#### Spring Boot

##### 之Spring DAO- JDBC实现

问题描述

Spring Boot 2.2.5

JDK 8

数据库是海量数据存储、管理的强大工具，是APP后台不可或缺的基础设施。Spring框架提供了面向对象的数据库接口（DAO）、实体-关系映射（ORM）组件的整合方法

这篇技术笔记记录了JDBC实现方式

在Spring5全注解配置中，@Repository注解与@Resource注解配对使用：@Repository注解用value值在上下文中注册DAO对象；@Resource注解使用name值在Service类中注入DAO对象

解决

1. 标记DAO

@Repository(value = "jdbcProductDao")  
public class ProductsDaoI implements ProductsDao {  
  
 @Autowired  
 private JdbcTemplate template;  
  
 public Product queryById(Long id) throws Exception {  
 // string builder 支持链式调用.  
 StringBuilder sb = new StringBuilder("SELECT \* FROM products WHERE id=")  
 .append(id);  
 try {  
 return template.query(sb.toString(), new RowMapper<Product>(){  
  
 @Override  
 public Product mapRow(ResultSet resultSet, int i) throws SQLException {  
 Long id = resultSet.getLong(1);  
 String name = resultSet.getString(2);  
 String description = resultSet.getString(3);  
 Long price = resultSet.getLong(4);  
 return new Product(id, name, description, price);  
 }  
 }).get(0);  
 } catch (Exception e) {  
 throw new Exception();  
 }  
 }  
}

1. 标记实现类

@Service  
public class JdbcDaoService {  
  
 @Resource(name = "jdbcProductDao")  
 private ProductsDao jdbcProductDao;  
  
 public Product queryById(Long id) throws Exception {  
 return jdbcProductDao.queryById(id);  
 }  
}

1. 异常处理

笔记

1. 私有框架可实现SqlTemplateManager类，将SQL碎块重构到XML文件中集中配置
2. The best way to guarantee that your Data Access Objects (DAOs) or repositories provide exception translation is to use the @Repository annotation. This annotation also lets the component scanning support find and configure your DAOs and repositories without having to provide XML configuration entries for them.
3. JDBC实现是项目中可用的方案

□