# Java 和oracle 的交互

资料来源： <https://blog.csdn.net/zl18310999566/article/details/52458858>

这种问题在xml处理sql的程序中经常需要我们来进行特殊处理。

     其实很简单，我们只需作如下替换即可避免上述的错误：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| < | <= | > | >= | & | ' | " |
| &lt; | &lt;= | &gt; | &gt;= | &amp; | &apos; | &quot; |

## Spring 配置

Spring 配置文件

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:aop=*"http://www.springframework.org/schema/aop"* xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.2.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.2.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx-3.2.xsd*

*http://www.springframework.org/schema/aop*

*http://www.springframework.org/schema/aop/spring-aop-3.2.xsd"*>

<bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"* init-method=*"init"* destroy-method=*"clone"*>

<!-- 基本属性driverClassName、 url、user、password -->

<property name=*"driverClassName"* value=*"${web.gateway.jdbc.driver}"* />

<property name=*"url"* value=*"${web.gateway.jdbc.url}"* />

<property name=*"username"* value=*"${web.gateway.jdbc.username}"* />

<property name=*"password"* value=*"${web.gateway.jdbc.password}"* />

<!-- 配置初始化大小、最小、最大 -->

<!-- 通常来说，只需要修改initialSize、minIdle、maxActive -->

<!-- 初始化时建立物理连接的个数，缺省值为0 -->

<property name=*"initialSize"* value=*"${web.gateway.jdbc.initialSize}"* />

<!-- 最小连接池数量 -->

<property name=*"minIdle"* value=*"${web.gateway.jdbc.minIdle}"* />

<!-- 最大连接池数量 -->

<property name=*"maxActive"* value=*"${web.gateway.jdbc.maxActive}"* />

<property name=*"removeAbandoned"* value=*"true"* />

<!-- 超时时间；单位为秒。180秒=3分钟 -->

<property name=*"removeAbandonedTimeout"* value=*"180"* />

<!-- 关闭abanded连接时输出错误日志 -->

<property name=*"logAbandoned"* value=*"true"* />

</bean>

</beans>

Properties 配置文件

##JDBC Setting

##common

web.gateway.jdbc.initialSize=50

web.gateway.jdbc.minIdle=50

web.gateway.jdbc.maxActive=300

web.gateway.jdbc.driver=oracle.jdbc.driver.OracleDriver

##aliyun nuannuan 共享纸巾

#web.gateway.jdbc.url=jdbc:oracle:thin:@101.132.172.78:1521:nnorcl

#web.gateway.jdbc.username=gx\_trd

#web.gateway.jdbc.password=gxzj716

#aliyun 口红

#web.gateway.jdbc.url=jdbc:oracle:thin:@47.99.216.231:1521:jhorcl

#web.gateway.jdbc.username=MAC\_TRD

#web.gateway.jdbc.password=lpmac116

# 嗨呗 数据库

web.gateway.jdbc.url=jdbc:oracle:thin:@101.132.172.78:1521:nnorcl

web.gateway.jdbc.username=hb\_trd

web.gateway.jdbc.password=123456

## 配置dao 接口

### 调用存储过程

接口的配置

package com.junhui\_hz.dao.system;

import java.util.Map;

public interface SystemDao {

/\*\*

\* 获取系统的信息

\* @param id

\*/

void getSysparameter(Map<String, Object> params);

/\*\*

\* app 进行系统更新

\* @param id

\*/

void appUpdate(Map<String, Object> params);

}

XML 文件的配置

<?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="com.junhui\_hz.dao.system.SystemDao">

<resultMap type ="java.util.Map" id= "userImga">

<result column ="ERRORINFO" property="errorinfo" jdbcType="VARCHAR"/>

<result column ="ERRORCODE" property="errorcode" jdbcType="VARCHAR"/>

</resultMap >

<select id="getSysparameter" statementType="CALLABLE"

parameterType="java.util.Map">

<![CDATA[

{#{systemInfo, jdbcType=VARCHAR,mode=OUT}=call cl\_get\_sysparameter

(#{parametertype, jdbcType=INTEGER ,mode=IN},

#{parameterkey, jdbcType=INTEGER ,mode=IN})}

]]>

</select>

<select id="appUpdate" statementType="CALLABLE"

parameterType="java.util.Map">

<![CDATA[

{call CL\_10019(

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

#{messageList,mode=OUT,jdbcType=CURSOR,javaType=java.sql.ResultSet,resultMap=userImga}

)}

]]>

</select>

<select id="getPublicPicUrl" statementType="CALLABLE"

parameterType="java.util.Map">

<![CDATA[

{call CL\_10020(

#{messageList,mode=OUT,jdbcType=CURSOR,javaType=java.sql.ResultSet,resultMap=userImga}

)}

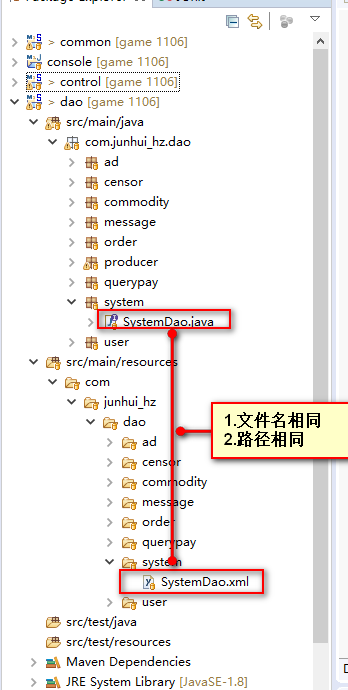
]]>

</select>

</mapper>

注意：

图片说明



### 调用函数

都是使用的mybatis ，调用 函数的 和调用 存储过程的方法相似。

大部分的要求都是一样的。只有XML 文件的配置不同

**package** com.junhui\_hz.dao.order;

**import** java.util.Map;

**public** **interface** OrderDao {

/\*\*

\* 生成订单号

\* **@param** paramMap

\*/

**void** getOrderNum(Map<String, Object> paramMap);

}

下面是XML 配置文件

<?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=*"com.junhui\_hz.dao.order.OrderDao"*>

<!-- 生成短信验证码 -->

<select id=*"getOrderNum"* statementType=*"CALLABLE"*>

{#{orderNum, jdbcType=VARCHAR,mode=OUT}=call cl\_generativerule

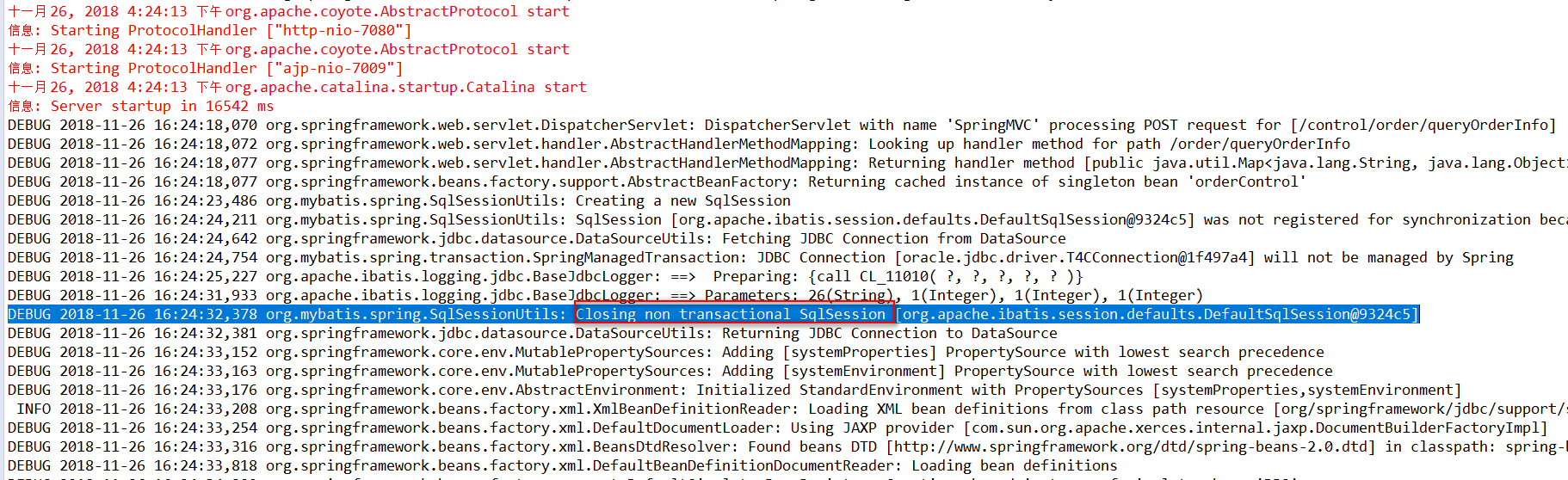
(#{code, jdbcType=INTEGER ,mode=IN})}

</select>

</mapper>

错误

DEBUG 2018-11-26 16:24:32,378 org.mybatis.spring.SqlSessionUtils: Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@9324c5]

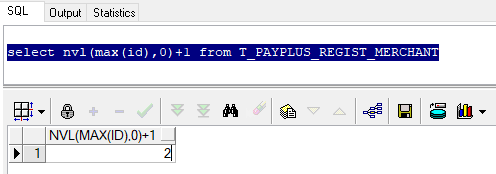


出现这种现象的愿意是，没有返回一个游标！！

### 主键自增长

资料来源： <https://blog.csdn.net/wal1314520/article/details/77132305>

通过sql 进行查询 ID



**使用selectKey标签**

<insert id="addLoginLog" parameterType="map" >

<selectKey keyProperty="id" resultType="int" order="BEFORE">

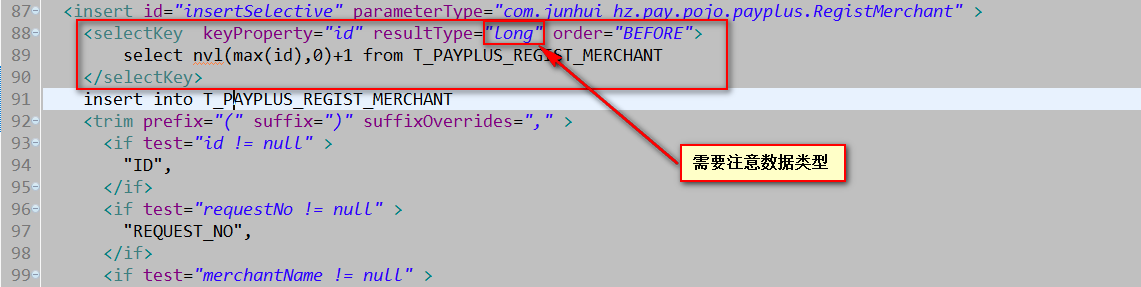
select nvl(max(id),0)+1 from ap\_loginlog

</selectKey>

insert into ap\_loginlog(ID,MEMBER\_ID) values(#{id},#{memberId})

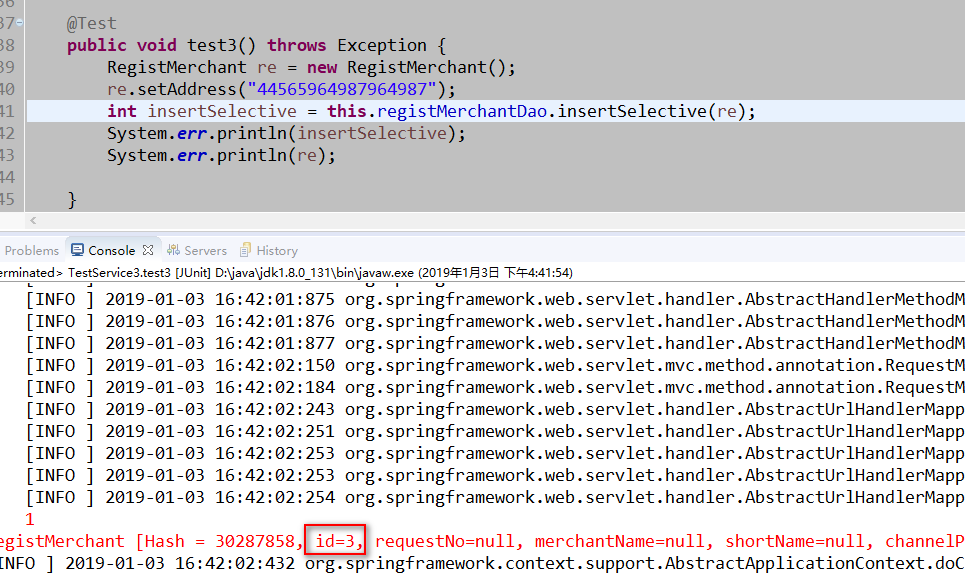
</insert>

注意： 数据类型



只需要在插入语句前面 加上查询最大的 ID 数值即可

可以直接获取id



只查询一条语句

<select id="selectByExample" resultMap="BaseResultMap" parameterType="com.junhui\_hz.pay.pojo.bank.BankAuth" >

<include refid="com.junhui\_hz.pay.dao.common.CommonDao.limitHead" />

SELECT

ID,

NAME,

BANKCARD,

IDNUMBER,

MOBILE,

ISDISABLE,

DATE\_FROM,

CREATE\_TIME,

TYPE,

RESULT

FROM

T\_BANK\_AUTH

<where >

1 = 1

<if test="id != null" >

AND "ID" = #{id,jdbcType=DECIMAL}

</if>

<if test="name != null" >

AND "NAME" = #{name,jdbcType=VARCHAR}

</if>

<if test="bankcard != null" >

AND "BANKCARD" = #{bankcard,jdbcType=VARCHAR}

</if>

<if test="idnumber != null" >

AND "IDNUMBER" = #{idnumber,jdbcType=VARCHAR}

</if>

<if test="mobile != null" >

AND "MOBILE" = #{mobile,jdbcType=VARCHAR}

</if>

<if test="isdisable != null" >

AND "ISDISABLE" = #{isdisable,jdbcType=DECIMAL}

</if>

<if test="dateFrom != null" >

AND "DATE\_FROM" = #{dateFrom,jdbcType=VARCHAR}

</if>

<if test="createTime != null" >

AND "CREATE\_TIME" = #{createTime,jdbcType=TIMESTAMP}

</if>

<if test="type != null" >

AND "TYPE" = #{type,jdbcType=DECIMAL}

</if>

<if test="result != null" >

AND "RESULT" = #{result,jdbcType=VARCHAR}

</if>

</where>

<include refid="com.junhui\_hz.pay.dao.common.CommonDao.limitTail" />

</select>

### 使用序列

使用序列进行主键自增长

**SQL Snippet in XML Configuration:**

<insert id="insertSelective" parameterType="com.jxxx.p2pp.model.UUserInfo">

<selectKey resultType="java.math.BigDecimal" order="BEFORE" keyProperty="id">

SELECT U\_USER\_INFO\_SEQ.Nextval as ID from DUAL

</selectKey>

insert into U\_USER\_INFO

<trim prefix="(" suffix=")" suffixOverrides="," >

<if test="id != null" >

ID,

</if>

<if test="userName != null" >

USER\_NAME,

</if>

<if test="realName != null" >

REAL\_NAME,

</if>

.....

</insert>

创建序列 根据实际情况进行修改

CREATE SEQUENCE CL\_Cash\_Pay\_ID

INCREMENT BY 1

START WITH 1

NOMAXvalue

NOCYCLE

NOCACHE;

### Sql 片段

只要在前后 加上

<include refid="com.junhui\_hz.pay.dao.common.CommonDao.limitHead" />

<include refid="com.junhui\_hz.pay.dao.common.CommonDao.limitTail" />

引用的内容

<?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" >

<!-- 公共的mapper 文件 -->

<mapper namespace=*"com.junhui\_hz.pay.dao.common.CommonDao"* >

<sql id=*"limitHead"*>

SELECT

T.\*, ROWNUM

FROM

(

</sql>

<sql id=*"limitTail"*>

) T

WHERE

ROWNUM &lt;= 1

</sql>

</mapper>

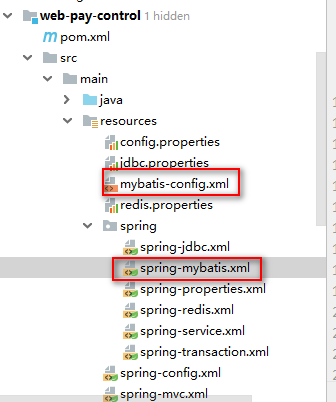
# 分页的使用

## 配置文件

Mybatis 中配置

Spring-mybatis.xml 配置

*<?*xml version="1.0" encoding="UTF-8"*?>*<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:aop="http://www.springframework.org/schema/aop" xmlns:tx="http://www.springframework.org/schema/tx"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans-3.2.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context-3.2.xsd  
 http://www.springframework.org/schema/tx   
 http://www.springframework.org/schema/tx/spring-tx-3.2.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop-3.2.xsd">  
  
 *<!-- 管理回话工厂 -->* <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  
 *<!-- 注入数据源 -->* <property name="dataSource" ref="dataSource"/>  
 *<!-- 加载mybatis的核心配合文件 -->* <property name="configLocation" value="classpath:mybatis-config.xml"/>  
 </bean>  
   
 *<!-- 管理mapper的代理对象 -->* <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  
 <property name="basePackage" value="com.junhui\_hz.pay.dao"/>  
 </bean>  
  
</beans>



Mybatis-config.xml 配置

*<?***xml version="1.0" encoding="UTF-8"** *?>***<!DOCTYPE configuration  
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-config.dtd"*>***<**configuration**>  
 *<!-- 别名 -->* <**typeAliases**>  
 <**package name="com.junhui\_hz.pay"**/>  
 </**typeAliases**>  
   
 <**plugins**>  
 <**plugin interceptor="com.github.pagehelper.PageHelper"**>  
 *<!--数据库方言-->* <**property name="dialect" value="oracle"**/>  
 *<!--设置成true ,使用rowbounds分页进行count会出现总条数的-->* <**property name="rowBoundsWithCount" value="true"**/>  
 </**plugin**>  
 </**plugins**>  
</**configuration**>

## PageHelper 的使用

@ Autowired  
**private** SmsAliyunDao **smsAliyunDao**;  
  
@Test  
**public void** name() {  
 PageHelper.*startPage*(1, 2);*//重点就这一行代码 必须在查询之前* List<SystemChannelName> systemChannelNames = **this**.**channelDao**.selectbyPage();  
  
  
 PageInfo<SystemChannelName> pageInfo = **new** PageInfo<SystemChannelName>(systemChannelNames);  
 System.***err***.println(pageInfo.getTotal());  
 System.***err***.println(pageInfo);  
 **for** (SystemChannelName name : systemChannelNames) {  
 System.***err***.println(name);  
 }  
}

Mybatis 配置 查询

<**select id="selectbyPage" resultMap="BaseResultMap"** >  
 SELECT  
 ID,  
 CHANNEL\_NAME,  
 CHANNEL\_SIGN,  
 IS\_DISABLE  
 FROM  
 T\_SYSTEM\_CHANNEL\_NAME  
</**select**>

# 日期格式的使用

查询日期

SELECT \*

FROM T\_SMS\_ALIYUN

WHERE 1 = 1

AND CREATE\_TIME > TO\_DATE('2019-04-03,13:25:59', 'yyyy-mm-dd,hh24:mi:ss')

Mybatis 的代码

<**select id="selectbyPage" resultMap="BaseResultMap"** >  
 SELECT <**include refid="Base\_Column\_List"**/>  
 FROM T\_SMS\_ALIYUN  
 <**where**>  
 1 = 1  
 <**if test="start != null"** >  
 <![CDATA[  
 AND CREATE\_TIME >= to\_date(#{start ,jdbcType=VARCHAR}, 'yyyy-mm-dd')  
 ]]>  
 </**if**>  
 <**if test="end != null"** >  
 <![CDATA[  
 AND CREATE\_TIME <= to\_date(#{end,jdbcType=VARCHAR}, 'yyyy-mm-dd')  
 ]]>  
 </**if**>  
 <**if test="phone != null"** >  
 AND PHONE like #{phone,jdbcType=VARCHAR}  
 </**if**>  
 <**if test="sign != null"** >  
 AND SIGN = #{sign,jdbcType=VARCHAR}  
 </**if**>  
 </**where**>  
 ORDER BY ID DESC  
 </**select**>  
</**mapper**>

运行结果

