

输入映射

通过parameterType指定输入参数类型，类型可以是简单类型、hashmap、pojo的包装类型。

1. 传递pojo的包装对象

1. 需求

- 完成用户信息的综合查询，需要传入查询条件（可能包括用户信息、其他信息、比如商品、订单的）。

2. 定义包装类型的pojo

- 针对以上需求，建议使用自定义的包装类型的pojo。
- 在包装类型的pojo中将复杂的查询条件包装进去。

```
package com.swxc.mybatis.po;

/**
 * Description 用户包装类型
 * Author shadowolf
 * Date 2017/7/24. 2:32
 * Version 1.0
 */
public class UserQueryVo {

    // 在这里你包装所有的查询条件

    // 用户查询条件
    // UserCustom是User类的扩展类，需要继承User类
    private UserCustom userCustom;

    public UserCustom getUserCustom() {
        return userCustom;
    }

    public void setUserCustom(UserCustom userCustom) {
        this.userCustom = userCustom;
    }

    // 可以包装其他的查询条件，比如订单、商品
    // .....
}
```

3. 映射文件mapper.xml

- 在UserMapper.xml中定义用户信息的综合查询（查询条件复杂，通常高级查询进行复杂关联查询）。

```
<!-- 用户信息的综合查询
#{userCustom.sex}: 取出pojo包装对象中性别的值。
${userCustom.username}: 取出pojo包装对象中用户名称。
-->
<select id="findUserList" parameterType="com.swxc.mybatis.po.UserQueryVo" resultType="com.swxc.mybatis.po.UserCustom">
    select * from user where user.sex = #{userCustom.sex} and user.username like '%${userCustom.username}%';
</select>
```

4. 接口文件mapper.java

```
// 用户信息综合查询
```

```
public List findUserList(UserQueryVo userQueryVo) throws Exception;
```

5. 测试代码

```
// 用户信息的综合查询
@Test
public void findUserListTest() throws Exception {
    SqlSession sqlSession = sqlSessionFactory.openSession();
    UserMapper userMapper = sqlSession.getMapper(UserMapper.class);
    UserQueryVo userQueryVo = new UserQueryVo();
    UserCustom userCustom = new UserCustom();
    userCustom.setSex("g");
    userCustom.setUsername("五");
    userQueryVo.setUserCustom(userCustom);
    List list = userMapper.findUserList(userQueryVo);
    sqlSession.close();
    System.out.println(list);
}
```

2. 传递hashmap

1. 映射文件mapper.xml

```
<!-- 传入hashmap综合查询用户信息 -->
<select id="findUserList2" parameterType="hashmap" resultType="user">
    select * from user where id = #{id} and username like '${username}$'
</select>
```