

2-19 通用Mapper接口所封装的常用方法

通用Mapper接口所封装的常用方法

在上一节中我们是用了Mybatis逆向工具生成了 pojo 、 mapper. java 以及 mapper. xml ，当然我们也提到了 MyMapper 这个接口，这个接口中封装了很多数据法，这一节我们一起来看一下吧。

建议大家`在idea中打开相关源码，以此参照下面内容一起阅读噢~`

1.首先先来看一下 MyMapper 所继承的父类，如：

```
interface MyMapper<T> extends Mapper<T>, MySqlMapper<T>
```

这里有两个父类， Mapper<T> 与 MySqlMapper<T> ，我们可以打开 MySqlMapper<T> 看一下：

```
interface MySqlMapper<T> extends InsertListMapper<T>, InsertUseGeneratedKeysMapper<T> {}
```

这里面又继承了两个mapper，从类名上可以看得出来，是用于操作数据库的，这两个类里又分别包含了如下方法，简单归类一下：

方法名	操作	备注
insertList(list)	数据批量插入	主键须自增
insertUseGeneratedKeys(record)	插入表数据	主键须自增

很明显，在传统JavaWeb开发，这两个方法使用是没有问题的，但是我们的数据库表主键设计肯定是全局唯一的，所以不可能使用自增长id（如何设计全局唯一，在后续课程里有具体的讲解），所以这两个方法在我们开发过程中是不会使用的，这一点需要注意噢~！

2.随后再来看一下 Mapper<T> 中所继承的父类，如下：

```
interface Mapper<T> extends BaseMapper<T>, ExampleMapper<T>, RowBoundsMapper<T>,
```

分别来看一下各个父类中的方法有些啥？

- BaseMapper<T>

类	方法	操作
BaseSelectMapper	T selectOne(T record)	根据实体类中的属性查询表数据，返回单个实体
	List select(T record)	根据实体类中的属性查询表数据，返回符合条件的list
	List selectAll()	返回该表所有记录
	int selectCount(T record)	根据条件查询记录数
	T selectByPrimaryKey(Object key)	根据主键查询单挑记录
BaseInsertMapper	boolean existsWithPrimaryKey(Object key)	查询主键是否存在，返回true或false
	int insert(T record)	插入一条记录，属性为空也会保存
	int insertSelective(T record)	插入一条记录，属性为空不保存，会使用默认值
BaseUpdateMapper	int updateByPrimaryKey(T record)	根据实体类更新数据库，属性有null会覆盖原记录
	int updateByPrimaryKeySelective(T record)	根据实体类更新数据库，属性有null改属性会忽略
BaseDeleteMapper	int delete(T record)	根据实体类中属性多条件删除记录
	int deleteByPrimaryKey(Object key)	根据主键删除记录

- ExampleMapper<T>，Example类是用于提供给用户实现自定义条件的，也就是 where 条件，主要方法见如下表格：