XML模式的开发顺序

1.创建DO对象

2.创建DAO接口，配置@Mapper注解

3.创建XML文件，并完成resultMap 配置

4.创建DAO接口方法

5.创建对应的XML语句

2.1 MyBatis XML 配置

添加配置

mybatis.mapper-locations=classpath:com/youkeda/comment/dao/\*.xml

这个配置用于指定MyBatis Mapper XML文件路径，一般来说这个路径和DAO的包路径一致。

如果有多个路径用 **，** 隔开添加：

mybatis.mapper-locations=classpath:com/youkeda/dao/\*.xml,classpath:com/youkeda/comment/dao/\*.xml

2.2 MyBatis XML Mapper

UserDAO.xml文件路径存放在resource目录下，

src/main/resources/com/youkeda/comment/dao/UserDAO.xml

**1.头信息**

创建完xml文件后，需要复制下面的内容，这个是固定格式的：

**<?xml version="1.0" encoding="UTF-8"?>**

**<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"**

**"http://mybatis.org/dtd/mybatis-3-mapper.dtd">**

**2.mapper根节点**

有了头信息后，继续添加mapper节点，比如：  
**<?xml version="1.0" encoding="UTF-8"?>**

**<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"**

**"http://mybatis.org/dtd/mybatis-3-mapper.dtd">**

<mapper namespace="com.youkeda.comment.dao.UserDAO">

</mapper>

Mapper这个节点有一个属性namespace，这个是命名空间的含义，一般就是我们这个mapper对应的DAO接口的全称。

**3.resultMap**

ResultMap用于处理表和DO对象的属性映射，确保表中的每个字段都有属性可以匹配

**<?xml version="1.0" encoding="UTF-8"?>**

**<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"**

**"http://mybatis.org/dtd/mybatis-3-mapper.dtd">**

<mapper namespace="com.youkeda.comment.dao.UserDAO">

<resultMap id="userResultMap" type="com.youkeda.comment.dataobject.UserDO">

<id column="id" property="id"/>

<result column="user\_name" property="userName"/>

</resultMap>

</mapper>

\*id 唯一标示，命名规则是xxxResultMap

\*type 对应的DO类完整路径

3.1 resultMap 子节点

最主要是id、result这两个子节点

\*id 设置数据库主键字段信息，column属性对应的是表的字段名称，property对应的是DO属性名称

\*result 设置数据库其它字段信息，column属性对应的是表的字段名称，property对应的是DO属性名称

完整的resultMap：

<resultMap id="userResultMap" type="com.youkeda.comment.dataobject.UserDO">

<id column="id" property="id"/>

<result column="user\_name" property="userName"/>

<result column="pwd" property="pwd"/>

<result column="nick\_name" property="nickName"/>

<result column="avatar" property="avatar"/>

<result column="gmt\_created" property="gmtCreated"/>

<result column="gmt\_modified" property="gmtModified"/>

</resultMap>

2.3 MyBatis XML Insert 语句

Mapper内insert 节点：

<insert id="add" parameterType="com.youkeda.comment.dataobject.UserDO" >

INSERT INTO user (user\_name, pwd, nick\_name,avatar,gmt\_created,gmt\_modified)

VALUES(#{userName}, #{pwd}, #{nickName}, #{avatar},now(),now())

</insert>

**属性id**：xml内唯一标示，此处改为“add”

**属性parameterType**：用于传递参数类型，一般是和UserDAO.add方法的参数一致

如果想同时获得插入的主键id值，还需要配置useGeneratedKeys、keyProperty，详见@Options

<insert id="add" parameterType="com.youkeda.comment.dataobject.UserDO" useGeneratedKeys="true" keyProperty="id">

INSERT INTO user (user\_name, pwd, nick\_name,avatar,gmt\_created,gmt\_modified)

VALUES(#{userName}, #{pwd}, #{nickName}, #{avatar},now(),now())

</insert>

2.4 MyBatis XML Update/Delete 语句

**Update**

[1.去掉DAO中update方法的@Update](mailto:1.去掉DAO中update方法的@Update)注解

2.再DAO.xml中添加update节点（无@Options内属性）

<update id="updateStock" parameterType="java.lang.Long">

update product

set gmtModified = now()

<if test="stock!=null">

stock = #{stock}

</if>

where id = #{id}

</update>

**Delete**

1.改造DAO类中的方法

2.修改xml文件（在方法参数带@Param时，xml节点可以不写parameterType属性）

2.5 MyBatis XML Select 语句

[1.去掉DAO类中方法的@Select](mailto:1.去掉DAO类中方法的@Select)

2.XML中增加Select语句

<select id="findAll" resultMap="userResultMap">

select \* from user

</select>

<select id="findByUserName" resultMap="userResultMap">

select \* from user where user\_name=#{userName} limit 1

</select>

resultMap属性：一般配置为该XML文件下的resultMap节点的id值。