



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌



MyBatis映射文件深入

目 录 Contents

◆ MyBatis映射文件深入

1.MyBatis映射文件深入

1.1 动态sql语句

1. 动态sql语句概述

Mybatis 的映射文件中，前面我们的 SQL 都是比较简单的，有些时候业务逻辑复杂时，我们的 SQL是动态变化的，此时在前面的学习中我们的 SQL 就不能满足要求了。

参考的官方文档，描述如下：

Dynamic SQL

One of the most powerful features of MyBatis has always been its Dynamic SQL capabilities. If you have any experience with JDBC or any similar framework, you understand how painful it is to conditionally concatenate strings of SQL together, making sure not to forget spaces or to omit a comma at the end of a list of columns. Dynamic SQL can be downright painful to deal with.

While working with Dynamic SQL will never be a party, MyBatis certainly improves the situation with a powerful Dynamic SQL language that can be used within any mapped SQL statement.

The Dynamic SQL elements should be familiar to anyone who has used JSTL or any similar XML based text processors. In previous versions of MyBatis, there were a lot of elements to know and understand. MyBatis 3 greatly improves upon this, and now there are less than half of those elements to work with. MyBatis employs powerful OGNL based expressions to eliminate most of the other elements:

- if
- choose (when, otherwise)
- trim (where, set)
- foreach

1.MyBatis映射文件深入

1.1 动态sql语句

2. 动态 SQL 之<if>

我们根据实体类的不同取值，使用不同的 SQL语句来进行查询。比如在 id如果不为空时可以根据id查询，如果 username 不同空时还要加入用户名作为条件。这种情况在我们的多条件组合查询中经常会碰到。

```
<select id="findByCondition" parameterType="user" resultType="user">
  select * from User
  <where>
    <if test="id!=0">
      and id=#{id}
    </if>
    <if test="username!=null">
      and username=#{username}
    </if>
  </where>
</select>
```

1.MyBatis映射文件深入

1.1 动态sql语句

2. 动态 SQL 之<if>

当查询条件id和username都存在时，控制台打印的sql语句如下：

```
... ..  
//获得MyBatis框架生成的UserMapper接口的实现类  
UserMapper userMapper = sqlSession.getMapper(UserMapper.class);  
User condition = new User();  
condition.setId(1);  
condition.setUsername("lucy");  
User user = userMapper.findByCondition(condition);  
... ..
```

```
- Created connection 586084331.  
- Setting autocommit to false on JDBC Connection [com.mysql.jdbc.:  
- ==> Preparing: select * from User WHERE id=? and username=?  
- ==> Parameters: 1(Integer), lucy(String)  
- <==          Total: 1
```

1.MyBatis映射文件深入

1.1 动态sql语句

2. 动态 SQL 之<if>

当查询条件只有id存在时，控制台打印的sql语句如下：

```
... ..  
//获得MyBatis框架生成的UserMapper接口的实现类  
UserMapper userMapper = sqlSession.getMapper(UserMapper.class);  
User condition = new User();  
condition.setId(1);  
User user = userMapper.findByCondition(condition);  
... ..
```

```
- Setting autocommit to false on JDBC Connection [com.mysql.  
- ==> Preparing: select * from User WHERE id=?  
- ==> Parameters: 1(Integer)  
- <==      Total: 1
```

1.MyBatis映射文件深入

1.1 动态sql语句

3. 动态 SQL 之<foreach>

循环执行sql的拼接操作，例如：SELECT * FROM USER WHERE id IN (1,2,5)。

```
<select id="findByIds" parameterType="list" resultType="user">
  select * from User
  <where>
    <foreach collection="array" open="id in(" close=")" item="id" separator=",">
      #{id}
    </foreach>
  </where>
</select>
```

1.MyBatis映射文件深入

1.1 动态sql语句

3. 动态 SQL 之<foreach>

测试代码片段如下：

```
... ..  
//获得MyBatis框架生成的UserMapper接口的实现类  
UserMapper userMapper = sqlSession.getMapper(UserMapper.class);  
int[] ids = new int[]{2,5};  
List<User> userList = userMapper.findByIds(ids);  
System.out.println(userList);  
... ..
```

```
11:21:02,237 DEBUG findByIds:159 - ==> Preparing: select * from User WHERE id in( ?, ? )  
11:21:02,262 DEBUG findByIds:159 - ==> Parameters: 2(Integer), 5(Integer)  
11:21:02,280 DEBUG findByIds:159 - <== Total: 2  
[User{id=2, username='tom', password='123'}, User{id=5, username='haohao', password='123'}]
```


1.MyBatis映射文件深入

1.1 动态sql语句

3. 动态 SQL 之<foreach>

foreach标签的属性含义如下:

<foreach>标签用于遍历集合, 它的属性:

- collection: 代表要遍历的集合元素, 注意编写时不要写#{}
- open: 代表语句的开始部分
- close: 代表结束部分
- item: 代表遍历集合的每个元素, 生成的变量名
- separator: 代表分隔符

1. MyBatis映射文件深入

1.2 SQL片段抽取

Sql 中可将重复的 sql 提取出来，使用时用 include 引用即可，最终达到 sql 重用的目的

<!--抽取sql片段简化编写-->

```
<sql id="selectUser" select * from User</sql>
```

```
<select id="findById" parameterType="int" resultType="user">
```

```
<include refid="selectUser"></include> where id=#{id}
```

```
</select>
```

```
<select id="findByIds" parameterType="list" resultType="user">
```

```
<include refid="selectUser"></include>
```

```
<where>
```

```
<foreach collection="array" open="id in(" close=")" item="id" separator=",">
```

```
  #{id}
```

```
</foreach>
```

```
</where>
```

```
</select>
```

1.MyBatis映射文件深入

1.3 知识小结

MyBatis映射文件配置:

<select>: 查询

<insert>: 插入

<update>: 修改

<delete>: 删除

<where>: where条件

<if>: if判断

<foreach>: 循环

<sql>: sql片段抽取



传智播客旗下高端IT教育品牌