2024-09-27

应用系统体系架构 — 作业2

# 作业要求

请你在大二开发的E-BookStore系统的基础上，完成下列任务：

## 针对现有E-BookStore中下订单功能，进行事务控制，具体要求为：

### 假设你的代码中在下订单的Service中，需要完成两件事情，在数据库的Order表中插入一条记录，在OrderItem表中插入多条记录。例如，如果一个订单包含3种不同的书，那么在Order表中会插入一条记录，在OrderItem表中会插入3条记录；

### 按照上述操作，Service需要调用OrderDao和OrderItemDao两个对象的响应方法来实现下订单，而且必须保证两个表的数据要么都插入成功，要么都不插入，即这两个表的插入操作必须在一个事务中完成；

### 如果你之前的系统不是这样实现的，那么就按照上述要求重构你的代码以满足要求；

### 请你参照上课讲解的transfer样例，通过声明方式实现上述事务管理功能，并按照课件中第25页所列表格通过修改@Transactional中的事务传播属性值来观察不同的执行结果，在文档中记录下结果，并解释为什么会出现这样的结果，将文档与代码一起上传。

## 参照上课给的样例，编写通过Kafka消息中间件处理订单的功能的前端，具体要求为：

### 你可以选择用以下三种方式中的任意一种实现下订单结果在前端呈现：

#### 在前端工程中，使用JavaScript监听订单处理结果消息发送到的Topic，然后刷新页面，可以参照<https://www.jianshu.com/p/d6b803fa808a> 上给出的代码；

#### 在前端发送Ajax请求获取订单的最新状态，后端接收到请求后将订单状态返回给前端去显示；

#### 采用WebSocket方式，后端的消息监听器类监听到消息处理结果Topic中的消息后，通过WebSocket发送给前端。

### 你应该将上述功能集成到你的E-Book系统中，如果你无法将上述功能集成到你的E-Book系统中，可以单独建立工程实现，但是会适当扣分。

### 请将你编写的相关代码整体压缩后上传，请勿压缩整个工程提交。

### 请你编写一个文档，解释你的程序设计方案，包括消息格式、Topic配置项、订单处理结果在前端呈现的方式，并且要对3种前端呈现方式的优缺点做出你的分析。将文档与代码一起上传。

# 提交要求

## 请将你自己编写的源码、脚本和文档压缩后上传，勿压缩整个工程提交，尤其是不要压缩第三方的Jar包。

# 评分标准：

## 正确完成“作业要求A”中的功能（1分）

## 正确解释所使用的Transactional传播属性值的原因 (1分)

## 正确地将订单处理的结果返回给前端，并且能够在前端呈现（2分）

## 对3种前端呈现方式的优缺点做出你的分析和对比（1分）

## 如果不能将作业要求整合在你的E-BookStore中，那么请开发单独的针对作业要求完成的独立工程并提交，但是需要扣分，未整合到系统中的作业要求扣1分。