处理模块的广播服务为用户信息更改,实时访问控制带来了方便。

当然由于采用 Pass 的方式,我们需要构建统一的 CA,需要在各个站点间加入广播机制,需要在资源端设置用户本地标识和全局标识的映射表,这些工作需要在系统中添加额外的模块,给系统构建带来了一定的开销。但是与单点登陆安全性、灵活度和可控制性的提高相比这些牺牲是值得的。

SAML 自身存在安全隐患如:信息篡改、信息插入、冒名用户、冒名服务器、重放攻击等<sup>[8]</sup>。通过 PKI 机制的引入,和对 Pass 中断言的数字签名,可以防止信息篡改、插入等攻击。通过日志机制、PKI 机制和独享机制可以对冒名用户、冒名服务器攻击进行控制。重放和拒绝服务攻击可以通过各站点的安全策略定义用户在单个站点的最大访问次数来进行控制。

### 3 结束语

引入 Pass 的基于 SAML 的单点登陆模型,既能发挥 SAML

优点,又能实现访问实时控制,更重要的是减少了安全信息交互的次数,在简化单点登陆过程的基础上,增加了系统的安全性。由于模型构建的前提是信任域内各站点都是可信的,对于冒名用户和冒名服务器攻击只能进行有限的防范。而且本模型暂时还未能完全避免 SAML 自身的一些安全隐患,如:重放攻击等,安全模块还需要进行不断改进。

### 参考文献:

- [1] Microsoft. .Net passport review guide[Z].2003.
- [2] SAML Specification. OASIS [EB/OL]. 2005.<http://xml.coverpages.org/draft-sstc-ftp3-saml-spec-00.pdf>,
- [3] Assertion and protocols for the OASIS security assertion markup language(SAML) V2.0[S].OASIS Standard, 2005.
- [4] Profile for the OASIS security assertion markup language (SAML) V2.0[S].OASIS Standard, 2005.
- [5] Jeff Hodges,Rob Philpott.Glossary for the OASIS security assertion markup language (SAML) v 2.0[S].2005.
- [6] Eve Maler,Prateek Mishra.Assertion and protocol for the OASIS security assertion markup language (SAML) v 1.1[S].2003.
- [7] Jeff Hodges, Rob Philpott. Assertion and protocol for the OASIS security assertion markup language (SAML) v 2.0[S]. 2005.
- [8] Jeff Hodges,Rob Philpott.Security and privacy consideration for the OASIS security assertion markup language (SAML) v 2.0 [S].2005.

(上接第 774 页)

```
session=(sessionFactory).openSession();
```

这里利用默认的配置文件 hibernate.cfg.xml 创建且初始化了 Configuration 实例,完成了与数据库的连接,并得到了 Session 实例,通过它就可以完成对象持久化操作。

```
tran=session.beginTransaction();
```

```
TUser user=new TUser();
```

```
user.setName("Doudy");
```

```
session.save(user);
```

```
session.flush();
```

```
tran.commit();
```

```
Assert.assertEquals(user.getId().intValue(),0,true);
```

这样我们就把 TUser 对象实例保存到数据库中,即在数据库表中添加了一条“Doudy”的记录。

以上是我做一个客户管理项目时起步的例子。我对 Hibernate 的理解正是从它开始的,在此基础上,进一步开发将会变得非常轻松。结合这个实例来理解 Myeclipse 对 Hibernate 的支持,可以很好地领会其中的奥妙。

### 4 结束语

Hibernate 的持久化服务,让开发人员得以把大部分精力用在业务逻辑的实现上,而无须为数据库的连接一筹莫展,让操作数据像操作对象一样方便,可以更好地运用面向对象的思想开发,Myeclipse 作为一个优秀的集成开发环境,进一步简

化了 Hibernate 的持久化实现,自动生成大部分的基础代码,使我们真正从烦琐的代码编写中解放出来。现在,我们所要做的重点仅仅是考虑数据库表的构建、业务逻辑实现等核心内容,无论是开发效率或是项目质量都获得了显著的提高。

### 参考文献:

- [1] 夏昕,曹晓钢,唐勇.深入浅出 Hibernate[M].北京:电子工业出版社, 2005.
- [2] Christian Bauer,Gavin King. Hibernate in action [M]. Manning Publications, 2004.
- [3] 张南平,梁方勇.基于 Struts 框架和 Hibernate 对象的 Web 应用[J].微机发展, 2005,15(9):39-41.
- [4] 宋汉增,沈琳.利用 Hibernate 对象持久化服务简化 Java 数据库访问[J].计算机应用, 2003,23(12):135-137.
- [5] 董洪彬,冀延平.利用 Hibernate 的 J2EE 数据持久层的解决方案[J].计算机工程, 2004,30:17-18.
- [6] Justin Couch.J2EE 宝典[M].北京:电子工业出版社, 2002.
- [7] 徐长盛,戴超,谢立.一种快速开发 Web 应用程序方法的研究[J].计算机工程与设计, 2004,25(12):2237-2239.
- [8] 梁添才,皮佑国. J2EE 数据持久层的 ORM 设计模式[J].深圳信息职业技术学院学报, 2005,(3):15-19.
- [9] 张秋余,余冬梅,唐静兵,等.基于 J2EE 的共享数据环境技术研究[J]. 计算机工程与设计, 2005,26(3):588-589.