【Java框架型项目从入门到装逼】第十五节 - jdbc模糊查询实现（附带详细调试过程）

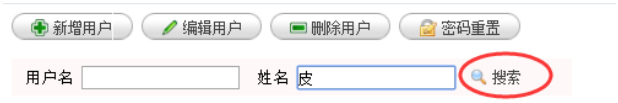
上一节，我们实现了用户列表查询，已经按条件精确查询：

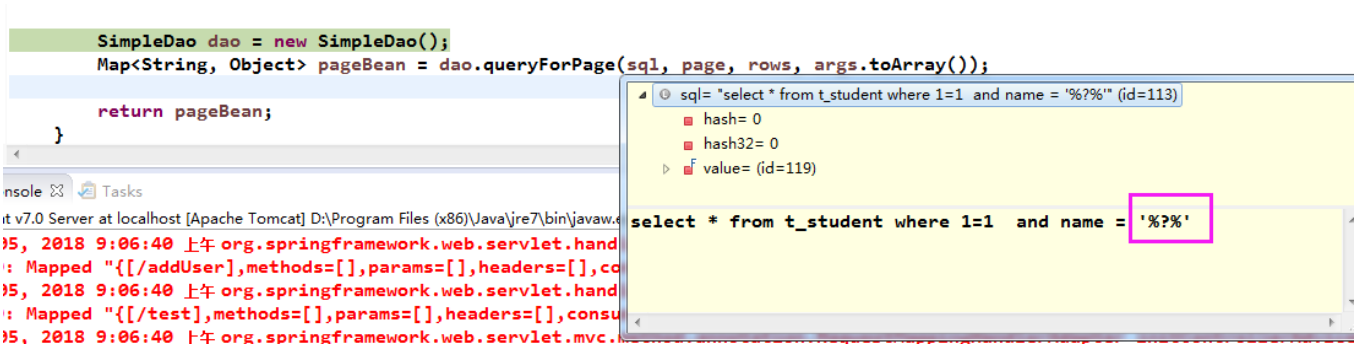
```
if(student.getUsername() != null && !"".equals(student.getUsername())){
    sql += " and username = ?";
    args.add(student.getUsername());
}

if(student.getName() != null && !"".equals(student.getName())){
    sql += " and name = ?";
    args.add(student.getName());
}
```

因为是精确查询，所以我们使用了等号，如果是模糊查询咋办呢?在sql语句里需要使用like关键字，第一次修改：

```
if(student.getUsername() != null && !"".equals(student.getUsername())){
    sql += " and username like '%?%'";
    args.add(student.getUsername());
}

if(student.getName() != null && !"".equals(student.getName())){
    sql += " and name like '%?%'";
    args.add(student.getName());
}
```

从代码上看，貌似没有啥问题。ok，来测试一下。





结果报错了：

```
java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0).
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:1055)
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:956)
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:926)
        at com.mysql.jdbc.PreparedStatement.checkBounds(PreparedStatement.java:3288)
        at com.mysql.jdbc.PreparedStatement.setInternal(PreparedStatement.java:3272)
        at com.mysql.jdbc.PreparedStatement.setString(PreparedStatement.java:4108)
        at com.mysql.jdbc.PreparedStatement.setObject(PreparedStatement.java:3527)
        at org.apache.commons.dbcp.DelegatingPreparedStatement.setObject(DelegatingPreparedStatement.java:169)
        at org.apache.commons.dbcp.DelegatingPreparedStatement.setObject(DelegatingPreparedStatement.java:169)
        at com.simple.dao.SimpleDao.queryForList(SimpleDao.java:108)
        at com.simple.dao.SimpleDao.queryForMap(SimpleDao.java:139)
        at com.simple.dao.SimpleDao.queryForJsonObject(SimpleDao.java:77)
```

出现问题不要怕，用调试工具来一步一步调，走进源码里进行调试：

```
public Map<String, Object> queryForPage(String sql, int pageIndex, int p
    HashMap pageMap = new HashMap();

    int startIndex = (pageIndex - 1) * pageSize;
    long total = this.getTotal(sql, objects);    获取总条数

    pageMap.put("total", Long.valueOf(total));

    sql = sql + " limit ?,?";
```

进入这个方法：

```
public long getTotal(String sql, Object... objects) {
    sql = "select count(1) from (" + sql + ") t";
    long total = this.queryForLong(sql, objects);
    return total;
}
```

走到这个queryForLong方法，再进去：

```
public long queryForLong(String sql, Object... objects) {
    return Long.parseLong(this.queryForString(sql, objects));
}
```

发现又调用了queryForString方法，ok，继续走进去：

```
public String queryForString(String sql, Object... objects) {
    JSONObject jsonObject = this.queryForJsonObject(sql, objects);
    Iterator sIterator = jsonObject.keys();
    if(sIterator.hasNext()) {

        String key = (String)sIterator.next();

        String value = jsonObject.getString(key);

        return value;
    } else {
        return null;
    }
}
```

sql:

select count(1) from (select * from t_student where 1=1 and name = '%?%') t

objects:

[皮]

再去看一下queryForJsonObject方法：

```
public JSONObject queryForJsonObject(String sql, Object... objects) {
    Map map = this.queryForMap(sql, objects);
    return map == null?null:JSONObject.fromObject(map);
}
```

我的天，又调用了queryForMap方法。。

```
public Map<String, Object> queryForMap(String sql, Object... objects) {
    new HashMap();
    List list = this.queryForList(sql, objects);
    if(list.size() != 1) {
        return null;
    } else {
        Map result = (Map)list.get(0);
        return result;
    }
}
```

queryForMap中，最终还是调用了queryForList方法，至于第一句话，是反编译出来的。这一点可以看出我之前写的源码是存在问题的，创建了一个Map对象但是最终没有用到。好吧，不用在意这些细节。我估计当初写源码的时候，那个Map本来是想要去返回的，可实际上 list.get(0)返回的就已经是一个Map对象了，所以我实际上不需要去new一个HashMap。

我们再走入queryForList方法，发现在这一行报错了：

```
try {
    statement = this.connection.prepareStatement(sql);
    for(int e = 0; e < objects.length; ++e) {
        statement.setObject(e + 1, objects[e]);
    }
```

```
java.sql.SQLException: Parameter index out of range (1 > number of parameters, which is 0).
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:1055)
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:956)
        at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:926)
        at com.mysql.jdbc.PreparedStatement.checkBounds(PreparedStatement.java:3288)
        at com.mysql.jdbc.PreparedStatement.setInternal(PreparedStatement.java:3272)
        at com.mysql.jdbc.PreparedStatement.setString(PreparedStatement.java:4108)
        at com.mysql.jdbc.PreparedStatement.setObject(PreparedStatement.java:3527)
```

终于找到问题了，就是在这个setObject的过程中，出了问题。这是原生的jdbc方法。

其实，这是jdbc内部的一个问题，再回过来看一下sql语句：

select count(1) from (select * from t_student where 1=1 and name = '%?%') t

？是不能放在单引号里面的，如果放在单引号里面，PreparedStatement并不视它为一个参数，错就错在这。

看到这里，有的人就要问了，那我如何才可以实现模糊查询呢？既然jdbc不允许我们把？写在单引号里面，那么我们干脆就写一个问号，没有单引号不就行了？

修改sql为：

select count(1) from (select * from t_student where 1=1 and name = ?) t

然后把%拼接进去：

```
if(student.getUsername() != null && !"".equals(student.getUsername())){
    sql += " and username like ?";
    args.add("%" + student.getUsername() + "%");
}

if(student.getName() != null && !"".equals(student.getName())){
    sql += " and name like ?";
    args.add("%" + student.getName() + "%");
}
```

结果：

这样就实现了一个模糊查询。相信很多jdbc初学者，在进行模糊查询的时候，都曾经踩过这个坑，再强调一遍：**？是不能放在单引号里面的，如果放在单引号里面，PreparedStatement并不视它为一个参数。**

我要下载源码

您的支持是我写作的最大动力：