

trim标签

trim标签中是整个字符串拼串后的结果

prefix="":前缀，prefix给拼串后的整个字符串加一个前缀

prefixOverrides="":前缀覆盖：去掉整个字符串前面多余的字符

suffix="":后缀，suffix给拼串后的整儿字符串加一个后缀

suffixOverrides="" 后缀覆盖：去掉整个字符串后面多余的字符

foreach标签

举例：

```
<foreach collection="list" item="item" separator="," open="(" close=")" index="index">
  #{item}
</foreach>
```

- 1.collection:指定要遍历的集合，list类型的参数会特殊处理封装到map中，map的key就叫list
- 2.item：将当前遍历出来的元素赋值给指定的变量
- 3.sepatator:每个元素之间的分隔符
- 4.open:遍历出所有结果拼接一个开始的符号
- 5.close:遍历出所有结果拼接一个结束的符号
- 6.index:索引，遍历list的时候index就是索引，item就是当前值；遍历map的时候index表示的就是map的key,item就是map的值
- 7.#{变量名}就能取出变量的值也就是当前遍历出的元素

缓存

一级缓存 即本地缓存 sqlSession级别的缓存，一级缓存是一直开启的

与数据库同一次会话期间查询到的数据会放到本地缓存中，以后如果需要获取相同的数据，直接从缓存中拿，没必要再去查询数据库。

二级缓存 全局缓存，基于namespace级别的缓存，一个namespace对应一个二级缓存。

工作机制：

- 1.一个会话，查询一条数据，这个数据就会被放在当前会话的一级缓存中；
- 2.如果会话关闭：一级缓存中的数据会被保存到二级缓存中；新的会话查询信息，就可以参照二级缓存
- 3.不同namespace查出的数据会放在自己对应的缓存中（map）

效果：数据会从二级缓存中获取

查春的数据都会被默认先放在一级缓存中。

只有会话提交或关闭以后，一级缓存中的数据才会移动到二级缓存中使用；

- 1.开启全局二级缓存配置

```
<setting name="cacheEnabled" value="true"/>
```

2.去mapper.xml中配置使用二级缓存

```
<cache></cache>
```

```
<cache eviction="FIFO" flashInterval="6000" readOnly="false"></cache>
<!--
eviction: 缓存的回收策略:
    LRU: 最近最少使用, 移除最长时间不被使用的对象
    FIFO: 先进先出, 按对象进入缓存的顺序来移除它们。
    SOFT: 软引用, 移除基于垃圾回收器状态和软引用规则的对象。
    WEAK: 弱引用, 更积极的移除基于垃圾收集器状态和弱引用规则的对象。
    默认的是LRU
flushInterval: 缓存刷新间隔
    设置缓存多长时间清空一次, 默认不清空, 设置一个毫秒值
readOnly: 是否只读
    true: 只读, mybatis认为所有从缓存中获取数据的操作都是只读操作, 不会修改数据。
    mybatis为了加快获取速度, 直接就会将数据在缓存中的引用交给用户, 不安全, 速度快。
    false: 非只读, mybatis觉得获取的数据有可能被修改。安全, 速度慢
size: 缓存存放多少元素。
type: 指定自定义缓存的全类名, 实现Cache接口即可。

-->
```

3.我们的pojo需要实现序列化接口Serializable

和缓存有关的设置/属性：

- 1.cacheEnabled=true(默认), 如果设置成false,即关闭缓存 (二级缓存关闭, 一级缓存一直可用)
- 2.每个select标签都有userCache,默认是true,如果设置成false,即关闭缓存 (二级缓存关闭, 一级缓存一直可用)
- 3.每个增删改标签都有flushCache,默认是true,清除缓存,(一级二级缓存都被清除),但是select标签, flushCache默认值为false,如果设置成true,每次查询之后都会清空缓存, 缓存是没有被使用的
- 4.sqlSession.clearCache();只是清除当前session的一级缓存。
- 5.localCacheScope:本地缓存作用域 (一级缓存, 默认是session),当前会话的所有数据都保存在当前会话缓存中, 如果设置为statement,即禁用掉一级缓存。

新会话进入会先去查找二级缓存中是否有对应的数据



