  Ibatis是我们经常使用的O/R映射框架，mybats是ibatis被Google收购后重新命名的一个工程，当然也做了大量的升级。而调用存储过程也是一次额C/S架构模式下经常使用的手段，我们知道，ibatis调用存储过程有一个专门的标签<procedure>，在mybats里面已经没有这本标签了，而是通过一个参数statementType=*"CALLABLE"*来区分。废话不多说，直接看怎么写吧！

测试环境：mybats3.0.4 + sqlserver2008 +Spring3.1.1

存储过程：

|  |
| --- |
| use base\_passport  go    create proc yhj\_test      @input\_a int,      @input\_b int,      @output\_c int out  as  begin      select \* from config.city;      set @output\_c = @input\_a\*@input\_b;      return @input\_a+@input\_b;  end |

Mybats配置：

|  |
| --- |
| <resultMap type=*"java.util.HashMap"* id=*"resultMap"*>         <result column=*"city\_key"* property=*"cityKey"*javaType=*"java.lang.Integer"* jdbcType=*"INTEGER"*/>         <result column=*"province\_key"* property=*"provinceKey"*javaType=*"java.lang.Integer"* jdbcType=*"INTEGER"*/>         <result column=*"city\_name"* property=*"cityName"*javaType=*"java.lang.String"* jdbcType=*"VARCHAR"*/>      </resultMap>        <select id=*"testByProc"* statementType=*"CALLABLE"*parameterType=*"java.util.HashMap"* resultMap=*"resultMap"*>               {#{d,mode=OUT,jdbcType=INTEGER}=call dbo.yhj\_test(                #{a,mode=IN,jdbcType=INTEGER},                #{b,mode=IN,jdbcType=INTEGER},                #{c,mode=OUT,jdbcType=INTEGER}             )}        </select> |

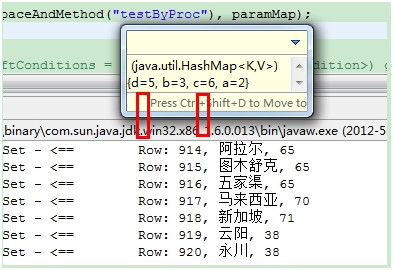
Java代码

|  |
| --- |
| Map<String, Object> paramMap = **new** HashMap<String, Object>();         paramMap.put("a", 2);         paramMap.put("b", 3);         List<City> citys = (List<City>) getTemplate().selectList(getNameSpaceAndMethod("testByProc"), paramMap);  **return** citys; |

通过以上代码我们即可获取对应的结果集，return参数和output参数。其中return参数和output参数放在传入的paramMap中，运行结果如下：



 如图所示，数据库中的返回结果集被查出。



 如图所示，其中

c是存储过程的output参数，而d为return参数。

需要注意的事项：

1、  存储过程的参数和名称无关，只和顺序有关系

2、  存储过程的output参数，只能通过传入的map获取

3、  存储过程返回的结果集可直接用返回的map接收

4、  存储过程的return结果需要使用?=call procName(?,?)的第一个参数接收，需要指定对应的mode为OUT类型

5、  存储过程对应的数据类型为枚举类型，需要使用大写，如*VARCHAR*

## 其他-调用存储过程：

### 1）无返回结果

<update id=*"callRtDetailProcedure"* parameterMap=*"callRtDetailProcedureMap"* statementType=*"CALLABLE"*>

CALL P\_RT\_DETAIL(?,?,?,?)

</update>

<parameterMap type=*"map"* id=*"callRtDetailProcedureMap"*>

<parameter property=*"V\_USER\_ID"* mode=*"IN"* javaType=*"LONG"*/>

<parameter property=*"V\_TASK\_ID"* mode=*"IN"* javaType=*"LONG"*/>

<parameter property=*"V\_C\_OPEN\_ID"* mode=*"IN"* javaType=*"STRING"*/>

<parameter property=*"V\_ITEM\_ID"* mode=*"IN"* javaType=*"LONG"*/>

</parameterMap>

2）

## Mybatis调用存储过程返回多个结果集

<resultMap type="Integer" id="count"> <!—无property属性-->

<result column="RecordCount" jdbcType="INTEGER" javaType="Integer" />

</resultMap>

<resultMap type="OrderForm" id="orders">

<result column="OrderId" property="id" jdbcType="VARCHAR" javaType="String"/>

</resultMap>

<!-- resultMap type="map" 需要property属性与之配对。property为key，查询结果为value -->

<resultMap type=*"map"* id=*"consumeRtnMap"*>

<result column=*"RTN\_RESULT"* property=*"rtnResult"* javaType=*"Integer"* jdbcType=*"INTEGER"*/>

</resultMap>

<select id="getOrders" statementType="CALLABLE" parameterType="Map" resultMap="count,orders,*consumeRtnMap* " >

CALL P\_U\_JIFEN\_CONSUME(

#{userid,mode=IN,jdbcType=BIGINT},

#{consumeVal,mode=IN,jdbcType=DECIMAL},

#{uType,mode=IN,jdbcType=VARCHAR},

#{orderNum,mode=IN,jdbcType=VARCHAR}

)

</select>

需要注意的地方 statementType="CALLABLE" 不能少resultMap="count,orders" 这里返回多个结果集，如果有更多可以继续加

Dao层的接口

public List<List<?>> getOrders(Map<String, Object> map); 🡪因为结果包含多个结果集，每个结果集type值不一样，故使用问号（?）

Service层调用

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

map.put("id", "22333");

//取得返回的结果集

List<List<?>> results = orderDao.getOrders(map);

//第一条结果集 总数量

System.out.println(((List<Integer>)results.get(0)).get(0));

//第二条订单列表

System.out.println((List<OrderForm>)results.get(1));

====================================mime====================================

prams.put("userid", rtnUserInfoMap.get("userid"));

params.put("consumeVal", tmpConsumeVal);

params.put("uType", uType);

params.put("orderNum", orderNum);

List<List<?>> consumeList = jiFenService.consumeJifen(params);

if(null!=consumeList){

HashMap rtnResultMap =(HashMap) consumeList.get(0).get(0);

HashMap useableTradeMap =(HashMap) consumeList.get(1).get(0);

System.out.println("返回结果：："+rtnResultMap.get("rtnResult"));

System.out.println("剩余可用积分::"+useableTradeMap.get("useableTrade"));