

82_代码实战：编写代码实现支付订单和完成订单功能

儒猿架构官网上线，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：www.ruyuan2020.com（建议 PC 端访问）

1、开篇

上周的课程介绍了互联网教育系统取消订单的功能,其中描述了主动取消订单和超时取消订单的场景，并且使用 RocketMQ 的延迟消息完成了超时取消订单的功能。本周的课程为延续后面的业务流程，订单创建之后会经历支付订单、完成订单、评论订单的环节，依旧会用到 RocketMQ 的相关技术。今天课程的内容包括以下几个部分：

- 修改 OrderController
- 修改 OrderService 和 OrderServiceImpl
- 修改 OrderMapper 接口与实现

2、修改 OrderController

由于订单的 Controller、Service 以及 Mapper 在前面几节课中都已经创建了，本节课的代码修改基本是在原有的基础上面进行调整。首先来看 OrderController 的部分，如图 1 所示，在 OrderController 类中增加了三个入口函数的调用，分别是 payOrder、finish 和 comment。其中 payOrder 方法是用来订单支付操作的，它会去调用 orderService 中的 payOrder 方法完成订单支付的功能。

finishOrder 方法会调用 orderService 中的 finishOrder 方法传入 orderId 用以完成订单操作。最后是 publishComment 方法，它是用来评论订单的，同样也会调用 orderService 中的 publishComment 方法。



图 1 OrderController 修改

3、修改 OrderService 和 OrderServiceImpl

在 OrderController 中大量地用到了 OrderService 中的方法，由于 OrderService 是一个接口其具体的实现都放在 OrderServiceImpl 中，因此我们将目光聚焦到 OrderServiceImpl 上。如图 2 所示，在 OrderServiceImpl 中对应着 OrderController 中调用的 3 个方法。

payOrder 方法会判断只有订单状态为“等待支付”，就会调用 orderMapper 将订单的状态修改为“已支付”。

publishComment 方法会通过 buiderComment 方法将订单的评论信息进行封装，然后将该评论信息通过 commentMapper 保存到数据库中。

finishOrder 方法先会判断是否为己支付状态，在已支付状态的情况下才对订单的状态进行修改，将其修改为“等待评论”。随后通过 consumerService 中的 orderFinishInformReceiveCredits 方法更新积分信息，也就是增加消费者的积分。也会通过 couponService 中的 orderFinishReceiverCoupon 方法更新优惠券信息，也就是给消费者发放优惠券。



图 2 OrderServiceImpl 修改

4、修改 OrderMapper 接口与实现

说完了 OrderController 和 OrderService 就顺势来到了 OrderMapper，一般而言 OrderMapper 也是一个通过 Mybatis 访问数据库的接口，与之相对应地有一个 orderMapper.xml 文件是用来对接口进行实现的，说白了就是存放接口方法中访问数据库的 SQL 语句。

如图 3 所示，可以看到 payOrder 方法是通过 update t_order 表对 status 进行更新的。

finishOrder 是通过 update t_order 表完成订单“完成”状态的更新。最后 publishComment 是通过 update t_order 表中的 comment_id 字段对评论进行更新的，在做这个操作之前会先 insert t_comment 记录到表中然后获取新增的 comment_id 再到 t_order 中更新。



图 3 OrderMapper.xml 修改

5、总结

本节课延续上节课创建订单的后续流程，完成了对应的代码实现。主要介绍了 OrderController、OrderServiceImpl、OrderMapper 中对应代码的修改，其中包括支付订单、完成订单、评论订单的业务代码。下期会思考上述代码会有哪些问题，并提出基于 MQ 对完成订单模块异步化改造的方案，从而提升系统性能。这里将[代码](#)给大家，下期见，拜拜。