# 42\_基于 JDBCTemplate 模板方式操作数据库,解决代码重复问题

#### 1、开篇

前面的课程在 JDBC 的基础上加入了数据库连接池,提高了访问数据库的效率。不过,作为 JDBC 而言需要使用 PreparedStatement 设置动态的 SQL 语句,同时需要设置 SQL 语句对应的参数,为了进一步简化对于数据库的开发工作,今天引入 JDBCTemplate,要介绍的内容包括:

- JDBCTemplate 简介
- JDBCTemplate 在项目中的实现

### 2、JDBCTemplate 简介

JDBCTemplate 是 Spring JDBC 抽象框架提供的 JDBC 模板类,其中 JDBCTemplate 是 core 包的核心类,其他模板类都是基于它封装完成的。

JDBCTemplate 类通过模板设计模式帮助消除了冗长的代码,只做需要的事: SQL 语句的编写,并且帮我们做固定的事:数据库连接的创建及关闭。

如图 1 所示,在 pom.xml 文件中加入 spring-jdbc 的依赖就可以在项目中使用 JDBCTemplate 了。



图 1 pom 文件中引入 spring-jdbc 的依赖

在实际工作中, JDBCTemplate 主要提供以下五类方法访问数据库:

- execute 方法:可以用于执行任何 SQL 语句,一般用于执行 DDL 语句;
- update 方法: 用于执行新增、修改、删除等语句;
- batchUpdate: 方法用于执行批处理的 update 语句;
- query 方法: 用于执行查询相关语句;
- call 方法:用于执行存储过程、函数相关语句。
   下面通过修改项目代码引入 JDBCTemplate,看看我们是如何使用它的。

## 3、JDBCTemplate 在项目中的实现

如图 2 所示,在/spring/dao 目录下面建立一个 TemplateBaseDao 类,在这个类中去实现对于 JDBCTemplate 的操作。这里通过使用 jdbcTemplate 中的queryForList 方法,传入 SQL 语句对数据库进行查询。如果需要查询参数就需要将 params 传入方法中。方法返回的值直接就是一个 List<Map<String, Object>>类型的对象。

```
java

com
                                               19
20
21
22
23
24
25
26
27
28
29
32
                                                         public class TemplateBaseDao {

▼ Imruyuan

                                                               private static final Logger LOGGER = LoggerFactorv.getLogger(TemplateBaseDao.class);
       ▼ 🛅 little
             spring
                                                                * jdbc的Template
               ▶ ■ aspect
                                                              private JdbcTemplate jdbcTemplate;
                  controller 🖿

    HealthControlle

                                                              public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate = jdbcTemplate; }
                     © 🖆 JdbcDemoController
               ▼ 🛅 dao

© 🚡 JdbcBaseDao
                                                               public int templateUpdate(String sql, Object[] params) {...}
                                               40
46
47
48
49
50
51
52
53
54
55
56
57
58
59
                    © № TemplateBaseDao

▼ Image: dto

                     © 1 Teacher
                                                                * 杏油
               ► enums
► exception
                                                                  @param sql 查询:
@param params 参数
@return 结果
                                                                                      查询sql
               handler
   mapper mapper
                                                              public List<Map<String, Object>> templateQuery(String sql, Object[] params) {
   if (params == null) {
      return this.jdbcTemplate.queryForList(sql);
}
    teacherMapper.xml
    logback.properties
    logback.xml
mybatis-config.xml
                                                                     return this.jdbcTemplate.queryForList(sql, params);
    🕦 spring-jdbc.xml
```

图 2 创建 TemplateBaseDao 类

从上面的代码可以看出并不需要像 JDBC 那样打开数据库连接,关闭数据库连接,同时还需要去包装返回对象等等操作。看完实现代码再回头看看配置文件中是如何对其进行实体化的。

如图 3 所示,在 spring-jdbc.xml 文件中,定义了 templateBaseDao 的 bean 它引用了 jdbcTemplate bean,然而 jdbcTemplate 又引用了 dataSource 的数据库连接池。这样的引用方式也说明了,JDBCTemplate 是实现了数据库连接池的。

图 3 spring-jdbc.xml 配置文件

好了最后来看看调用的入口,如图 4 所示,在 JdbcDemoController 中有一个 templateGetAll 方法。其中定义了查询 t\_teacher 表的语句,直接传给 templateBaseDao 中的 templateQuery 方法,该方法返回的对象直接返回给调用

```
端。
 spring
                                                 tist = japcmasevao.queryList(sqt, params: nutt, reacher.c
                                             } catch (Exception e) {
   log.error("查询教师列表失败",e);
   aspect
                           106
   constants
                           107

▼ □ controller

                            108
                                             return CommonResponse.success(list);
       © 1/2 HealthController
                            109
      C & JdbcDemoController
                            110
       😊 🖫 JdbcBaseDao
                                         @RequestMapping("/template/all")
       💿 🖫 TemplateBaseDao
                            113 🙉
                                         public CommonResponse templateGetAll(){
   ▼ 🛅 dto
                            114
                                             String sql = "select * from spring.t_teacher";
       © 🖫 Teacher
                                             List list = null;
                                             try {
   ▶ ■ enums
                           116
                                                 list = templateBaseDao.templateQuery(sql, params: null);
   exception
                           117
   ▶ 🖿 handler
                                             } catch (Exception e) {
                           118
                                                 log.error("查询教师列表失败",e);
   mapper
                           119
                           121
                                             return CommonResponse.success(list);
icherMapper.xml
roperties
                                         @RequestMapping("/mybatis/all")
ck.properties
                           124
                           125 👰
                                        public CommonResponse mybatisGetAll(){
ck.xml
is-config.xml
                           126
                                             List<Teacher> list = sqlSession.selectList( s: "com.ruyuan.lit
-idhe v
                                             raturn CommonResnance success(list)
```

图 4 JdbcDemoController 中的 templateGetAll 方法

#### 4、总结

本节课介绍了 JDBCTemplate 以及它对应的几种数据库操作方法,然后在互联网教育系统中加入了 TemplateBaseDao 类用来执行 JDBCTemplate 的相关操作。下节课,我们会尝试运行代码看看结果,并且思考一下 JDBC 还存在哪些问题,有哪些办法可以进行提升的。我们下期见,拜拜。