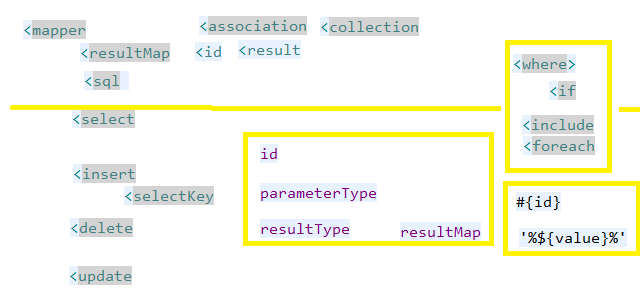
# 映射配置文件Mapper.xml

通过parameterType完成输入映射，通过resultType和resultMap完成输出映射。

## 概述



## 映射类型



## resultType映射pojo，map使用方法

结果映射时：查询的列名和映射的pojo属性名

1. 只要有一个一致，就会创建pojo对象，没对应的返回空，通常pojo属性数量大于列。

2.全部不一致，则不会创建pojo对象；

## resultMap映射使用方法

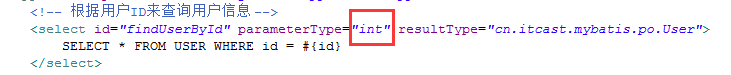
如果查询出来的列名和属性名不一致，通过定义一个resultMap将列名和pojo属性名之间作一个映射关系。

1. 定义resultMap
2. 使用resultMap作为statement的输出映射类型。

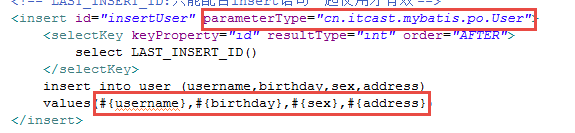
如果sql查询列名和最终要映射的pojo的属性名不一致，使用resultMap将列名和pojo的属性名做一个对应关系 （列名和属性名映射配置）

# parameterType--输入映射

## 传递简单类型



## 传递POJO类型（推荐）



## 传递自定义类（扩展类）

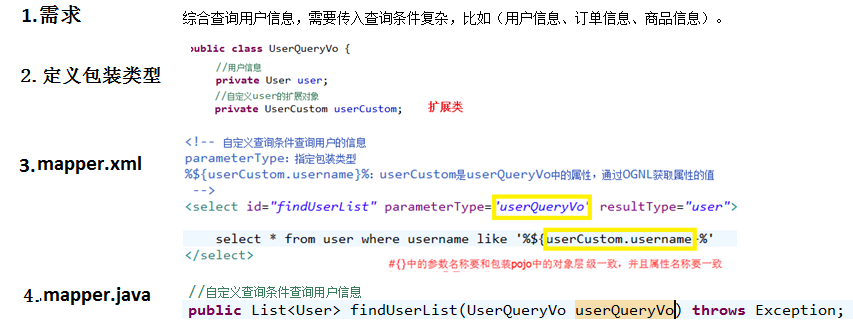
两种方式：1.继承pojo并且新增属性

2.自定义一个pojo类

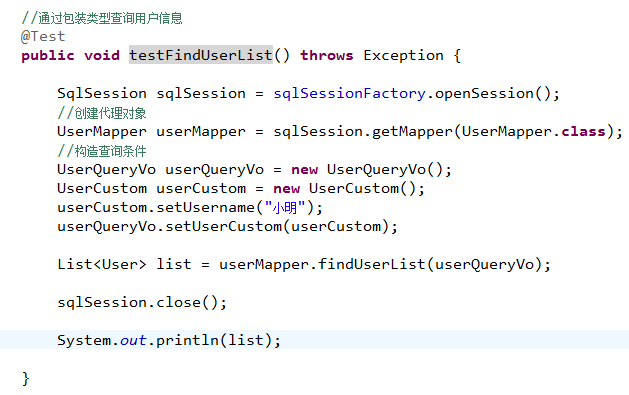


## 传递pojo包装类型

可以定义pojo包装类型扩展mapper接口输入参数的内容。

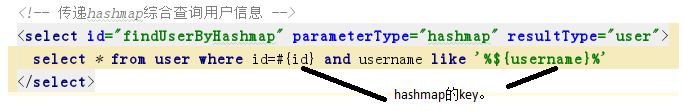


测试代码



## 传递Map或hashmap

同传递POJO对象一样，**map的key相当于pojo的属性**。



测试代码:

**Public void** testFindUserByHashmap()**throws** Exception{

//获取session

SqlSession session = sqlSessionFactory.openSession();

//获限mapper接口实例

UserMapper userMapper = session.getMapper(UserMapper.**class**);

//构造查询条件Hashmap对象

HashMap<String, Object> map = **new** HashMap<String, Object>();

map.put("id", 1);

map.put("username", "管理员");

//传递Hashmap对象查询用户列表

List<User>list = userMapper.findUserByHashmap(map);

//关闭session

session.close();

}

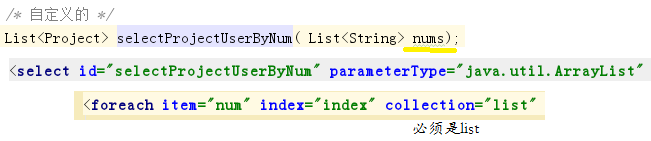
异常测试：

传递的map中的key和sql中解析的key不一致。

测试结果没有报错，只是通过key获取值为空。

## 传递list和数组

参考：https://www.cnblogs.com/ruiati/p/6410339.html

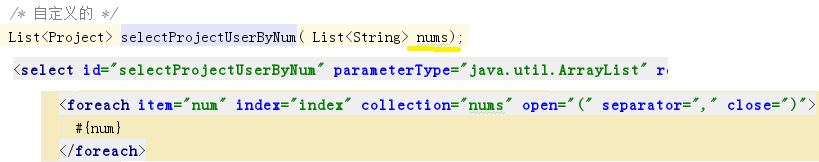


数组必须是array

# @param（****多参数传递****）

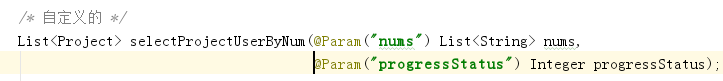
## List

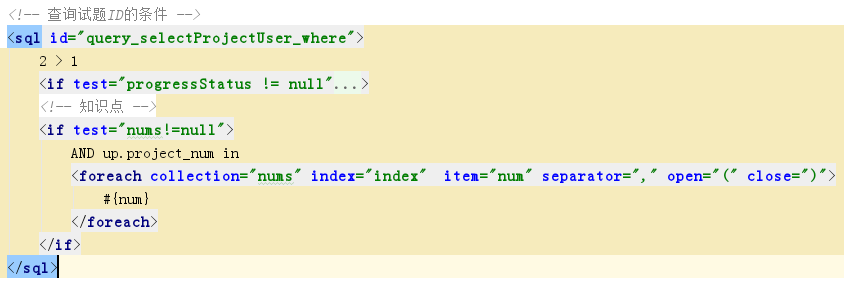
在上面如果不指定@param注解，然后数据就比较死板

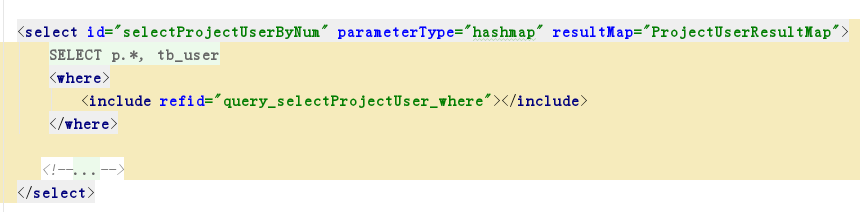


https://blog.csdn.net/u012031380/article/details/54924641

## 使用map传递复杂类型（包含list）







# 注意

## parameterType异常

如果parameterType中指定属性错误，异常，找不到getter方法：

org.apache.ibatis.exceptions.PersistenceException:

### Error querying database. Cause: org.apache.ibatis.reflection.ReflectionException: There is no getter for property named 'userCusto' in 'class cn.itcast.mybatis.po.UserQueryVo'

### Cause: org.apache.ibatis.reflection.ReflectionException: There is no getter for property named 'userCusto' in 'class cn.itcast.mybatis.po.UserQueryVo'

注意：如果将来和spring整合后，不是通过调用getter方法来获取属性值，通过反射强读取pojo的属性值。

## mybatis中#{}和${}的区别是什么？

${}是Properties文件中的变量占位符，它可以用于标签属性值和sql内部，属于静态文本替换，比如${driver}会被静态替换为com.mysql.jdbc.Driver。

#{}是sql的参数占位符，Mybatis会将sql中的#{}替换为?号，在sql执行前会使用PreparedStatement的参数设置方法，按序给sql的?号占位符设置参数值，比如ps.setInt(0, parameterValue)，#{item.name}的取值方式为使用反射从参数对象中获取item对象的name属性值，相当于param.getItem().getName()。

## 设置jdbcType

可以看出，是因为你传入的参数的字段为null对象无法获取对应的jdbcType类型，而报的错误。   
你只要在insert语句中insert的对象加上jdbcType就可以了，修改如下：   
#{menuTitle,jdbcType=VARCHAR}   
这样就可以解决以上错误了。 

## 解决实体类与表名不一样（select \*）

