# 动态SQL简介

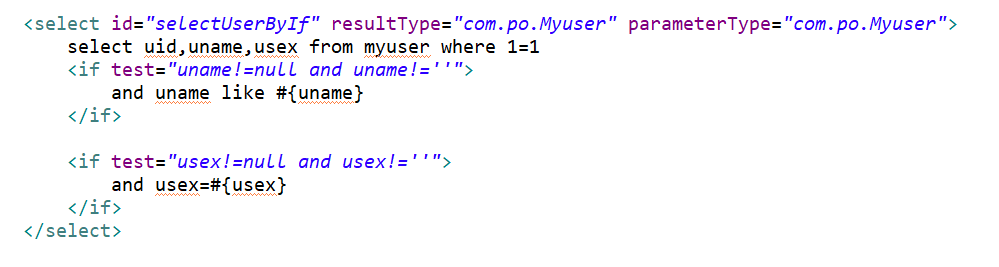
1）开发人员通常需要根据需求手动拼接SQL语句，这是一个及其麻烦的工作，而MyBatis 提供了对SQL语句动态组装的功能，恰能解决这一问题。

2）MyBatis的动态SQL标签与JSTL相似，常用的有<if>、<choose>、<when>、<otherwise>、 <trim>、<where>、<set>、<foreach>、<bind>等标签。

3）动态SQL的主要作用是根据需要动态地拼凑SQL语句。

# <if>标签

<if>标签类似于if语句，其作用是根据传递进来的参数的情况来拼凑SQL语句。如下：



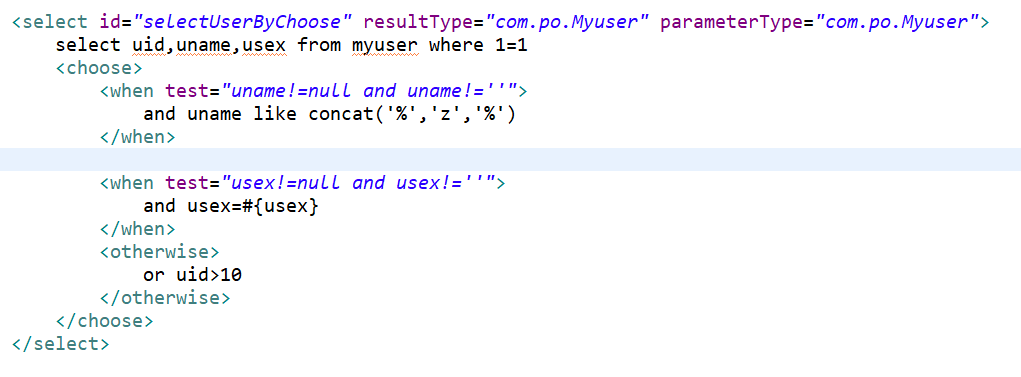


判断Myuser类对象的uname属性是否不为空或’ ‘，如果为true，则进行拼凑。

判断Myuser类对象的usex属性是否不为空或’ ‘，如果为true，则进行拼凑。

# <choose>、<when>、<otherwise>标签

对于<if>标签，判断之后，如果后面依然存在<if>标签，则会继续执行进行判断。但有 些时候不想用到所有的条件语句，而只想从中取一、二，针对这种情况，MyBatis提供 了<choose>标签，类似于Java中的switch语句。同时<choose>标签需要与<when>标签、 <otherwise>标签共同使用。如下：

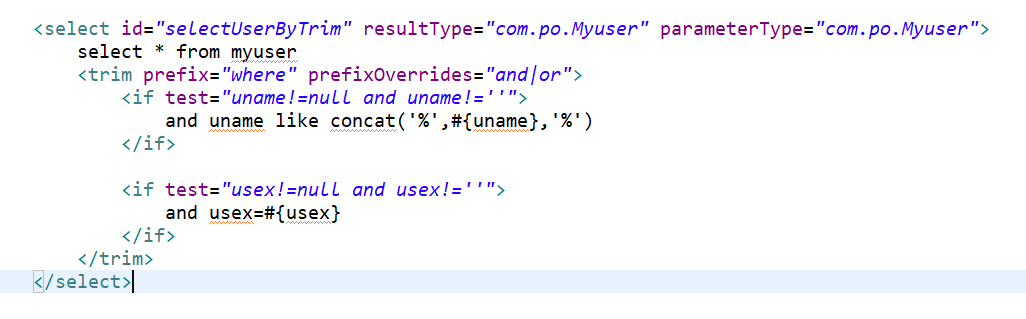


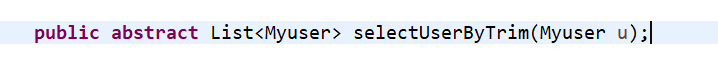
进入<choose>标签后，进行<when>标签的判断，如果有一个when标签符合，则跳出 <choose>标签，如果没有，则执行<otherwise>标签，进行SQL语句的拼凑。

# <tirm>标签

1）<tirm>标签的的功能是可以在自己包含的内容前加上某些前缀，也可以在其后面加 上某些后缀，与之对应的属性是prefix和suffix；可以把包含内容的首部某些内容覆 盖，也可以把尾部的某些内容覆盖，对应的属性是prefixOverrides和suffixOverrides。

2）正因为<tirm>标签有这样的功能，所以也可以非常简单地利用<trime>标签来代替 <where>标签的功能。





在上面的例子中，<tirm>标签的前缀属性是’ where ’，前缀覆盖属性是’ and ‘ 或者 ‘ or ‘

当<trim>标签的第一个<if>标签满足时，拼凑成的SQL语句为：

**select \* from myuser where uname like concat( ‘%’ , #{uname} , ‘%’ )**

当<trim>标签中<if>标签都不满足时，此<trim>标签作废，拼凑成的SQL语句为：

**select \* from myuser**

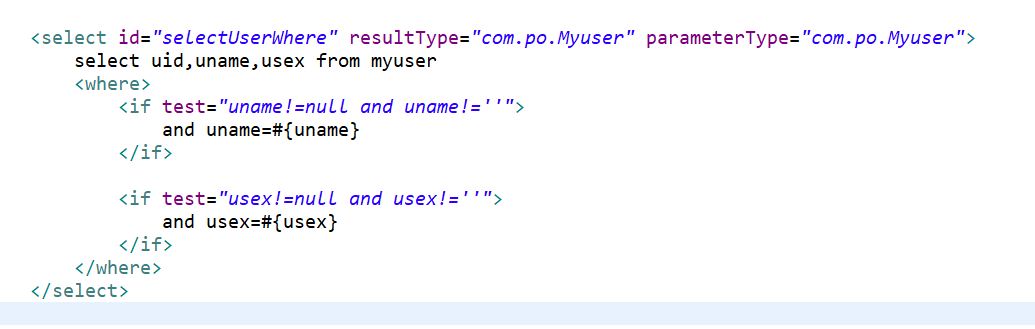
# <where>标签

1）<where>标签的作用是转换为一条where子句。

2）<where>标签内部通常嵌套<if>、<choose>标签，如果内部所有的<if>、<choose>标 签都不满足，则该<where>标签作废，MyBatis会查询出所有的记录。

3）使用<where>标签时如果拼凑的where子句是以and或or开头的，MyBatis会将它 忽略；此外，在<where>标签中也不需要考虑空格的问题，MyBatis将智能加上。

如下：





在上面这个例子中，

如果第一个<if>标签满足，第二个<if>不满足，则该条SQL语句将拼凑为：

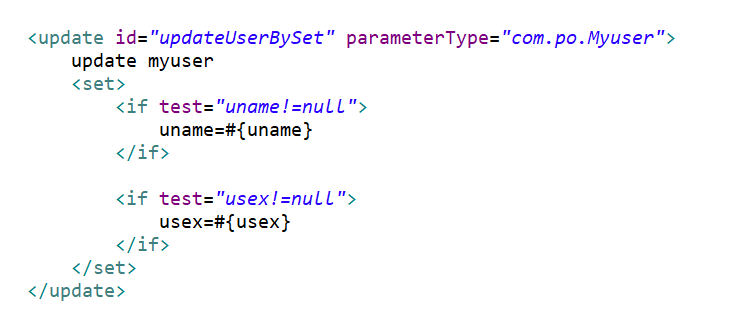
**select \* from myuser where uname = # { uname }**

如果都不满足，则该条语句将拼凑为：

**select \* from myuser**

# <set>标签

<set>标签用于更新语句<update>中，作用是生成set子句，完成动态更新列的功能。

如下：



在上面这个例子中，如果<if>标签都满足，则拼凑的SQL语句为：

**update myuser set uname = # {uname} usex = #{usex}**

# <foreach>标签

1. <foreach>标签用于构建in条件表达式，它可以在SQL语句迭代一个集合。
2. <foreach>标签的属性主要有item、index、collecation、open、seperator、close

·item属性表示集合中每一个元素进行迭代时的别名

·index指定一个名字，用于表示在迭代过程中每次迭代到的位置

·open表示该语句以什么开始

·seperator表示在每次迭代之间以什么符号作为分隔符

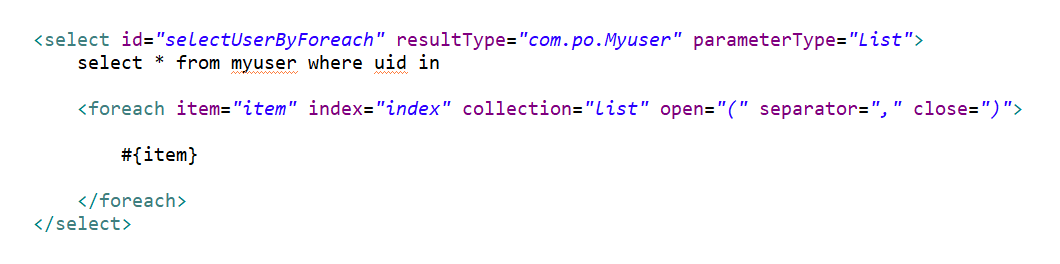
·close表示以什么结束

在<foreach>标签中最重要的是collection属性，该属性是必选的，但是在不同情况下该属性的值是不一样的，主要有以下3种情况。

·如果传入的是单参数且参数类型是一个List，collecation属性值为list

·如果传入的是单参数且参数类型是一个array数组，collection的属性值为 array

·如果传入的参数时是个，需要把它们封装成一个Map，当然单参数也可以 封装成Map。Map是key的参数名，collection属性值是传入的List或array 在自己封装的Map中的key。



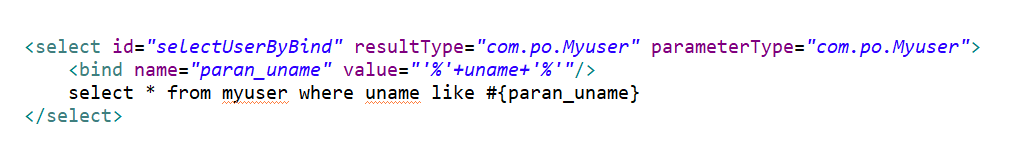


在上面的例子中，使用了<foreach>构建in表达式，如果uidList包含了1、2， 则拼凑的SQL语句为：

**select \* from myuser where uid in ( 1 , 2 )**

# <bind>标签

1. <bind>标签用于模糊查询。
2. 在进行模糊查询时，如果使用’${ } ‘拼凑字符串，则无法防止SQL注入问题；如果使 用字符串拼接函数或者连接符号，又需要考虑兼容性的问题，因为不同数据库的 拼接函数或连接符号不同，例如MySQL使用concat函数、Oracle使用连接符号”||” 。对此，MyBatis提供了<bind>标签解决此类问题。





在上面这个例子中，，如果使用的是MySQL数据库，则拼凑的SQL语句为：

**select \* from myuser where uname like concat( ‘%’ , #{uname} , ‘%’ )**

所以，在MyBatis中，如果在查询数据时涉及到了模糊查询，则推荐使用<bind>标 签来完成。