

ORM 框架的底层实现机制

周志刚, 王 彬

(安徽理工大学 计算机科学与工程学院, 安徽 淮南 232001)

摘 要: ORM(Object/Relation Mapping), 即对象/关系数据库映射。ORM 框架就是为了解决面向对象程序设计语言与关系型数据库之间“阻抗不匹配”而产生的, 这种框架能将业务逻辑层和持久层有效分离, 使两层之间以松耦合方式衔接起来, 便于代码的维护。探讨了 ORM 框架的实现机制。

关键词: 对象/关系映射; 关系型数据库; 阻抗不匹配; 松耦合

中图分类号: TP301

文献标识码: A

文章编号: 1672-7800(2011)02-0008-02

0 引言

随着计算机技术的蓬勃发展, 传统面向过程的编程语言已逐渐失去了以前的主流地位。由于面向对象的编程语言更接近人类的思维习惯, 具有封装、继承、多态等优良特性, 从而成为了当今计算机领域的主流语言。而目前主流的数据库产品依然是关系型数据库, 使得编程语言和底层数据库的发展不协调, 也就是我们常说的“阻抗不匹配”, 这就需要一座桥梁把二者结合在一起, ORM 框架充当的就是这个功能。

1 ORM 框架实现机制

所有的 ORM 框架都是采用图 1 所示方式映射的。

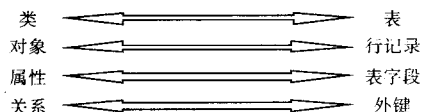


图 1 映射方式

即把实体类映射成表, 类的一个实例映射成表中的一行记录, 类的属性映射成表的字段, 实体类之间的关系映射成外键。那么 ORM 框架又是如何实现这种映射机制的呢? 熟悉 hibernate 的开发人员都知道, 它是由映射类(*.java)、映射文件(*.hbm.xml)和配置文件(*.cfg.xml)来完成以上映射的。映射类也就是我们说的持久类, 由图 1 可以看出, 它的作用是对应数据库中的表, 一个实例对应表中的一条记录, 属性对应表字段。映射文件提供了持久类与数据库之间具体的对应信息, 如: 持久类对应了数据库的哪张表, 属性对应哪些表字段等。配置文件主

要的作用是配置数据库的连接信息, 描述了数据库的驱动、用户名、密码等。所有的 ORM 框架都一样, 也没有什么神奇之处, 底层都是采用 JDBC, 它们只是对底层的 JDBC 进行了封装, 让我们使用者看不到 JDBC 那些常用 CURD 操作。每种 ORM 框架都像 hibernate 一样提供了配置文件(*.xml), 要完成映射就得解析相关的.xml 的配置文件, hibernate 是采用 dom4j 来解析的。

1.1 dom4j 解析.xml 文件

Dom4j 是一个易用的、开源的库, 用于 XML, XPath 和 XSLT。它应用于 Java 平台, 采用了 Java 集合框架并完全支持 DOM, SAX 和 JAXP。XML 文件的结构就是由一个根元素和一些子元素组成的, dom4j 他为我们提供以 java 方式解析 XML 文件的 API, 我们可以利用这些 API 来完成解析取得有用的信息。Hibernate 通过 dom4j 取得配置信息后, 底层仍采用 JDBC 来进行数据库连接。数据库连接过程如图 2 所示。

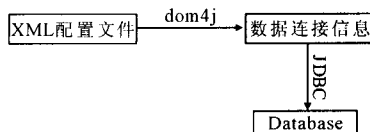


图 2 Hibernate 与数据库的连接过程

1.2 Java 的反射机制

要完成 ORM 映射还得用到 java 的反射机制。反射机制的作用是完成解析 java 类文件。

通过解析映射文件和数据库配置文件我们可以建立数据库的连接, 可以得到映射类的名字、属性名、数据库表名、字段名以及类型等信息。要把数据库中表的数据映射成为对象, 首先需要把表中的记录取出, 然后将每个字段值给映射类对象的每个属性, 这个赋值过程要调用对象中

的 set 方法。我们现在通过映射文件只知道类名和属性名,如何根据类名和属性名调用相应的 set 和 get 方法,是一个关键问题。在 Java 中有一种机制叫反射机制,使用这种机制我们可以得到类的信息,包括类只用的修饰符、方法、属性、继承的父类以及实现接口等信息。反射机制相关的类有 Class、Field、Method 以及 Constructor 等。通过 Class 的 getFields()、getMethods() 和 getConstructors() 方法得到相应的属性、方法和构造方法。通过 Field 类的 getName()、getType() 和 getModifiers() 方法得到相应的属性名、属性类型、属性修饰符信息。通过 Method 类 getReturnType() 可以获取方法的返回类型,invoke() 方法可以根据给定的方法名和参数值执行对象中对应的方法。我们可以首先通过以上方法获取类中的属性名,然后拼写成 setXXX 和 getXXX 方法名,最后根据方法名执行对应的方法,将数据库数据加载到对象中,完成映射的过程,如图 3 所示。

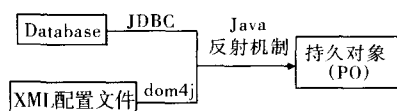


图3 利用反射完成数据库到持久对象的映射

由以上分析可知,要完成一个 ORM 机制并不是一件非常艰难的事情,说简单点就是 dom4j + Java 的反射机制。现在有些公司用的是自己开发的 ORM 框架,学会了

这两样技术,加点 JDBC 的基础知识,你也可以开发一个小小的 ORM 框架了。这里只是简单谈了下对象到关系型数据库的映射,就不讨论 hibernate 那些懒加载和缓存等技术了。

2 结束语

正如大多数开发人员所说的,Java 框架太多,已到了泛滥的程度,在我们学习框架的过程中,我们不应该花太多的时间在学习框架上,而是去掌握那些最基础最底层的东西,即使是出现了新的框架也不必担心,因为大多数框架的设计都是大体相同的,了解了一种框架,结合基础知识去学习一个新框架就是小菜一碟了。要想掌握 ORM 框架,还是得先学习好底层的 JDBC 操作。

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] 张淑全.基于 hibernate 数据层设计模式的研究与实现[D].大连:大连海事大学,2007.
- [3] 何铮.对象/关系映射框架的研究与应用[J].计算机工程与应用,2003(6).
- [4] 林寒超,张南平. Hibernate 技术的研究[J].计算机技术与发展,2006(5).

(责任编辑:杜能钢)

The Underlying Implementation Mechanism of ORM Framework

Abstract: ORM is object / relation database mapping. The generation of ORM framework is to solve the "impedance mismatch" problem between object-oriented programming language and relational databases. It can effectively separate the business logic layer from persistence layer, making the two layers link up in loosely coupled, to maintain code. This thesis is mainly to discuss the implementation mechanism of ORM framework.

Key Words: Object/Relation Mapping; Relational Database; Impedance Mismatch; Loosely Coupled