ResultSetExtractor 与 Rowmapper 有什么区别?

问 9 年零 6 个月前　活跃 2 年零 4 个月前　浏览了 65k 次

49

我在行映射器和结果集提取器回调接口上工作。我发现了不同之处，即，

1.行映射器可以按行处理。但是结果集提取器我们可以导航所有行并且返回类型是对象。

除了上述之外还有什么区别吗？ Rowmapper 内部和返回类型是如何工作的列表？。

20

春天　jdbc模板

分享　改进这个问题　跟随

12 年 4 月 9 日 22:04编辑

托马斯·努尔凯维奇
317k　66　673　654

问2012 年 4 月 9 日 13:35

用户1127214
2,789　7　23　29

1　在我看来，RowMapper 简单得多，您可能最终会使用它而不是 ResultSetExtractor。但有关更多信息，请查看JavaDoc和文档 —— 扎克 2012 年 4 月 9 日，13:45

4 个回答

积极的 | 最老的 | 投票

46

基本区别在于，使用 ResultsetExtractor 时，您需要自己遍历结果集，例如在 while 循环中。此接口为您提供一次对整个 ResultSet 的处理。接口方法 extractData(ResultSet rs) 的实现将包含该手动迭代代码。 查看 ResultsetExtractor 的一种实现

虽然有些回调处理程序如 RowCallbackHandler，但接口方法 processRow(ResultSet rs) 会为您循环。

RowMapper 既可以用于映射每一行，也可以用于映射整行。

对于整行对象（通过模板方法 jdbcTemplate.query()）

```
 public List findAll() {
    String sql = "SELECT * FROM EMPLOYEE";
    return jdbcTemplate.query(sql, new EmployeeRowMapper());
}
without casting will work
```

对于单个对象（使用模板方法 jdbcTemplate.queryForObject()）

```
@SuppressWarnings({ "unchecked", "rawtypes" })
public Employee findById(int id) {
    String sql = "SELECT * FROM EMPLOYEE WHERE ID = ?";
//  jdbcTemplate = new JdbcTemplate(dataSource);
```

```
    // Method 2 very easy
    //  Employee employee = (Employee) jdbcTemplate.queryForObject(sql, new Object[] {
id }, new BeanPropertyRowMapper(Employee.class));

    return employee;
}



@SuppressWarnings("rawtypes")
public class EmployeeRowMapper implements RowMapper {

public Object mapRow(ResultSet rs, int rowNum) throws SQLException {
    Employee employee = new Employee();
    employee.setId(rs.getInt("ID"));
    employee.setName(rs.getString("NAME"));
    employee.setAge(rs.getInt("AGE"));
    return employee;
}


}
```

最佳用例：

Row Mapper：当一个ResultSet的每一行都映射到一个域对象时，可以实现为私有内部类。

RowCallbackHandler：当每一行的回调方法没有返回值时，例如将行写入文件、将行转换为 XML、在添加到集合之前过滤行。非常高效，因为 ResultSet 到 Object 的映射没有在这里完成。

ResultSetExtractor：当多行 ResultSet 映射到单个 Object 时。就像在查询中进行复杂连接时，可能需要访问整个 ResultSet 而不是单行 rs 来构建复杂的对象，并且您希望完全控制 ResultSet。就像将从 TABLE1 和 TABLE2 的连接返回的行映射到完全重构的 TABLE 聚合一样。

ParameterizedRowMapper用于创建复杂对象

分享 改进这个答案 跟随 15 年 1 月 8 日 20:57编辑 15 年 1 月 8 日 17:41 回答

维马尔克里希纳
2,512 25 21

1 很好地解释了。 —— 凤凰卫视 2018 年 1 月 18 日 12:44

33

的JavaDoc `ResultSetExtractor`：

> 该接口主要用于 JDBC 框架本身。RowMapper 通常是 ResultSet 处理的一种更简单的选择，它为每一行映射一个结果对象，而不是为整个 ResultSet 映射一个结果对象。

`ResultSetExtractor` 假设提取整个 `ResultSet` （可能是多行），而 `RowMapper` 一次输入一行。

大多数情况下 [illegible] 将循环 [illegible] 使用 [illegible] Spring 的片段示例

```
List<T> results = (this.rowsExpected > 0 ? new ArrayList<T>(this.rowsExpected) : new
ArrayList<T>());
int rowNum = 0;
while (rs.next()) {
    results.add(this.rowMapper.mapRow(rs, rowNum++));
}
return results;
```

注意，所有结果都将被转换，这可能会产生 Out Of Memory 异常。

也可以看看

- RowMapperResultSetExtractor

分享 改进这个答案 跟随

19 年 6 月 12 日 8:58编辑

托马斯·德科
19k 2 86 100

2012 年4 月 9 日 13:46回答

托马斯·努尔凯维奇
317k 66 673 654

9 ResultSetExtractor 具有框架之外的用途，例如，它可用于使用单个 SQL 语句将一对多映射，例如具有一组假期的员工。 —— 碧玉蓝调 2015 年 3 月 31 日 9:53

但是 RowMapper 和 ResultSetExtractor 可用于获取记录列表？？ —— 苏珊097 2018 年 5 月 30 日 12:53

3

`RowMapper`：一次处理一条ResultSet记录。

`ResultSetExtractor`：一次处理多条ResultSet记录。

分享 改进这个答案 跟随

18 年 2 月 7 日 9:52编辑

阿尔卡姆
3,082 4 24 59

2018 年 2 月 7 日 5:52 回答

普拉卡什·库马尔·贝赫拉
31 1

应该添加为注释。 —— 阿尔卡姆 2018 年 2 月 7 日 5:58

0

我认为 ResultSetExtractor 可能有优势的一个地方是当你有一个结果集（比如来自对存储过程的调用）和一个行映射器，并且想要像在 jdbcTemplate 方法中那样处理它们，例如查询（字符串 sql，RowMapper rowMapper）。在这种情况下，您可以通过使用 ResultSetExtractor 而不是 RowMapper 来避免手动迭代结果集。

例如：

行映射器

```
ResultSet resultSet = cs.executeQuery();
int row = 0;
DateRowMapper dateRowMapper = new DateRowMapper();
List<String> dates = new ArrayList<>();
```

```
}
return dates;
```

结果集提取器

```
ResultSet resultSet = callableStatement.executeQuery();
return new RowMapperResultSetExtractor<>(new DateRowMapper()).extractData(resultSet);
```

分享 改进这个答案 跟随

18 年 5 月 24 日 12:32编辑

托盖皮
63 1 4

17 年 7 月 27 日 15:56 回答

毫米波
71 1 3