3.1 MyBatis XML 条件语句

**If语句**

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

update user set

<if test="nickName != null">

nick\_name=#{nickName},gmt\_modified=now()

</if>

where id=#{id}

</update>

**Set语句（包裹if语句，防止if判断导致null错误）**

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

update user

<set>

<if test="nickName != null">

nick\_name=#{nickName},

</if>

<if test="avatar != null">

avatar=#{avatar},

</if>

gmt\_modified=now()

</set>

where id=#{id}

</update>

**If+select**

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

select \* from user where

<if test="keyWord != null">

user\_name like CONCAT('%',#{keyWord},'%')

or nick\_name like CONCAT('%',#{keyWord},'%')

</if>

<if test="time != null">

and gmt\_created <![CDATA[ >= ]]> #{time}

</if>

</select>



**@GetMapping("/user/search")**

**@ResponseBody**

**public** List<UserDO> **search**(**@RequestParam("keyWord")** String keyWord,

**@RequestParam("time")**

**@DateTimeFormat(pattern = "yyyy-MM-dd HH:mm:ss")**

LocalDateTime time) {

**return** userDAO.search(keyWord, time);

}

**Where（包裹if语句，防止if条件内为否导致null错误）**

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

select \* from user

<where>

<if test="keyWord != null">

user\_name like CONCAT('%',#{keyWord},'%')

or nick\_name like CONCAT('%',#{keyWord},'%')

</if>

<if test="time != null">

and gmt\_created <![CDATA[ >= ]]> #{time}

</if>

</where>

</select>

3.2 MyBatis XML 循环语句

Foreach语句批量操作

批量插入user，DAO方法：

**int** **batchAdd**(**@Param("list")** List<UserDO> userDOs);

foeeach语法：

<insert id="batchAdd" parameterType="java.util.List" useGeneratedKeys="true" keyProperty="id">

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

VALUES

<foreach collection="list" item="it" index="index" separator =",">

(#{it.userName}, #{it.pwd}, #{it.nickName}, #{it.avatar},now(),now())

</foreach >

</insert>

Foreach属性：

\*collection指定集合的上下文参数名称，比如这里的list，对应的是@Param（“list”）

\*item 指定遍历中的每一个数据的变量，一般用it 命名，所以可以使用it.userName获取具体的值

\*index 集合的索引值，从0开始

\*separator（分离器） 遍历每条记录并添加分隔符

Foreach查询

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

select \* from user

<where>

id in

<foreach item="item" index="index" collection="ids"

open="(" separator="," close=")">

#{item}

</foreach>

</where>

</select>

\*open 表示的是节点开始时自定义的分隔符

\*close 表示的是节点结束时自定义的分隔符

上面的SQL执行后会变成

**select** \* **from** **user** **where** id **in** (?,?,?)