

基于 MyEclipse 的 Hibernate 持久层 框架的开发过程

According to the MyEclipse Hibernate holds out for long time the
process of development of the layer frame

曹大有 王瑜 (郧阳师范高等专科学校 计算机科学系 湖北 丹江口 442700)

摘要: Hibernate 框架是当前最炙手可热的持久层框架技术,它为 Java 应用提供了一个高效的 O/R Mapping 框架。MyEclipse 是一个 Eclipse 支持下的 J2EE 开发插件,功能十分强大,具备完美的编码、调试、测试和发布功能并完整支持 Struts、Spring 和 Hibernate 等框架技术。这里将通过实例介绍如何通过 MyEclipse 来简化 Hibernate 框架的开发过程,并使开发人员真正的从繁重的代码编写工作中解脱出来。

关键词: 数据持久化 框架 Eclipse MyEclipse

1 引言

在如今的企业级应用开发环境中,面向对象的开发方法已成为主流,但对象只能存在于内存中,而内存不能永久保存数据。如果要永久保存对象的状态,需要进行对象的持久化,即把对象存储到专门的数据存储库中。目前关系数据库仍然是主流的,它存放的是关系数据,而非面向对象的数据。对象数据和关系数据是业务实体的两种表现形式,业务实体在内存中表现为对象数据,在数据库中表现为关系数据。因此把对象数据持久化到关系数据库中,需要进行对象-关系的映射,这是一项繁琐耗时的工作。

在 Java 传统的领域中,企业应用可以直接通过 JDBC 编程来访问关系数据库,通过 JDBC 来实现对象-关系的映射。这种方式虽然运行效率高,但因为要在代码中嵌入大量的 SQL 语句,使得项目难以维护。幸运的是,为了方便企业应用与关系数据库之间的数据交换,开发人员便在两者之间设计了一个“持久化的中间层”,它是专注于实现数据持久化的一个相对独立的领域。目前已有多种现成的对象-关系映射工具可供选用,有些是商业性的,如 TopLink,有些是非商业性的,如 JDO 和 Hibernate。

2 Hibernate 简介

Hibernate 是一个基于 Java 的开放源代码的持久化中间层技术,它通过对 JDBC 进行轻量级的封装,不仅提供了对象关系映射服务,而且还提供了数据查询和数据缓存功能,Java 开发人员可以方便地通过 Hibernate API 来操纵数据库。Hibernate 即适用于独立的 Java 应用程序,也适用于 Java Web 应用,而且还可以在 J2EE 架构中取代 CMP,完成对象持久化的重任。利用 Hibernate 我们可以按照 Java 的基本语义进行持久层开发,使用完全面向对象的查询语言(HQL)操作数据库,使开发人员把主要精力集中在业务层上,完全抛开冗长无味的 JDBC 编程工作,而且不用关心底层数据库结构的实现。Hibernate 的核心思想是:数据库中的一行对应 Java 类的一个实例。

Hibernate 框架是当前最炙手可热的持久层框架技术之一,它为 Java 应用提供了一个高效的 O/R Mapping 框架。Hibernate 的高层概览如图 1 所示。

应用层(Application)通过 PO(Persistent Object)作为 Hibernate 框架的桥梁,而 PO 会根据 Hibernate 的配置文件找到映射文件,并通过映射文件与数据库中的表

结构进行运行时的映射。Hibernate 的配置文件可以是 properties 文件,也可以是一个 XML 文件。使用 properties 文件配置 Hibernate 时,无法直接在文件中描述映射文件的位置,而需要编写代码。而使用 XML 配置文件则可以直接配置映射文件的位置。在图 1 中 properties 指的就是配置文件,而 XML Mapping 指的就是映射文件。

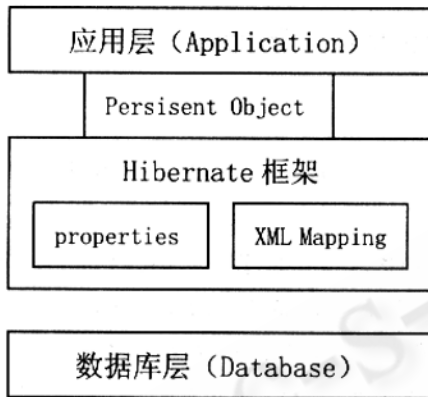


图 1 Hibernate 高层概览

由于应用层可以通过 PO 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,对数据库的操作变得和操作对象一样简单。如使用 Hibernate 的 HQL 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,大大简化了开发过程。

3 MyEclipse 简介

Eclipse 是一个基于 Java 的、开源的和可扩展的集成开发环境。就其本身而言,它只是一个框架和一组服务,它的强大之处在于它可以通过安装不同的插件来建立不同的开发环境以满足各种开发需求。另外 Eclipse 是一个开放源代码的项目,任何人都可以免费下载它的源代码,然后根据自己的需要并在此基础之上开发自己的功能插件。

MyEclipse 是一个 Eclipse 支持下的 J2EE 开发插件,功能十分强大,具备完美的编码、调试、测试和发布功能并完整支持 Struts、Spring 和 Hibernate 等框架技术。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,基本上是安装了它就无须其它插件。甚至可以这样理解,MyEclipse 就是一个比较完备的、功能十分强大的开发环境。

MyEclipse 提供了对 Hibernate 很好的支持,它使得 Hibernate 的开发实现变得十分简单,大部分基础性的代码都可以由 Hibernate 帮我们完成,极大地提高了应用程序开发的效率。

Hibernate 的基础代码包含 POJO 类文件和 Hibernate 的配置文件、Hibernate 的映射文件。配置文件名默认为 hibernate. cfg. xml,它通常会配置持久层框架的如下信息:数据库连接 JDBC、底层数据库方言、事务管理类型、映射文件的加载等。POJO 就是数据库表所对应的域模型对象,它通常只包含 get/set 方法。映射文件用来定义要持久化的对象,在这个文件中要描述持久化对象的变量属性和表字段的对应和对象与对象之间、表与表之间的对应关系。原来这些都是要采用手工的方式来编写它们,这样我们不但要考虑到 POJO 类与数据库表之间的对应关系,而且还要遵循 Hibernate 的相关规范,难免会顾此失彼出现错误。应用 Hibernate 在代码编写上较之传统的 JDBC 方式并没有显著的改善。但 MyEclipse 的出现则极大地改变了这种情况,它能帮助我们完成许多烦琐重复的工作。通过 MyEclipse 不但可以生成 Hibernate 的映射文件 hibernate. cfg. xml,根据数据库表的结构轻松地生成对应的映射文件和 POJO 类文件,还可以创建辅助类 HibernateSessionFactory,该辅助类 HibernateSessionFactory 利用单例模式来完成创建 Session 实例和关闭 Session 实例工作,也可以把它看成是创建数据库连接和关闭数据库连接的工作。它提供了两个方法分别对应这两种工作:getSession()和 closeSession()。这两个方法可以保证在客户端调用时永远只生成一个 Session 实例。所以通过 MyEclipse 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 类文件与实际数据库结构的一致性、同步性,提高了编码的质量且不容易出现错误。

4 实例开发应用

下面我们将通过一个完整实例来体现如何用 MyEclipse 来简化 Hibernate 的编程。首先需要安装了 MyEclipse 5.0 插件的 eclipse - SDK - 3.2.1 - win32 开发环境,MySQL 数据库和 MySQL 数据库的 JDBC 驱动程序 mysql.driver.jar。

(1) 导入实例数据库。运行 schema 目录下的 sampledb.sql,将会创建 MySQL 数据库 SAMPLEDB 及客户表

格 CUSTOMERS 和订单表格 ORDERS。它们的关联关系如图 2 所示。

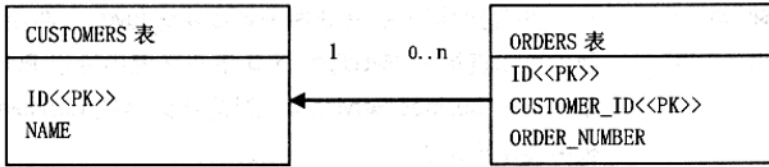


图 2 ORDER 表参照 CUSTOMERS 表

(2) 启动 MyEclipse, 新建 Java Project: MyHibernate 并通过 Add External Jars 导入 MySQL 的 JDBC 驱动程序 mysql-connector.jar, 在目录 MyHibernate 下建一 Source Folder: src, 在 src 下建一 Package: mypack。我们将在 src 下存放配置文件和映射文件, 而在 mypack 下存放类文件。接着建立 MyEclipse 与 MySQL 数据库集成, 然后在 DB Browser 面板中新建一 Profile Name 为 mysql, User Name 为 root 的数据库简表, 并打开指向 SAMPLEDB 数据库的连接。

(3) 创建 Hibernate 的配置文件。hibernate.cfg.xml。利用 MyEclipse 菜单的 Add Hibernate Capabilities 子菜单来为项目增加 Hibernate 2.1 的支持库, 这时 MyEclipse 就会将使用 Hibernate 2.1 时所需的各个库增加进来, 指定在 src 目录下生成配置文件 hibernate.cfg.xml, 并指定在 mypack 包下建立辅助类 HibernateSessionFactory.java。配置文件 hibernate.cfg.xml 的内容大致为:

```

<hibernate-configuration>
<session-factory>
<property name="dialect">
net.sf.hibernate.dialect.MySQLDialect
</property>
<property name="connection.url">
jdbc:mysql://localhost:3306/SAMPLEDB
</property>
<property name="connection.username">root
</property>
<property name="connection.password">
651218
</property>
<property name="connection.driver_class">
com.mysql.jdbc.Driver
  
```

```
</property>
```

```

<property name="myeclipse.connection.profile">
mysql
</property>
<mapping resource="Customer.
hbm.xml" />
<mapping resource="Order.hbm.
xml" />
</session-factory>
</hibernate-configuration>
  
```

(4) 建立 POJO 类文件和 Hibernate 的映射文件: 在 MyEclipse 的 DB Browser 面板中找到表 CUSTOMERS 和 ORDERS, MyEclipse 可以根据选择的表结构, 完成 POJO 类文件和映射文件的自动生成。如在 MyEclipse 的 DB Browser 面板中右键单击表 CUSTOMERS, 在弹出的菜单中选择 Hibernate Reverse Engineering 子菜单, 经过一系列简单的选择确认之后, 就可以在 mypack 包下生成与表 CUSTOMERS 对应的 Customer.java 文件和在 src 目录下生成 Customer.hbm.xml 映射文件。

Customer.java 文件内容为:

```

package mypack;
public class Customer implements java.io.Serializable{
private static final long serialVersionUID = 1L;
private long id;
private String name;
public Customer(){}
public Customer(String name){this.name =
name;}
public long getId(){return this.id;}
public void setId(long id){this.id = id;}
public String getName(){return this.name;}
public void setName(String name){this.name =
name;}}
  
```

Customer.hbm.xml 映射文件内容为:

```

<hibernate-mapping>
<class name="mypack.Customer" table="
customers">
<id name="id" type="long">
<column name="ID" />
  
```

```

        <generator class = "increment" > </gen-
erator >
    </id >
    <property name = "name" type = "string" >
        <column name = "NAME" length = "15"
/ >
    </property >
</class >
</hibernate - mapping >

```

需要注意的是:Customer 类的中域:private static final long serialVersionUID = 1L;需手工加入。由于表 CUSTOMERS 和表 ORDERS 之间存在一对多的关系,所以在自动生成映射文件 Customer.hbm.xml 时默认是生成双向关联关系,而我们这里只考虑单向关联关系,所以最后要对自动生成的 Customer.java 文件和 Customer.hbm.xml 映射文件进行简单的修改。而对 Order.java 和 Order.hbm.xml 文件也可仿此处理。

(5)对象的持久化操作:上面我们完成了 Hibernate 的基础代码和配置文件、映射文件的生成,下面我们需要编写一个测试类来完成对象的持久化操作。我们在 mypack 包下建测试类 BusinessService,Session 实例的创建我们用 HibernateSessionFactory.getSession()方法,该方法会自动根据配置文件和映射文件完成对数据库的连接操作并得到 Session 实例,通过 Session 实例我们就可以完成对象持久化的操作。如:

```

Session session = HibernateSessionFactory.getSession();
Transaction tx = null;
tx = session.beginTransaction();
Customer customer = new Customer("Tom");
session.save(customer);
Order order1 = new Order("Tom_Order001", customer);
Order order2 = new Order("Tom_Order002", customer);

```

```

session.save(order1);
session.save(order2);
tx.commit();

```

其它操作可参见 BusinessService.java 文件所示。我们可以直接在 MyEclipse 5.0 集成环境中通过 Run 菜单来测试 BusinessService 类,测试结果会在 Console 面板中显示出来。

5 总结讨论

Hibernate 的持久化服务,可以让程序开发人员将主要精力集中在业务逻辑的实现上,而无需关注如何与数据库的连接问题上。而 MyEclipse 则是一个支持 Hibernate 编程的优秀集成环境,它可以简化 Hibernate 的持久化过程,自动生成基础代码、配置文件和映射文件以及关联关系。我们所要考虑的重点是数据库的创建和业务逻辑的实现,这样程序的开发速度和代码的质量都得到了很大的提高。项目 MyHibernate 中的代码是在 MyEclipse 5.0 上调试通过的。

参考文献

- 1 梁立新编著,项目实践精解:基于 Struts - Spring - Hibernate 的 Java 应用程序开发[M],北京:电子工业出版社,2006.
- 2 孙卫琴编著,精通 Hibernate:Java 对象持久化技术详解[M],北京:电子工业出版社,2006.
- 3 戎伟、张双编著,精通 Struts - Java 流行服务器、框架、工具及整合应用[M],北京:人民邮电出版社,2006.
- 4 曹大有、王启东,JAVA 2 EventQueue 类与 JAVA 2 Swing 组件的界面刷新[J],微计算机应用,2006(3).
- 5 曹大有,JAVA 2 数组容量扩充的通用方法[J],微计算机应用,2005(2).
- 6 曹大有,JAVA 2 AWT 中 paint()和 repaint()方法的合理使用[J],微计算机应用,2005(4).