



链滴

Spring 之 JdbcTemplate 进行查询的三种回调方式的比较

作者: [melon](#)

原文链接: <https://ld246.com/article/1456560382192>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

我们做过的很多项目中都会使用spring的JdbcTemplate进行结果集的查询操作，以前使用的时候一直都是“拿来主义”，功能实现了就OK了，没有深究什么内容，特别是查询接口的回调内容方法，没有过多的研究过细节内容。最近一次使用JdbcTemplate进行查询操作，发现了一些有规律的内容，所以就深入学习了一下。和大家一起探讨一下：对于spring的JdbcTemplate进行结果集查询操作，spring给我们开发的是一系列的query方法，这些查询的方法中回调的接口主要有三种：ResultSetExtractor，RowCallbackHandler，RowMapper。

但是这三种回调接口具体的用法和区别，我们一起来看一下：

场景设定，数据库中存在T_USER表，表中存在多条数据，需要将表中数据映射到User对象上

 1、 org.springframework.jdbc.core.ResultSetExtractor

 ResultSetExtractor接口中定义的方法下：

```
public interface ResultSetExtractor {  
    Object extractData(ResultSet rs) throws SQLException,  
        DataAccessException;  
}
```


如果使用ResultSetExtractor接口作为回调方法，查询方式如下：


```
List<User> userList = (List<User>)jdbcTemplate.  
    query("select * from T_USER", new ResultSetExtractor() {  
        @Override  
        public Object extractData(ResultSet rs) throws SQLException,  
            DataAccessException {  
            List<User> userList = new ArrayList<User>();  
            while (rs.next()) {  
                User user = new User();  
                user.setId(rs.getInt("N_ID"));  
                user.setName(rs.getString("C_NAME"));  
                userList.add(user);  
            }  
            return userList;  
        }  
    });
```


 2、 org.springframework.jdbc.core.RowCallbackHandler

 RowCallbackHandler接口中定义的方法如下：

```
public interface RowCallbackHandler {  
    void processRow(ResultSet rs) throws SQLException;  
}
```


[illegible]

1、RowMapper应该就是一个精简版的ResultSetExtractor，RowMapper能够直接处理一条结果集内容，而ResultSetExtractor需要我们自己去ResultSet中去取结果集的内容，但是ResultSetExtractor拥有更多的控制权，在使用上可以更灵活；

2、与RowCallbackHandler相比，ResultSetExtractor是无状态的，他不能够用来处理有状态的源。

备注：在spring3中，这三个接口的内容也做出了修改，其中：ResultSetExtractor和RowMapper开始支持泛型，返回值的内容也是泛型中的对象，而RowCallbackHandler没做出任何修改。