

# 基于 XML 应用的 MyBatis 技术

贺 军<sup>1</sup>, 焦 荣<sup>2</sup>

(1. 北方自动控制技术研究所, 太原 030006; 2. 北京汉林信通信息技术有限公司, 北京 100085)

**摘 要:**持久层的概念越来越多地应用于 J2EE 框架中, MyBatis 作为持久层中一个优秀的技术框架, 凭借其良好的可扩展性和易维护性等优势, 有着越来越多的应用。主要介绍基于 XML 应用的持久层框架 MyBatis 框架的技术特点, 对从事相关技术的开发研究具有一定的参考价值。

**关键词:**数据持久层, MyBatis, XML, SqlSessionFactory

中图分类号: TP393

文献标识码: A

## Technical Research of MyBatis Based on XML Applications

HE Jun<sup>1</sup>, JIAO Rong<sup>2</sup>

(1. North Automatic Control Technology Institute, Taiyuan 030006, China;

2. Beijing Hanlin Xintong Information Technology Co., Ltd., Beijing 100085, China)

**Abstract:** The concept of persistence layer concept is used in the J2EE framework more and more. MyBatis as an excellent technical framework in the persistence layer, by its good scalability and easy maintenance, is more and more used in many applications. This paper describes the technical characteristics of the MyBatis framework in the persistence layer, based on XML applications, which has some certain reference value for engaging in the research and development of related technologies.

**Key words:** data persistence, MyBatis, XML, SqlSessionFactory

### 引 言

J2EE(Java 2 Platform, Enterprise Edition)作为一个企业级的应用,广泛地应用于各类大型 web 应用系统中,不仅需处理各类异构平台之间的差异与信息交互,还需灵活应对各类变更,因此,需具有良好的可伸缩性、灵活性、易维护性。基于此类需求,处理应用层对象模型和数据库层关系模型的数据持久层被越来越多的应用于 J2EE 平台中<sup>[1]</sup>。MyBatis 技术作为一个处理数据持久层的框架,使用 XML 简单映射,将应用层的对象模型映射成数据库层的关系模型,实现了对象范例和关系范例之间的信息交互。本文主要介绍 XML 在处理数据持久层的 MyBatis 框架中的应用。

### 1 数据持久层简介

J2EE 核心是一组技术规范与指南,其中所包含的各类组件、服务架构及技术层次,均有共同的标准及规格,让各种遵循 J2EE 架构的不同平台之间,存在良好的兼容性和实时有效的数据交互性。根据不同平台的功能,可将 J2EE 平台由

上到下划分为 5 层体系:表示层(Presentation)、控制/中介层(Controller/Mediator)、领域层(Domain)、数据持久层(Data Persistence)和数据源层(Data Source)<sup>[2]</sup>。

在五层体系中,数据持久层位于处理业务逻辑的领域层和负责与数据库交互的数据源层之间。由于领域层的对象范例和数据源层的关系范例这两大领域之间存在“阻抗不匹配”,所以数据持久层需为二者之间提供一个成功的企业级映射解决方案,需尽最大可能弥补这两种异构平台之间的差异,并很好地完成两者之间的信息交互。作为处理对象映射的 MyBatis 框架,以其良好的扩展性和可移植性等优点,在数据持久层的框架中有着越来越广泛的应用。

### 2 MyBatis 框架和 XML 文档概述

#### 2.1 MyBatis 框架概述

MyBatis 是一个处理对象关系映射映射的数据持久层框架,支持普通 SQL 查询,存储过程和高级映射,使用简单的 XML 或注解用于配置和原始映射,将应用层的接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java 对象)映射成数

数据库中的记录,消除了大部分的 JDBC 代码和参数的手工设置以及结果集的检索。不仅实现领域层的对象范例和数据源层的关系范例之间的信息交互,且使用 XML 配置文件将访问数据层的数据库相关配置及 sql 语言和应用层的程序代码分离,很大程度上提高了程序的可维护性、可扩展性和可移植性。

Mybatis 的功能架构分为为开发人员调用的 API 接口层、根据配置的 xml 文件完成具体的数据操作的数据处理层和为上层提供功能支撑的基础支撑层,具体的架构层如下图所示:



图 1 MyBatis 架构图

## 2.2 XML 概述

XML (Extensible Markup Language, 可扩展标记语言), 作为标准通用标记语言 (SGML) 的子集, 可以用来标记数据、定义数据类型, 是一种允许用户对自己的标记语言进行定义的源语言, 可提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。凭借其良好的可扩展性和强大的数据交互性, 在异构平台的信息交互中有着广泛的应用。XML 具有自描述性、可扩展性、内容与显示可分离和异构数据集成等特点。

XML 并不仅仅包括 XML 标记语言, 它同时还包括了很多相关的规范, 比如文档模式技术、文档样式技术、文档查询技术、文档解析技术、文档链接技术以及文档定位技术等。基于 XML 的这些规范, 还有很多高层的应用协议, 比如 SOAP (Simple Object Access Protocol) 和 BizTalk 等<sup>[3]</sup>。

## 3 XML 在 Mybatis 中的应用

MyBatis 使用简单的 XML 或注解用于配置和原始映射, 将接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象) 映射成数据库中的记录。现在以一个应用实例和动态 sql 为例来讲述 mybatis 的应用。

### 3.1 XML 在 MyBatis 中的应用实例

在 MyBatis 应用的基本流程为, 构建 SqlSessionFactory 实例, 获取数据库配置信息, 根据映射文件访问 java 中定义中的接口信息处理 java 与数据库直接的交互。具体如下:

\* 构建 SqlSessionFactory 实例及其应用

MyBatis 应用程序主要都是使用 SqlSessionFactory 实例的, 一个 SqlSessionFactory 实例可以通过 SqlSessionFactoryBuilder 获得。SqlSessionFactoryBuilder 可以从一个 xml 配置文件或者一个预定义的配置类的实例获得, 根据配置文件构建 SqlSessionFactory 实例, 创建会话, 即可操作数据库文件。具体的 JAVA 代码为:

代码清单 1: SqlSessionFactoryTest

// 读取 XML 配置文件, 获取数据库信息

```
String configurl = "MyBatis-Configuration.xml";
Reader reader = Resources.getResourceAsReader(configurl);
// 获取 SqlSessionFactory 实例 s
SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();
SqlSessionFactory factory = builder.build(reader);
// 打开数据库访问回话
SqlSession session = factory.openSession();
```

\* MyBatis 配置文件

根据 mybatis 标准中的 dtd 文件规定: 一个 MyBatis-Configuration.xml 文件应包括配置环境信息的 environments 元素和与映射相关的 mappers 元素等。Environment 元素包括一个获取数据库连接实例的数据源, 以及一个控制事务的 TransactionManager (事务管理)。Mapper 元素中定义 sql 映射文件的路径。代码实例为:

代码清单 2: MyBatis-Configuration.xml

```
<transactionManager type="JDBC"/>
<dataSource type="POOLED">
    <property name="driver" value="${driver}"/> <!-- 数据库驱动 -->
    <property name="url" value="${url}"/> <!-- 数据库路径 -->
    <property name="username" value="${username}"/> <!-- 用户名 -->
    <property name="password" value="${password}"/> <!-- 密码 -->
</dataSource>
```

<!-- sql 映射文件 -->

```
<mapper resource="com/mybatis/config/DaoMapper.xml"/>
```

\* MyBatis 映射文件

根据 mybatis 标准中的 dtd 文件规定: 一个映射文件包含多个 mapper 节点, 节点中定义对应的 java 接口名称, 其子节点包含 select、insert、update、delete 等元素, 分别用于执行 SQL 中的 select、insert、update、delete, 这些子节点的属性基本类似, 包含作为标示符的 id, 规定返回值类型的 resultType, 规定参数类型的 parameterType 等。以下为代码详解:

代码清单 3: DaoMapper.xml

```
<!-- 对应的 java 接口为 public List<User> selectAll(); -->
<select id="selectAll" resultType="com.mybatis.config.User">
    select * from user
</select>
<!-- 对应的 java 接口为 public int insert(User user); -->
<insert id="insert" parameterType="com.mybatis.config.User">
    insert into user(userName, password, comment)
    values(#{userName}, #{password}, #{comment})
</insert>
```

### 3.2 MyBatis 中的动态 sql 中的应用

MyBatis 的动态 SQL 是基于 OGNL 表达式的, 可以利用 XML 文档可扩展性和结构性实现。MyBatis 中用于实现动态

SQL 的 XML 元素主要有:if、choose (when, otherwise)、trim、where、set、foreach 等,可借助这些元素将相关的逻辑控制在 XML 文档中,便于控制和修改。

#### \* Where 元素

where 元素的作用是在 sql 需要拼写 where 元素的地方输出一个 where,且 MyBatis 会智能处理不同条件下 where 输出。如果所有的条件都不满足那么 MyBatis 就会查出所有的记录,且会根据拼写出的条件忽略掉首个 and 关键字,保证输出的 sql 语句是规范的。如下面的代码,如果两个条件都不满足,则输出的 sql 为“select \* from User”,如果条件 1 不满足,条件 2 满足,则输出的 sql 为“select \* from User where id=#{id}”。相对于 JDBC 的拼写 sql,更为简洁与易维护。

代码清单 4:whereTest.xml

```
1. <select?id="whereTest"?parameterType="String"?resultType="com.mybatis.config.User">
2.   select*from User
3. <where>
4.   <if test="name !=?null">
5.     name = #{name}
6.   </if>
7.   <if test="id != null">
8.     and id = #{id}
9.   </if>
10. </where>
11.</select>
```

#### \* Set 元素

set 元素主要是用在更新操作的时候,它的主要功能和 where 元素类似,主要是在包含的语句前输出一个 set,会智能地忽略掉最后一个逗号,但如果包含的所有的子元素都为空的话,语句就会报错。以下面的一段代码为例:如果条件 1 满足,条件 2 不满足,则输入的 sql 语句为“update user set name = #{name} where id = #{id}”;如果两个条件都不满足,则输出的 sql 为“update user where id = #{id}”,显然是错误的。

代码清单 5:setTest.xml

```
1.<update id="setTest" parameterType="String">
2.  update user
3. <set>
4.   <if test="name != null">
5.     name = #{name},
6.   </if>
```

```
7.   <if test="nl != null">
8.     nl = #{nl},
9.   </if>
10. </set>
11. where id = #{id}
12.</update>
```

#### \* foreach 应用

foreach 的主要用在构建 in 条件中,可以作为 SQL 语句的迭代集合。foreach 元素的属性主要有 item、index、collection、open、separator、close。item 表示集合中每一个元素进行迭代时的别名,index 用于指定每次迭代位置的名称,open 表示迭代的开始,separator 表示迭代之间的分隔符,close 表示迭代的结束,collection 属性,可以传递 List、array 数组、Map 类型 3 种指定类型的参数,且必须指定。以下以单参数 List 类型的代码:

代码清单 6:foreachTest.xml

```
1.<!-- List<User> users = userMapper.foreachTest(list)-->
2.<select id="foreachTest" resultType="User">
3.   select* from user where id in
4.   <foreach collection="list" index="index" item="item"
open="(" separator="," close=")">??
5.     #{item}
6.   </foreach>
7.</select>
```

## 4 结束语

本文通过 MyBatis 框架的简单介绍和实际应用,证明其在处理异构平台信息交互时,具有很好的可伸缩性、灵活性、易维护性,且简单易用,会在持久层的应用中有着越来越广泛的应用。

#### 参考文献:

- [1] Alur D.J2EE 核心模式[M].北京:机械工业出版社,2005.
- [2] Kumar B V.深入浅出 J2EE 架构[M].北京:清华大学出版社,2006.
- [3] 麦克劳夫林.JAVA 与 XML[M].北京:中国电力出版社,2011.

(上接第 85 页)

务。目的服务处理结果返回的过程同上。

## 5 结束语

研究窄带通信环境下指挥控制系统软件的服务化解决方法是实现指挥控制系统网络化的关键技术之一。将商用的 web 服务标准进行优化改造,是实现窄带通信环境下指挥控制系统软件服务化的重要方法。本文研究的内容也是基于以上思想。另外,对 SOAP 消息进行压缩与差分处理也是研究的重要方法。

#### 参考文献:

- [1] 王珩,黄松华,易侃,等.指挥控制服务交互方法研究[J].指挥信息系统与技术,2010(6):18-21.
- [2] 舒振,马建威,罗雪山.基于 SOA 的新型指挥控制系统体系框架及开发方法[J].兵工自动化,2010,32(8):65-69.
- [3] 邓铁山,王晓东.SOAP 在实时 Web 服务中的性能评估与改进[J].计算机应用,2009,33(9):25-31.

作者：[贺军](#)，[焦荣](#)，[HE Jun](#)，[JIAO Rong](#)  
作者单位：[贺军, HE Jun \(北方自动控制技术研究所, 太原, 030006\)](#)，[焦荣, JIAO Rong \(北京汉林信通信息技术有限公司, 北京, 100085\)](#)  
刊名：[火力与指挥控制](#)[ISTIC](#)[PKU](#)  
英文刊名：[Fire Control & Command Control](#)  
年，卷(期)：2013(z1)

## 参考文献(3条)

1. [Alur D](#) [J2EE核心模式](#) 2005
2. [Kumar B V](#) [深入浅出J2EE架构](#) 2006
3. [麦克劳夫林](#) [JAVA与XML](#) 2011

引用本文格式：[贺军](#)，[焦荣](#)，[HE Jun](#)，[JIAO Rong](#) [基于XML应用的MyBatis技术](#)[期刊论文]-[火力与指挥控制](#) 2013(z1)