# xml取值 <select

## 1、存储过程 一个in 一个out

## 2、xml中编辑文字

|  |
| --- |
| <!--  存储过程查询  parameterMap:传入的参数是一个Map  statementType：CALLABLE调用存储过程查询  -->  <select id=*"getUserCount"* parameterMap=*"getUserCountMap"* statementType=*"CALLABLE"*resultType="java.util.HashMap  >  CALL ges\_user\_count(?,?)  </select>  <!--  输入参数对应的map  map.put("sexid", 0);  map.put("userCount", -1);//任意给定默认值  -->  <parameterMap type=*"java.util.Map"* id=*"getUserCountMap"*>  <parameter property=*"sexid"* mode=*"IN"* jdbcType=*"INTEGER"*/>  <parameter property=*"userCount"* mode=*"OUT"* jdbcType=*"INTEGER"*/>  </parameterMap> |

## 3、测试使用，这里没有用到逆向工程的方法

|  |
| --- |
| SqlSessionFactory sessionFactory = DBUtil.getSessionFactory();  SqlSession session = sessionFactory.openSession();  // 创建存储参数的Map  Map<String, Integer>map = new HashMap<String, Integer>();  map.put("sexid", 1);  map.put("userCount", -1);//任意给定默认值，  String statement ="com.bart.mybatis.mapper.procedureMapper.getUserCount";  session.selectOne(statement, map);  //获得存储过程执行完后的userCount的值  Integer count = map.get("userCount");  System.out.println("查询到的值："+count);  session.close(); |

# 2、第二种xml使用，<procedure I 得到的是对象

<http://blog.csdn.net/wangxy799/article/details/51206692>

## 1. 创建存储过程

|  |
| --- |
| create or replace procedure getEmps(p\_dno in emp.empno%type,rs\_cursor out SYS\_REFCURSOR)  AS  BEGIN  --没有给定部门工ID则返回所有员工信息  if p\_dno is null then  OPEN rs\_cursor for select \* from emp;  else  --返回指定部门下的员工信息  OPEN rs\_cursor for select \* from emp where deptno = p\_dno ;  end if;  END; |

## 2. 返回对象的实体类

public class Employee {

String name;

long employeeNumber;

long departmentNumber;

...getter setter...

}

## 3. sqlMap.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE sqlMap PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN" "http://ibatis.apache.org/dtd/sql-map-2.dtd" >  <sqlMap namespace="AUDIT\_DEPARTMENT">  <!-- ===============调用存储过程开始=============== -->  <typeAlias alias="emp" type="com.cmsz.auditvces.sell.model.Employee"/>  <resultMap class="emp" id="empMap">  <result property="name" column="ENAME" />  <result property="employeeNumber" column="EMPNO" />  <result property="departmentNumber" column="DEPTNO" />  </resultMap>  <parameterMap class="java.util.Map" id="swapParameters1">  <parameter property="p\_dno" jdbcType="INTEGER" javaType="int" mode="IN" />  <parameter property="rs\_cursor" jdbcType="ORACLECURSOR" javaType="cursor" mode="OUT" resultMap="empMap" />  </parameterMap>  <!-- 调用一个存储过程,存储过程必须指定 parameterMap,如果有返回值的话必须指定 resultMap -->  <procedure id="getEmps" parameterMap="swapParameters1">  <![CDATA[  {call getemps(?,?)}  ]]>  </procedure>  <!-- ===============调用存储过程结束=============== -->  </sqlMap> |

## 4. java调用

|  |
| --- |
| /\*\*  \* 查询部门信息  \*/  public List<Employee> getEmps(Integer dno) throws Exception {  Map<String, Object> paramMap = new HashMap<String, Object>();  paramMap.put("p\_dno", dno);  // getSqlMapClientTemplate().queryForList("AUDIT\_DEPARTMENT.getEmps", paramMap);  //这两个都可以用  getSqlMapClientTemplate().queryForObject("AUDIT\_DEPARTMENT.getEmps", paramMap);  List<Employee> list = (List<Employee>) paramMap.get("rs\_cursor");  return list;  } |

## 注意事项

1. 存储过程的返回项不宜用Boolean类型，用起来麻烦。
2. Oracle中的NUMBER类型 在sqlMap中配置时一定要注意，不能写成 jdbcType=”NUMBER”。没有这个类型

# 3、接口逆向工程

<http://www.cnblogs.com/shihaiming/p/6248439.html>

## 1、dao方法

Map<String, Object> testProcedure(Map<String, Object> paraMap);

## 2、ProcedureMapper.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >  <mapper namespace="cn.zsmy.mapper.ProcedureMapper" >  <select id="testProcedure" parameterType="java.util.Map" statementType="CALLABLE" resultType="java.util.HashMap">  {call loginandreg(  #{out\_ret,mode=OUT,javaType=java.lang.Integer,jdbcType=INTEGER},  #{out\_desc,mode=OUT,javaType=java.lang.String,jdbcType=VARCHAR},  #{userId,jdbcType=VARCHAR,mode=OUT},  #{user\_pwd,jdbcType=VARCHAR,mode=IN},  #{nickname,jdbcType=VARCHAR,mode=IN}  )  }  </select>    </mapper> |

## 3、测试使用

|  |
| --- |
| @RequestMapping(value = "/testProcedure", method = RequestMethod.GET)  public @ResponseBody String testProcedure(HttpServletRequest request, HttpServletResponse response) throws IOException {  Map<String, Object> paraMap = new HashMap<String, Object>();  //输出参数可以不传  //paraMap.put("out\_ret", -1);  //paraMap.put("out\_desc", "");  //paraMap.put("userId", null);  paraMap.put("user\_pwd", "123456");  paraMap.put("nickname", UUID.randomUUID().toString().replace("-", "").substring(0, 10));  procedureMapper.testProcedure(paraMap);  System.out.println(paraMap);  return null;  } |