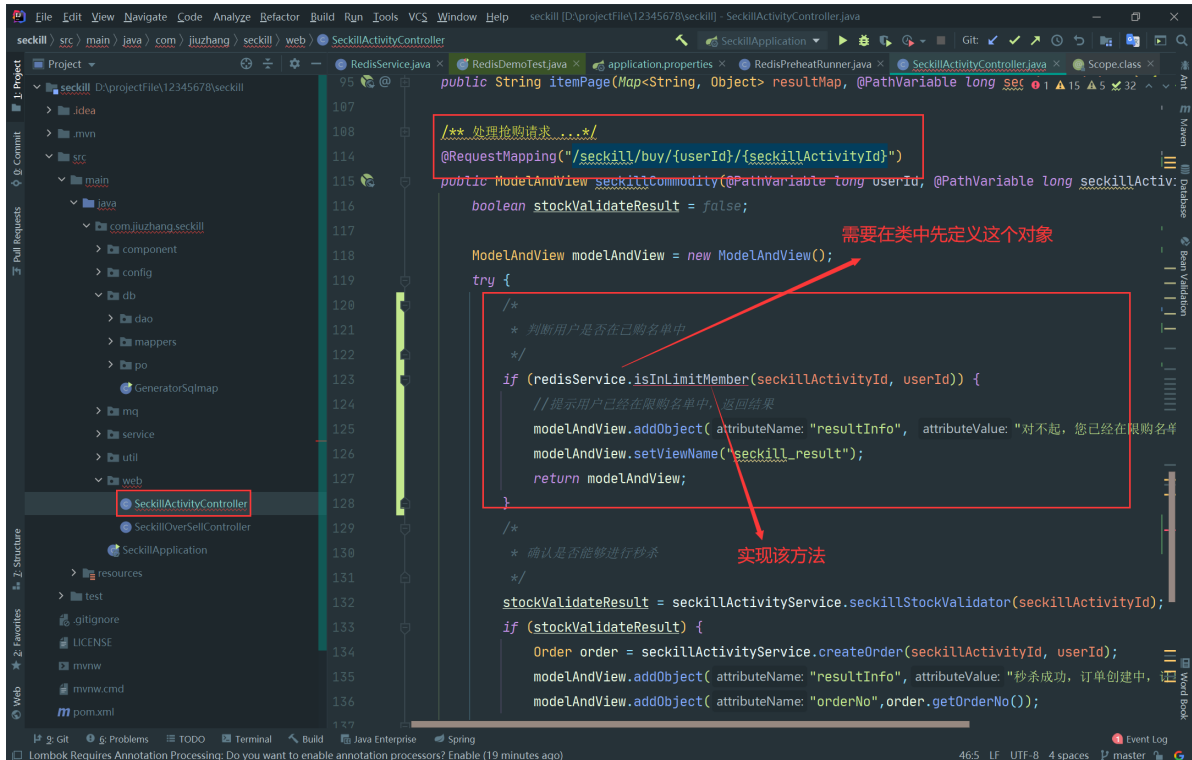


《保姆级教程》第十一集 限购处理 & 付款库存扣减

1. 首先在下单前进行限购判断

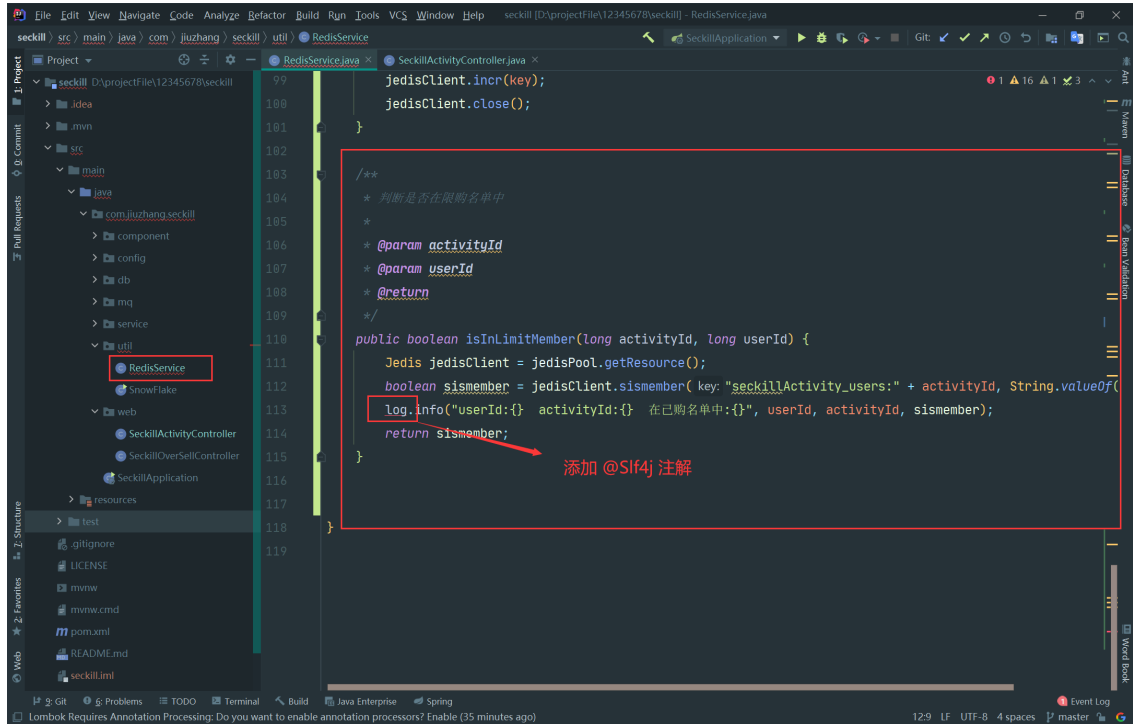
1. 限购用户进行判断



```
@Resource
private RedisService redisService;
```

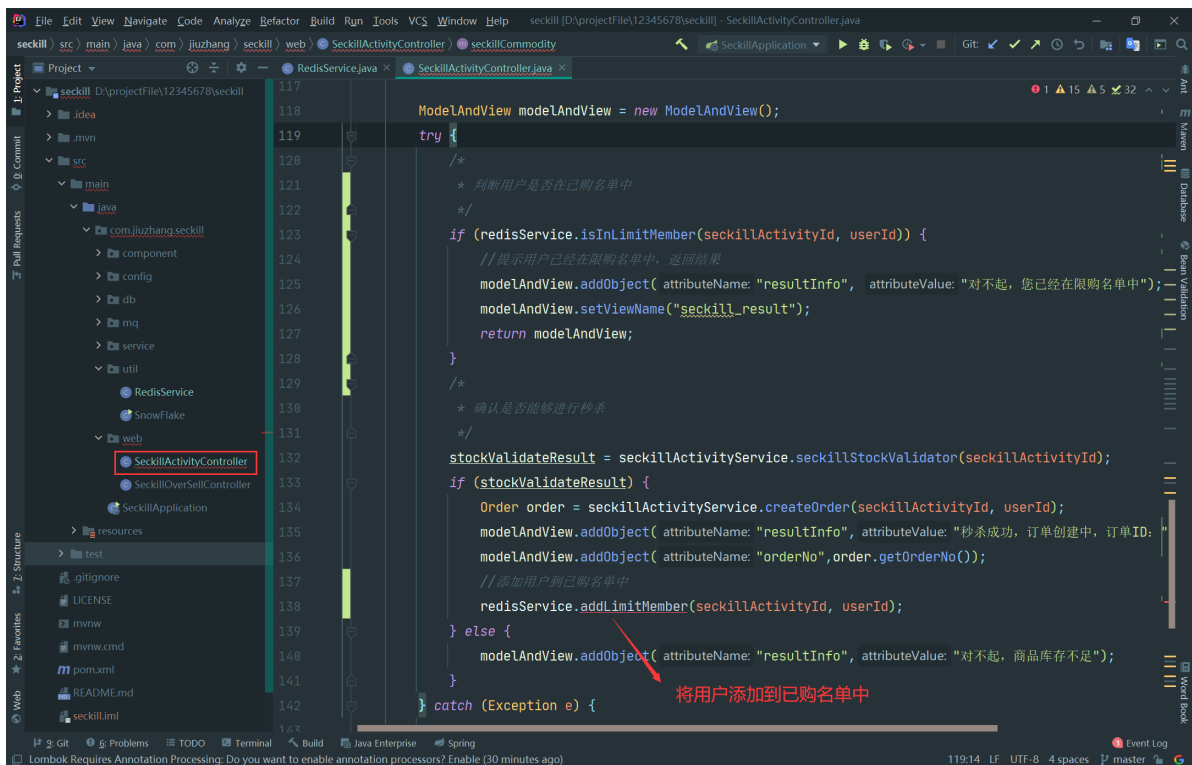
```
/**
 * 判断用户是否在已购名单中
 */
if (redisService.isInLimitMember(seckillActivityId, userId)) {
    // 提示用户已经在限购名单中, 返回结果
    modelAndView.addObject("resultInfo", "对不起, 您已经在限购名单中");
    modelAndView.setViewName("seckill_result");
    return modelAndView;
}
```

2. 实现 isLimitMember () 方法，判断是否在限购名单中



```
/**  
 * 判断是否在限购名单中  
 *  
 * @param activityId  
 * @param userId  
 * @return  
 */  
public boolean isLimitMember(long activityId, long userId) {  
    Jedis jedisClient = jedisPool.getResource();  
    boolean sismember = jedisClient.sismember("seckillActivity_users:" +  
activityId, String.valueOf(userId));  
    jedisClient.close();  
    log.info("userId:{} activityId:{} 在已购名单中:{}", userId, activityId,  
sismember);  
    return sismember;  
}
```

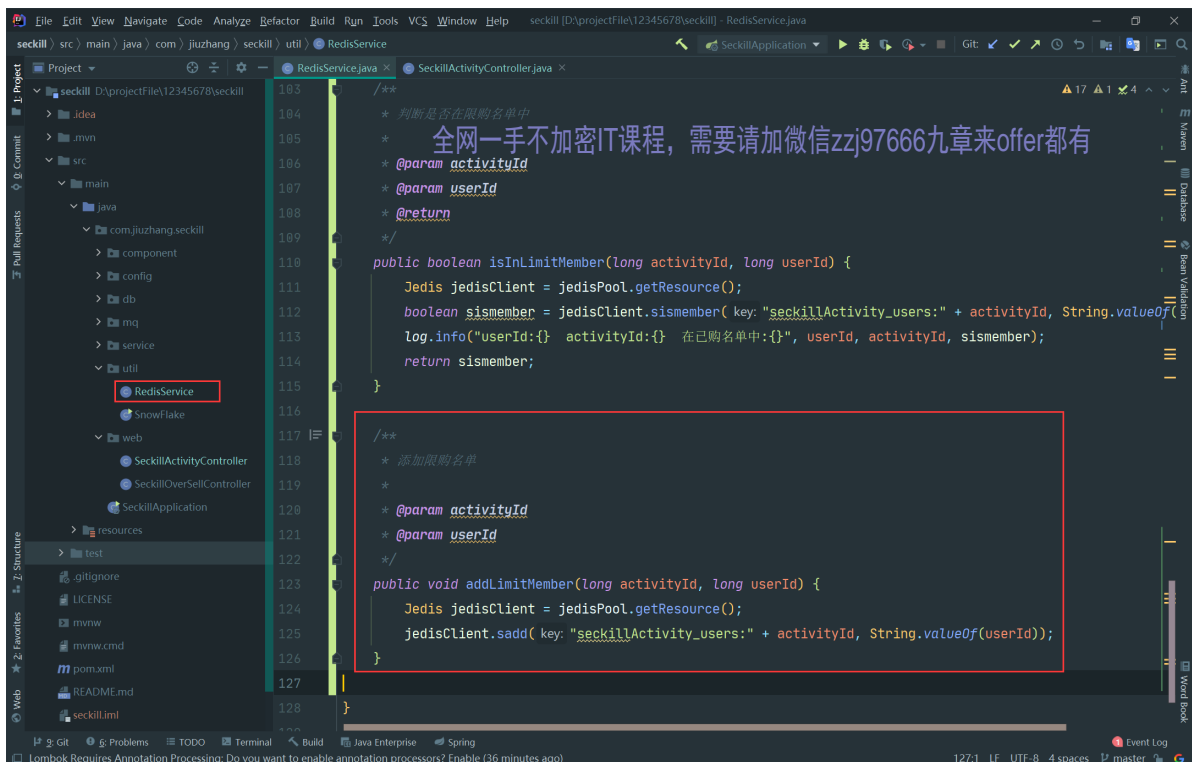
2. 添加限购功能



//添加用户到已购名单中

```
redisService.addLimitMember(seckillActivityId, userId);
```

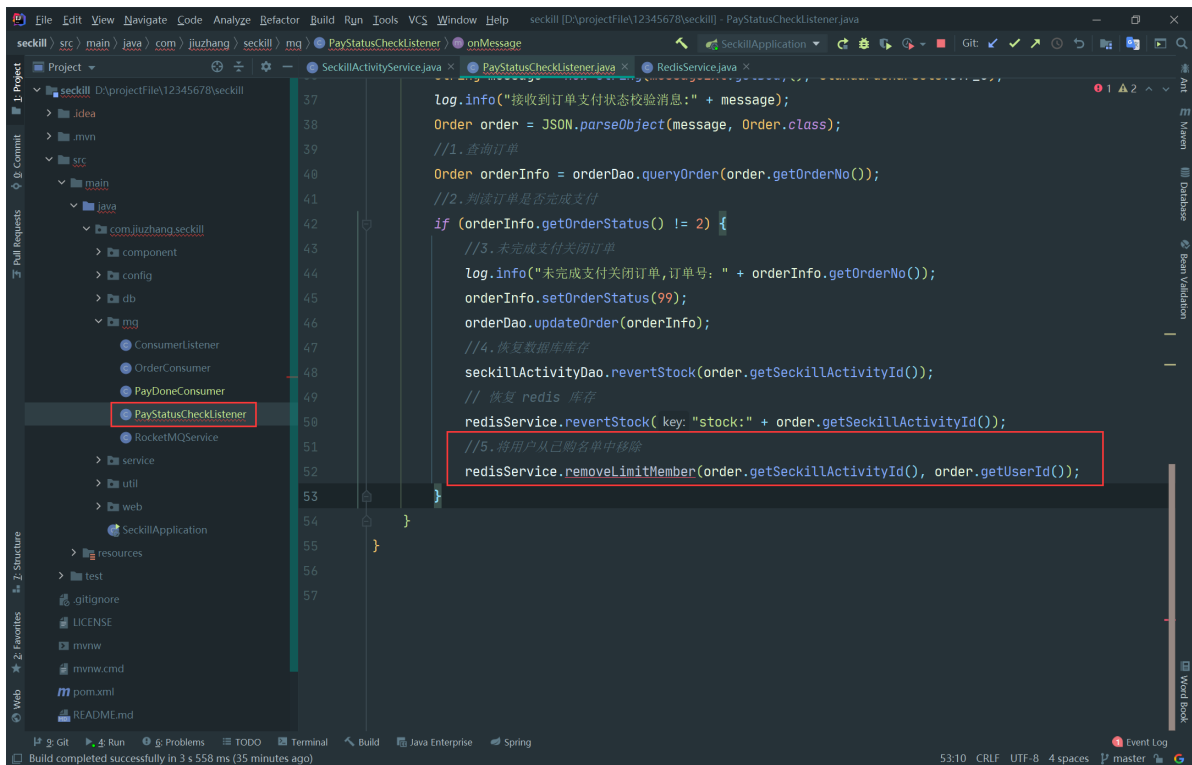
实现 addLimitMember () 方法, 将用户添加到已购名单中



```
/**
 * 添加限购名单
 *
 * @param activityId
 * @param userId
 */
public void addLimitMember(long activityId, long userId) {
    Jedis jedisClient = jedisPool.getResource();
    jedisClient.sadd("seckillActivity_users:" + activityId,
String.valueOf(userId));
    jedisClient.close();
}
```

3. 超时订单 清除用户限购

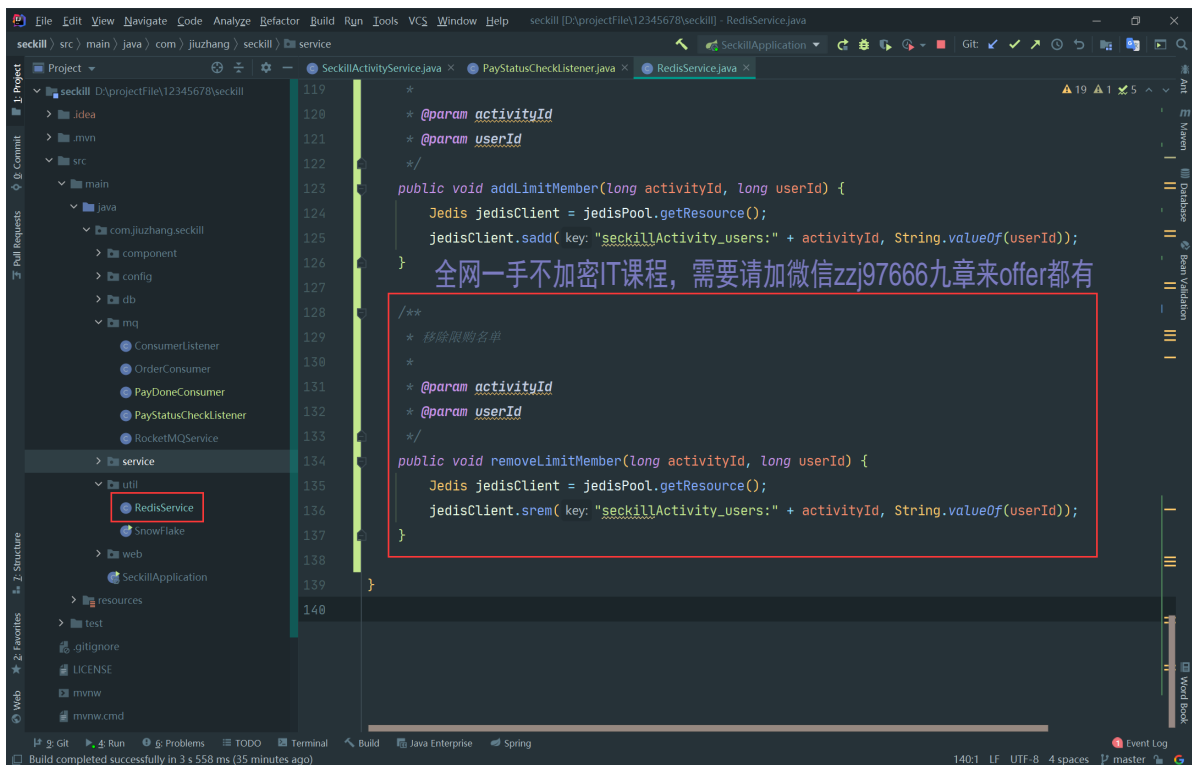
1. 超时取消用户限购



//5. 将用户从已购名单中移除

```
redisService.removeLimitMember(order.getSeckillActivityId(), order.getUserId());
```

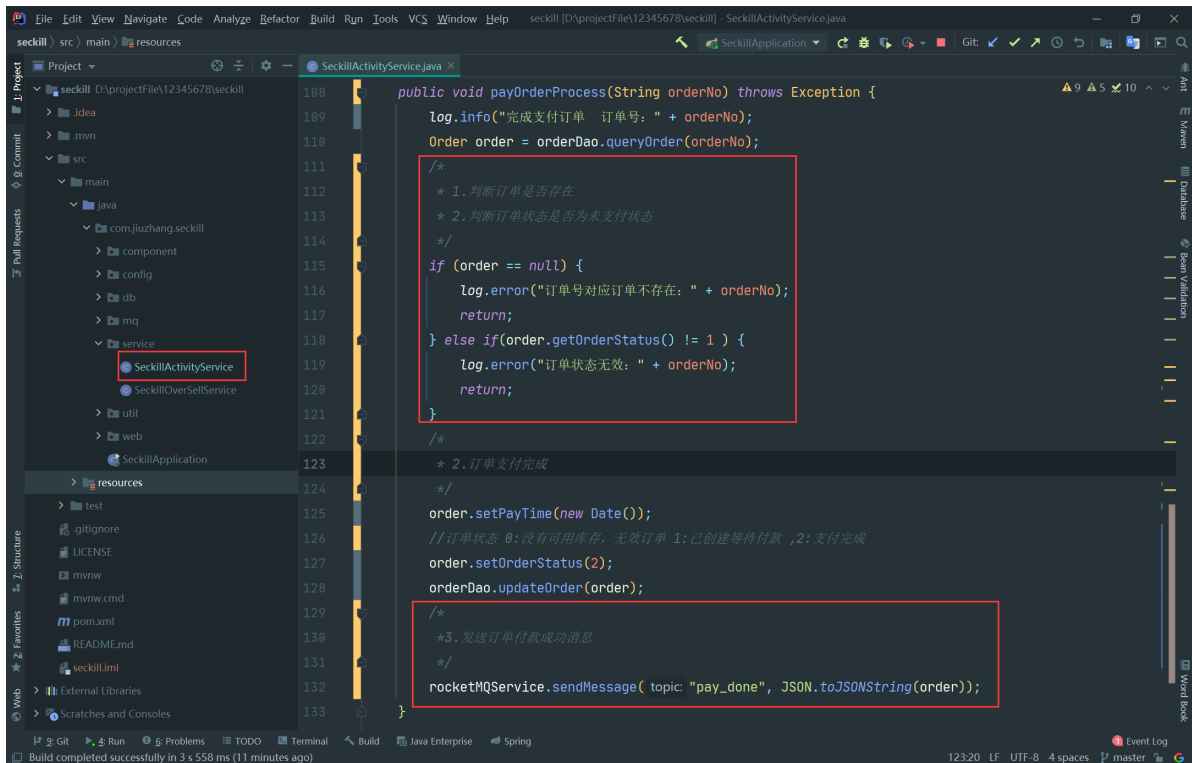
2. 编写 redisService.removeLimitMember() 方法，移除限购名单



```
/**
 * 移除限购名单
 *
 * @param activityId
 * @param userId
 */
public void removeLimitMember(long activityId, long userId) {
    Jedis jedisClient = jedisPool.getResource();
    jedisClient.srem("seckillActivity_users:" + activityId,
String.valueOf(userId));
    jedisClient.close();
}
```

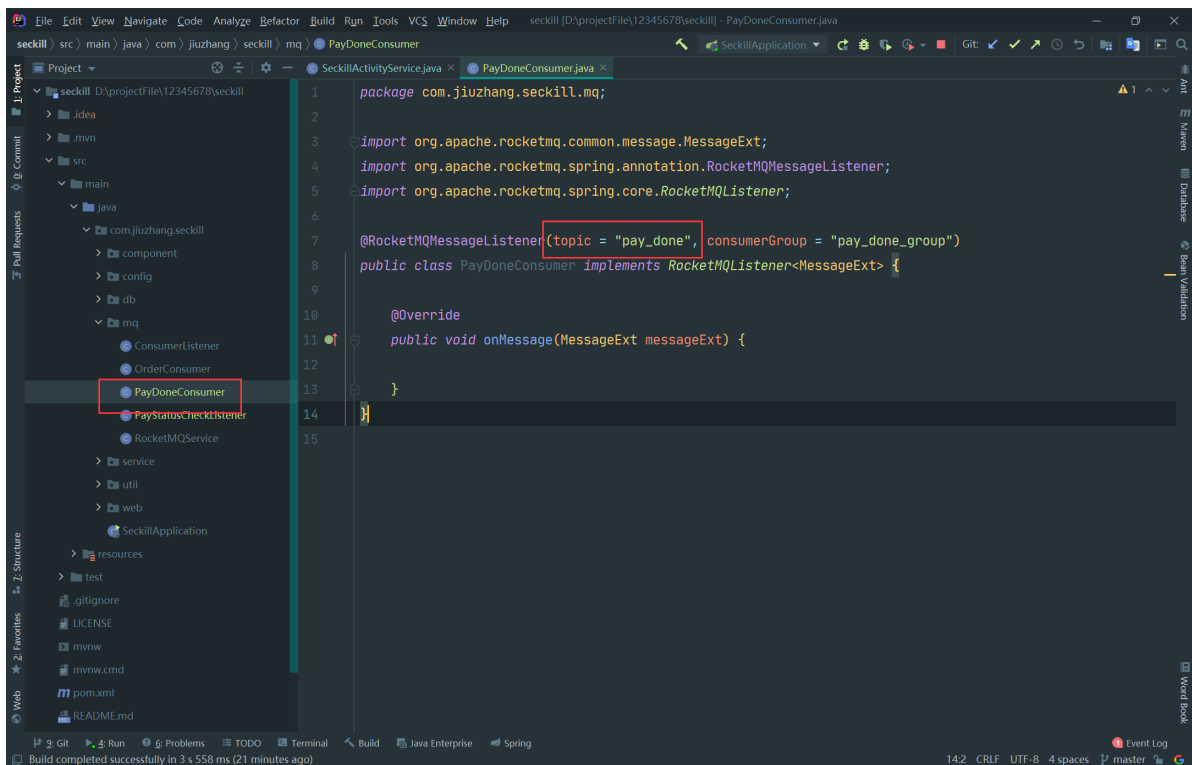
4. 付款处理方法升级

1. 修改 SeckillActivityService.payOrderProcess() 方法

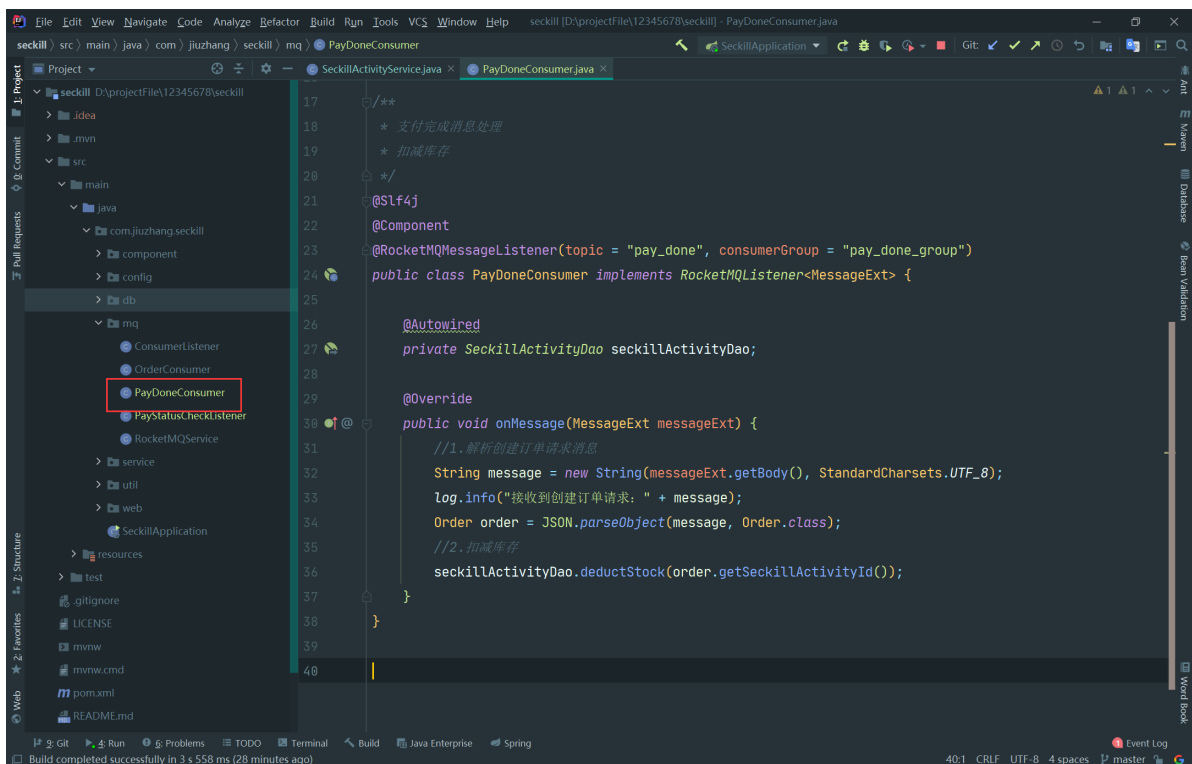


```
/**
 * 订单支付完成处理
 *
 * @param orderNo
 */
public void payOrderProcess(String orderNo) throws Exception {
    log.info("完成支付订单 订单号: " + orderNo);
    Order order = orderDao.queryOrder(orderNo);
    /**
     * 1. 判断订单是否存在
     * 2. 判断订单状态是否为未支付状态
     */
    if (order == null) {
        log.error("订单号对应订单不存在: " + orderNo);
        return;
    } else if (order.getOrderStatus() != 1) {
        log.error("订单状态无效: " + orderNo);
        return;
    }
    /**
     * 2. 订单支付完成
     */
    order.setPayTime(new Date());
    // 订单状态 0: 没有可用库存, 无效订单 1: 已创建等待付款, 2: 支付完成
    order.setOrderStatus(2);
    orderDao.updateOrder(order);
    /**
     * 3. 发送订单付款成功消息
     */
    rocketMQService.sendMessage("pay_done", JSON.toJSONString(order));
}
```

2. 添加 订单付款消息的消费者 PayDoneConsumer类



3. 编写 onMessage 方法 全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有



```
/**
 * 支付完成消息处理
 * 扣减库存
 */
@Slf4j
@Component
@RocketMQMessageListener(topic = "pay_done", consumerGroup = "pay_done_group")
public class PayDoneConsumer implements RocketMQListener<MessageExt> {

    @Autowired
    private SeckillActivityDao seckillActivityDao;

    @Override
    public void onMessage(MessageExt messageExt) {
        //1. 解析创建订单请求消息
        String message = new String(messageExt.getBody(), StandardCharsets.UTF_8);
        log.info("接收到创建订单请求: " + message);
        Order order = JSON.parseObject(message, Order.class);
        //2. 扣减库存
        seckillActivityDao.deductStock(order.getSeckillActivityId());
    }
}
```

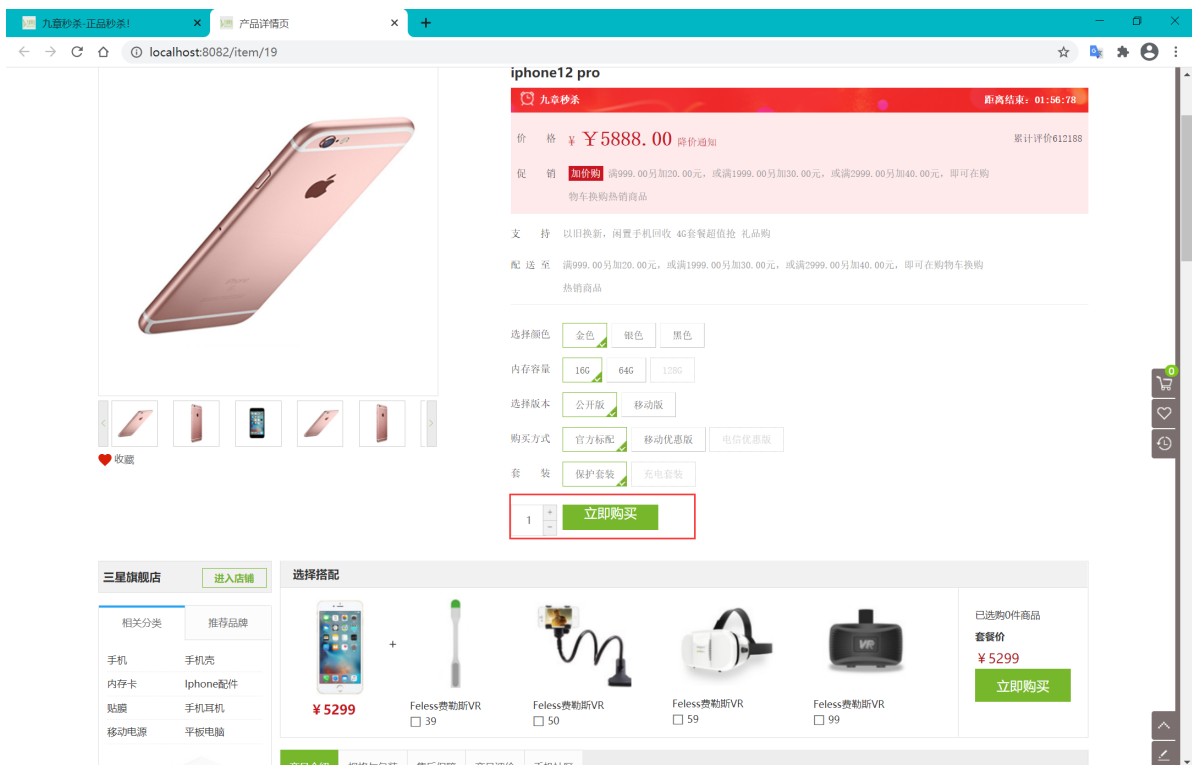
```

@Override
public void onMessage(MessageExt messageExt) {
    //1.解析创建订单请求消息
    String message = new String(messageExt.getBody(),
StandardCharsets.UTF_8);
    log.info("接收到创建订单请求: " + message);
    Order order = JSON.parseObject(message, Order.class);
    //2.扣减库存
    seckillActivityDao.deductStock(order.getSeckillActivityId());
}
}

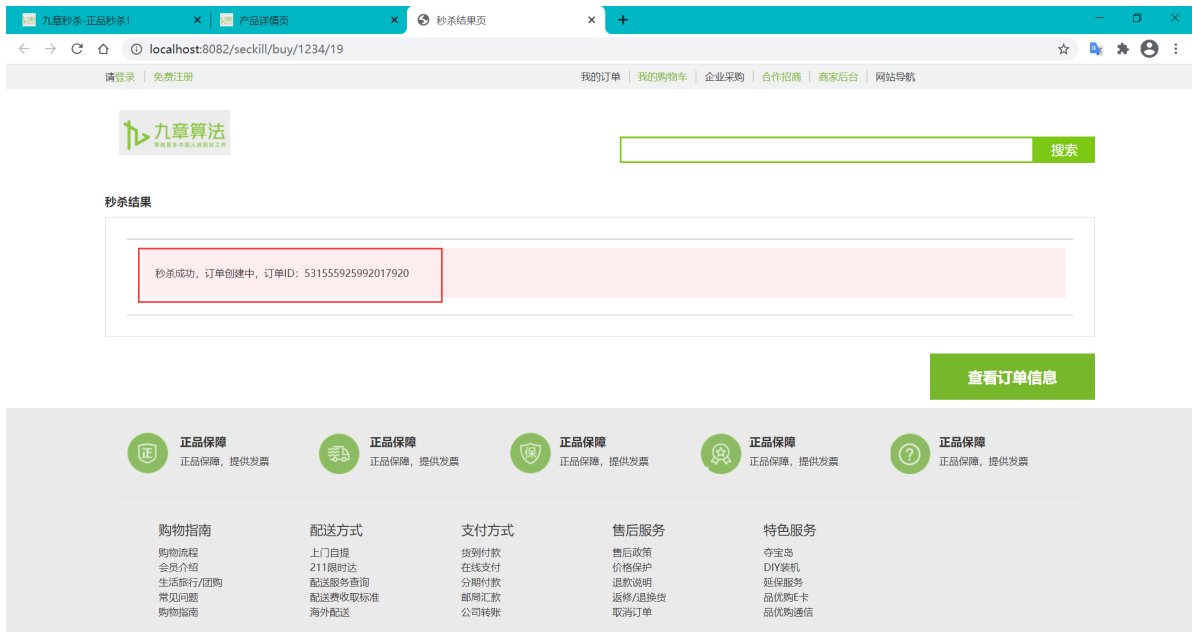
```

5. 测试功能 全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

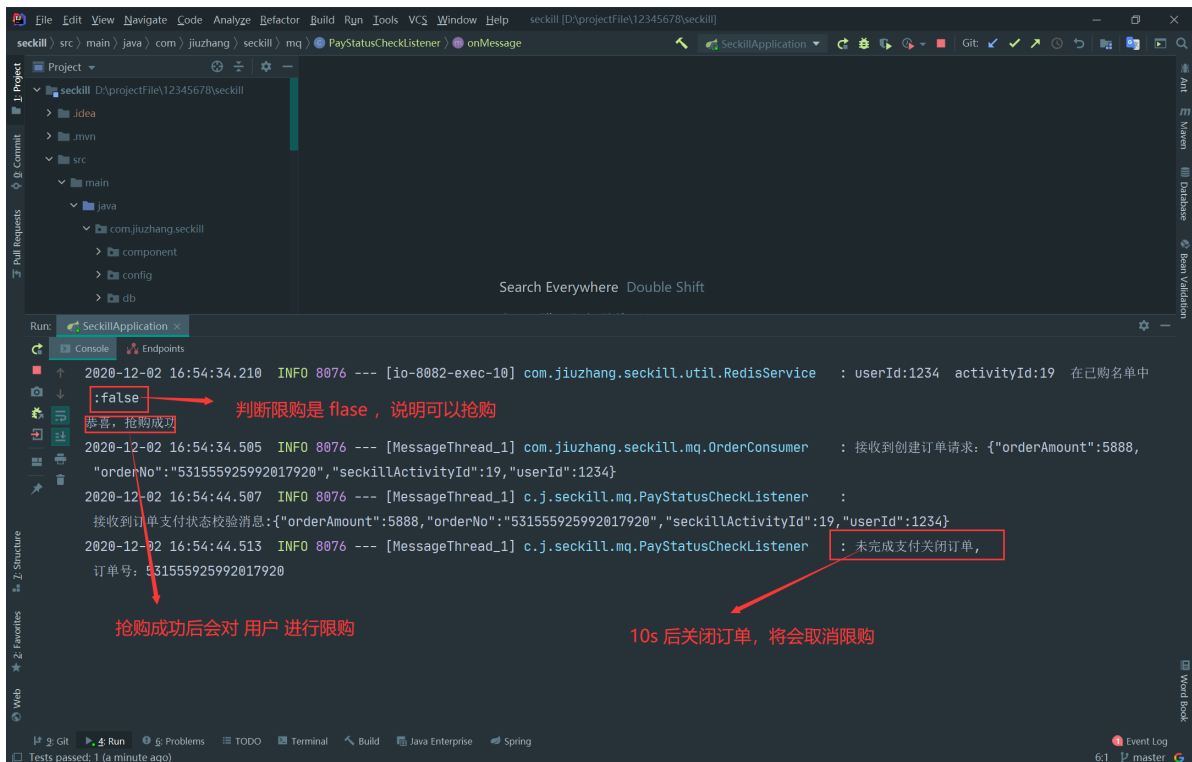
1. 启动项目，准备抢购



2. 点击抢购



3. 查看控制台

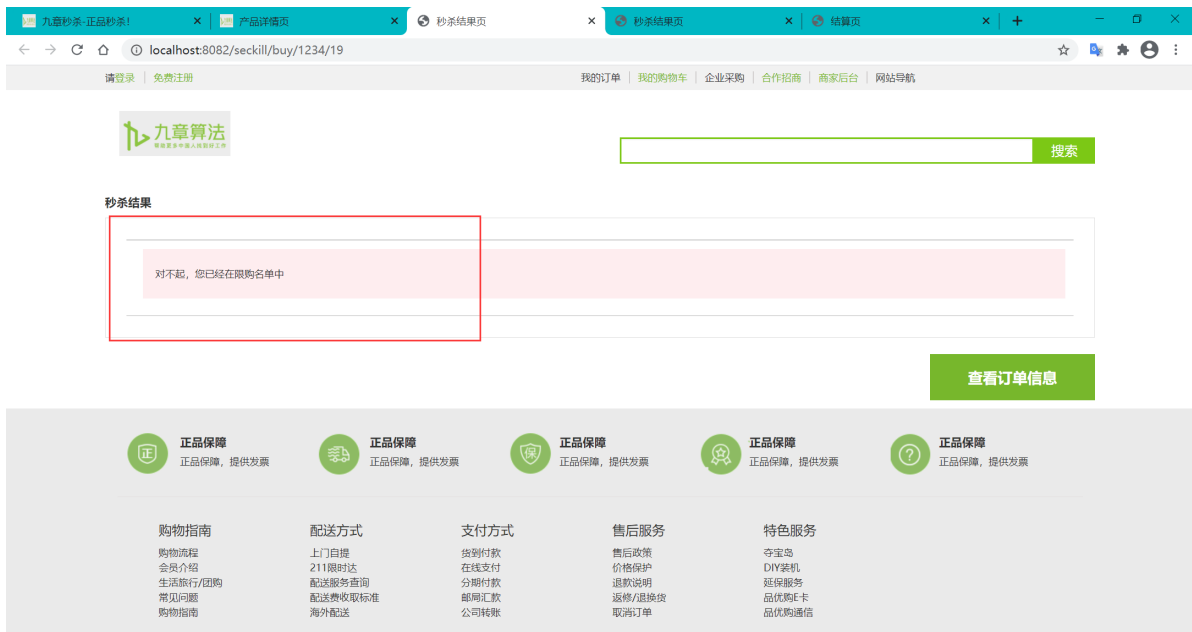


4. 我们再抢购一次，这次我们抢购后马上付款试试看下次还能抢购吗

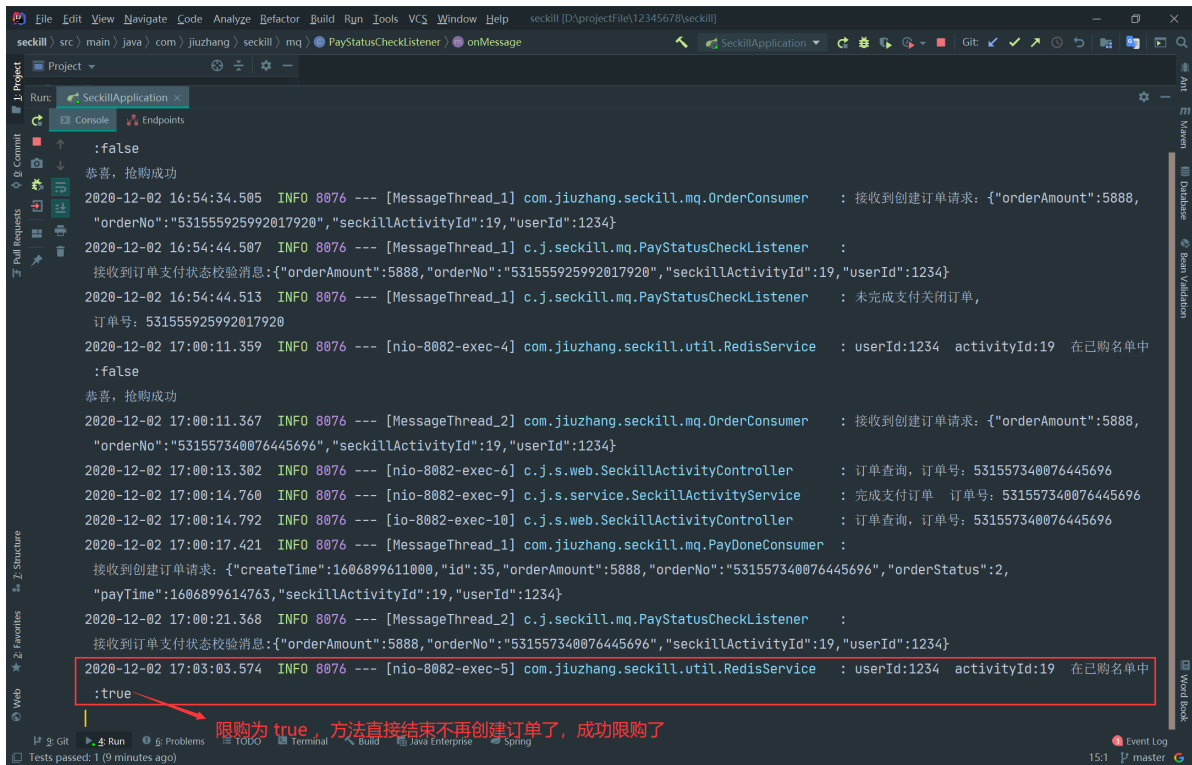
```

2020-12-02 16:54:34.210 INFO 8076 --- [io-8082-exec-10] com.jiuzhang.seckill.util.RedisService : userId:1234 activityId:19 在已购名单中
:false
恭喜, 抢购成功
2020-12-02 16:54:34.505 INFO 8076 --- [MessageThread_1] com.jiuzhang.seckill.mq.OrderConsumer : 接收到创建订单请求: {"orderAmount":5888,
"orderNo":"531555925992017920","seckillActivityId":19,"userId":1234}
2020-12-02 16:54:44.507 INFO 8076 --- [MessageThread_1] c.j.seckill.mq.PayStatusCheckListener :
接收到订单支付状态校验消息:{"orderAmount":5888,"orderNo":"531555925992017920","seckillActivityId":19,"userId":1234}
2020-12-02 16:54:44.513 INFO 8076 --- [MessageThread_1] c.j.seckill.mq.PayStatusCheckListener : 未完成支付关闭订单,
订单号: 531555925992017920
2020-12-02 17:00:11.359 INFO 8076 --- [nio-8082-exec-4] com.jiuzhang.seckill.util.RedisService : userId:1234 activityId:19 在已购名单中
:false
恭喜, 抢购成功
2020-12-02 17:00:11.367 INFO 8076 --- [MessageThread_2] com.jiuzhang.seckill.mq.OrderConsumer : 接收到创建订单请求: {"orderAmount":5888,
"orderNo":"531557340076445696","seckillActivityId":19,"userId":1234}
2020-12-02 17:00:13.302 INFO 8076 --- [nio-8082-exec-6] c.j.s.web.SeckillActivityController : 订单查询, 订单号: 531557340076445696
2020-12-02 17:00:14.760 INFO 8076 --- [nio-8082-exec-9] c.j.s.service.SeckillActivityService : 完成支付订单 订单号: 531557340076445696
2020-12-02 17:00:14.792 INFO 8076 --- [io-8082-exec-10] c.j.s.web.SeckillActivityController : 订单查询, 订单号: 531557340076445696
2020-12-02 17:00:17.421 INFO 8076 --- [MessageThread_1] com.jiuzhang.seckill.mq.PayDoneConsumer :
接收到创建订单请求: {"createTime":1606899611000,"id":35,"orderAmount":5888,"orderNo":"531557340076445696","orderStatus":2,
"payTime":1606899614763,"seckillActivityId":19,"userId":1234}
2020-12-02 17:00:21.368 INFO 8076 --- [MessageThread_2] c.j.seckill.mq.PayStatusCheckListener :
接收到订单支付状态校验消息:{"orderAmount":5888,"orderNo":"531557340076445696","seckillActivityId":19,"userId":1234}
    
```

5. 再次抢购



6. 已经被限购了，我们再看看控制台



```

: false
恭喜，抢购成功
2020-12-02 16:54:34.505 INFO 8076 --- [MessageThread_1] com.jiuzhang.seckill.mq.OrderConsumer : 接收到创建订单请求: {"orderAmount":5888,
"orderNo":"531555925992017920","seckillActivityId":19,"userId":1234}
2020-12-02 16:54:44.507 INFO 8076 --- [MessageThread_1] c.j.seckill.mq.PayStatusCheckListener :
接收到订单支付状态校验消息: {"orderAmount":5888,"orderNo":"531555925992017920","seckillActivityId":19,"userId":1234}
2020-12-02 16:54:44.513 INFO 8076 --- [MessageThread_1] c.j.seckill.mq.PayStatusCheckListener : 未完成支付关闭订单,
订单号: 531555925992017920
2020-12-02 17:00:11.359 INFO 8076 --- [nio-8082-exec-4] com.jiuzhang.seckill.util.RedisService : userId:1234 activityId:19 在已购名单中
: false
恭喜，抢购成功
2020-12-02 17:00:11.367 INFO 8076 --- [MessageThread_2] com.jiuzhang.seckill.mq.OrderConsumer : 接收到创建订单请求: {"orderAmount":5888,
"orderNo":"531557340076445696","seckillActivityId":19,"userId":1234}
2020-12-02 17:00:13.302 INFO 8076 --- [nio-8082-exec-6] c.j.s.web.SeckillActivityController : 订单查询, 订单号: 531557340076445696
2020-12-02 17:00:14.760 INFO 8076 --- [nio-8082-exec-9] c.j.s.service.SeckillActivityService : 完成支付订单 订单号: 531557340076445696
2020-12-02 17:00:14.792 INFO 8076 --- [io-8082-exec-10] c.j.s.web.SeckillActivityController : 订单查询, 订单号: 531557340076445696
2020-12-02 17:00:17.421 INFO 8076 --- [MessageThread_1] com.jiuzhang.seckill.mq.PayDoneConsumer :
接收到创建订单请求: {"createTime":1606899611000,"id":35,"orderAmount":5888,"orderNo":"531557340076445696","orderStatus":2,
"payTime":1606899614763,"seckillActivityId":19,"userId":1234}
2020-12-02 17:00:21.368 INFO 8076 --- [MessageThread_2] c.j.seckill.mq.PayStatusCheckListener :
接收到订单支付状态校验消息: {"orderAmount":5888,"orderNo":"531557340076445696","seckillActivityId":19,"userId":1234}
2020-12-02 17:03:03.574 INFO 8076 --- [nio-8082-exec-5] com.jiuzhang.seckill.util.RedisService : userId:1234 activityId:19 在已购名单中
: true
    
```

限购为 true，方法直接结束不再创建订单了，成功限购了

好的，到这里限购功能和支付库存扣减功能升级就结束了，本章课程内容也结束了，我们开始下一章吧。