

应用系统体系架构 — 作业 2

1. 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：

A. 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，具体要求为：

- i. 前端发送的下清单请求被后端的 Controller 接收，然后按照你设计的消息格式，组装下订单消息，并将其发送到 Kafka 中你预先创建好的 Topic 中；
- ii. 在后端增加类似样例中用@Component 注解的消息监听器类，监听指定 Topic 中的下订单消息；
- iii. 在监听到下订单消息后，监听器类会调用下订单服务完成下订单，并且将下订单的处理结果发送到预先创建好的另一个 Topic 中；其中，下订单服务应该与同步处理下订单功能复用相同的 Service，以提高代码的可维护性；
- iv. 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：
 - 1) 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；
 - 2) 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；
 - 3) 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；此方式为第 5 课的内容。
- v. 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。
- vi. 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。

- vii. 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。

2. 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。

3. 评分标准：

- A. 正确完成异步“下订单”的功能，即能够在接收到前端请求后，经过你自己设计的逻辑生成订单，并将其持久化到数据库中（1.5 分）
- B. 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）
 - i. 注：作业要求中提到的三种实现方案都可以满足要求，无论哪一种，你都需要把处理结果发送到另一个 Topic 中，为第 5 课后的作业做准备。
- C. 对 3 种前端呈现方式的优缺点做出你的分析和对比（1.5 分）
- D. 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。