

85_阶段测试：提交代码后，再来验证支付订单和完成订单功能是否正确？

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

上节课使用 RocketMQ 实现了异步的订单完成功能，设计到延迟消息和及时消息的部分。修改代码包括：OrderServiceImpl、OrderEventManagerImpl、OrderDelayMessageListener、OrderMessageListener 等类文件。本节课带大家讲上述修改的代码进行测试。今天课程的内容包括以下几个部分：

- 打包服务
- 上传服务
- 启动服务
- 测试支付和完成订单功能

2、打包服务

这里直接对代码进行打包。如图 1 所示，打开我们熟悉的酒店管理后台应用服务，在 IntelliJ IDEA 中选择下方的“Terminal”按钮，在显示的命令行中（红色箭头的地方）输入对应的命令，用来对项目进行打包。

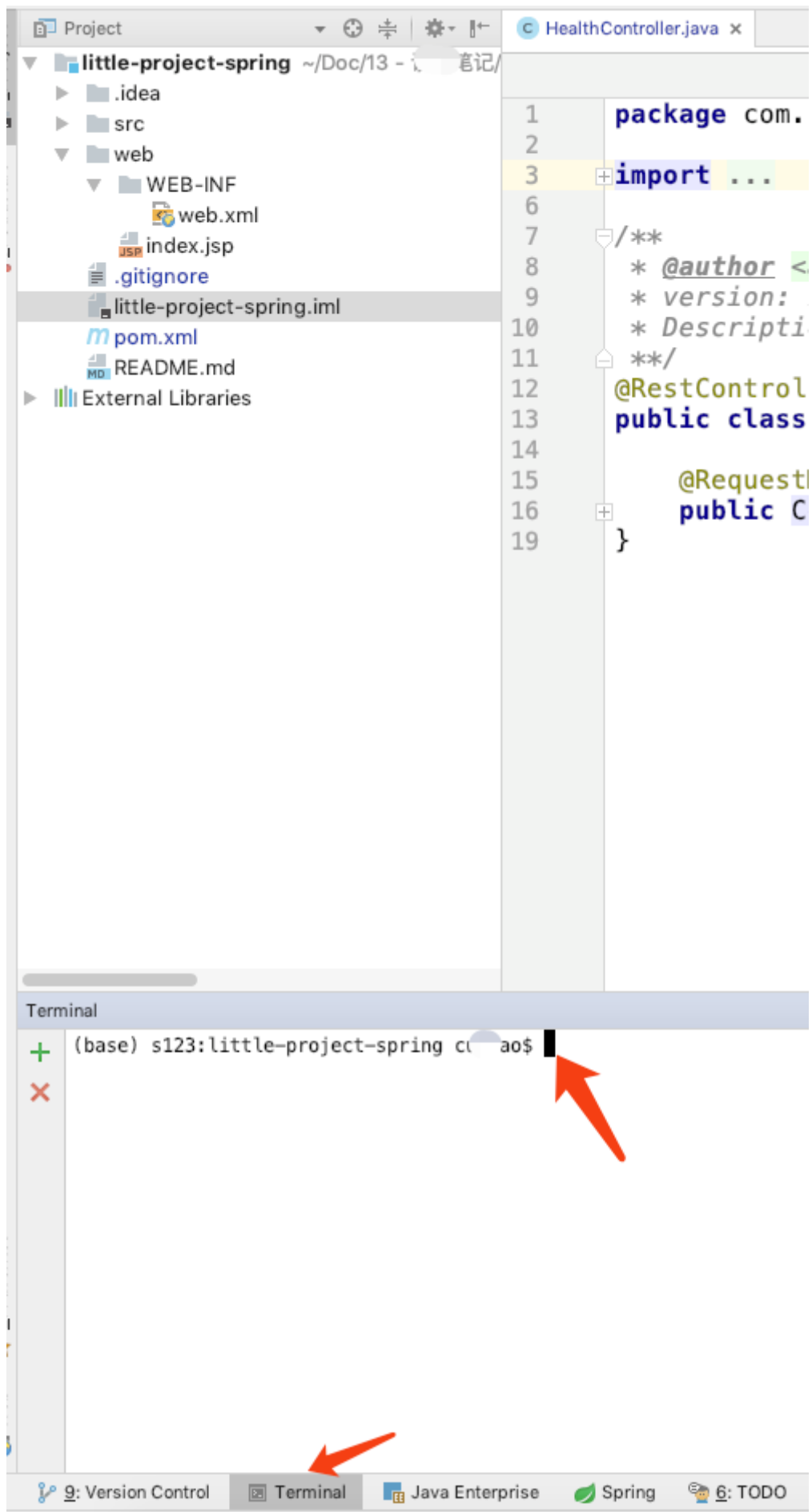


图 1 打开命令行

如图 2 所示，这里输入“mvn clean package”的命令。

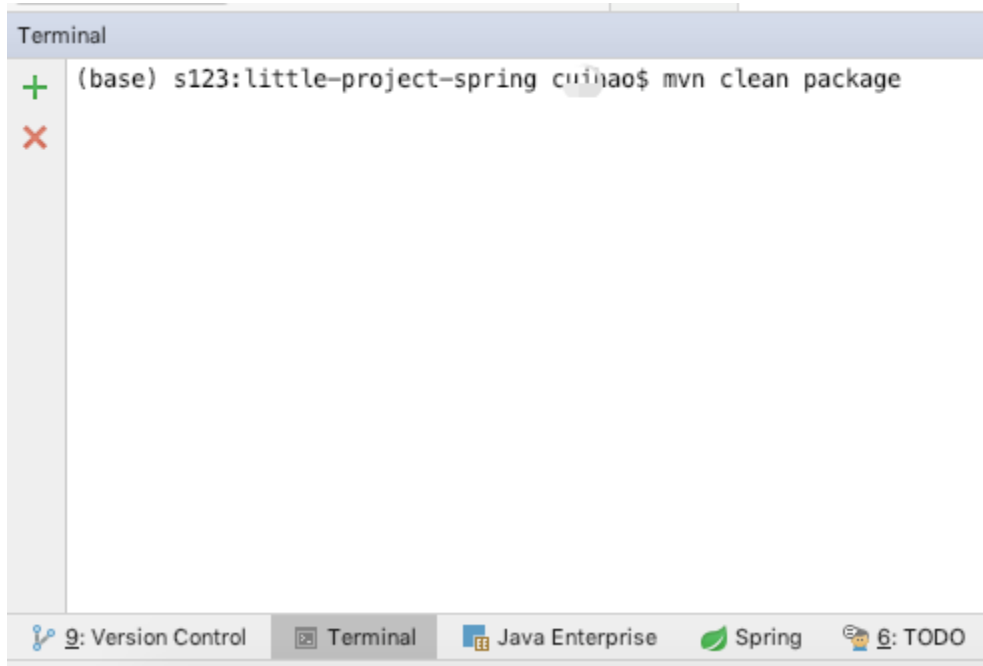
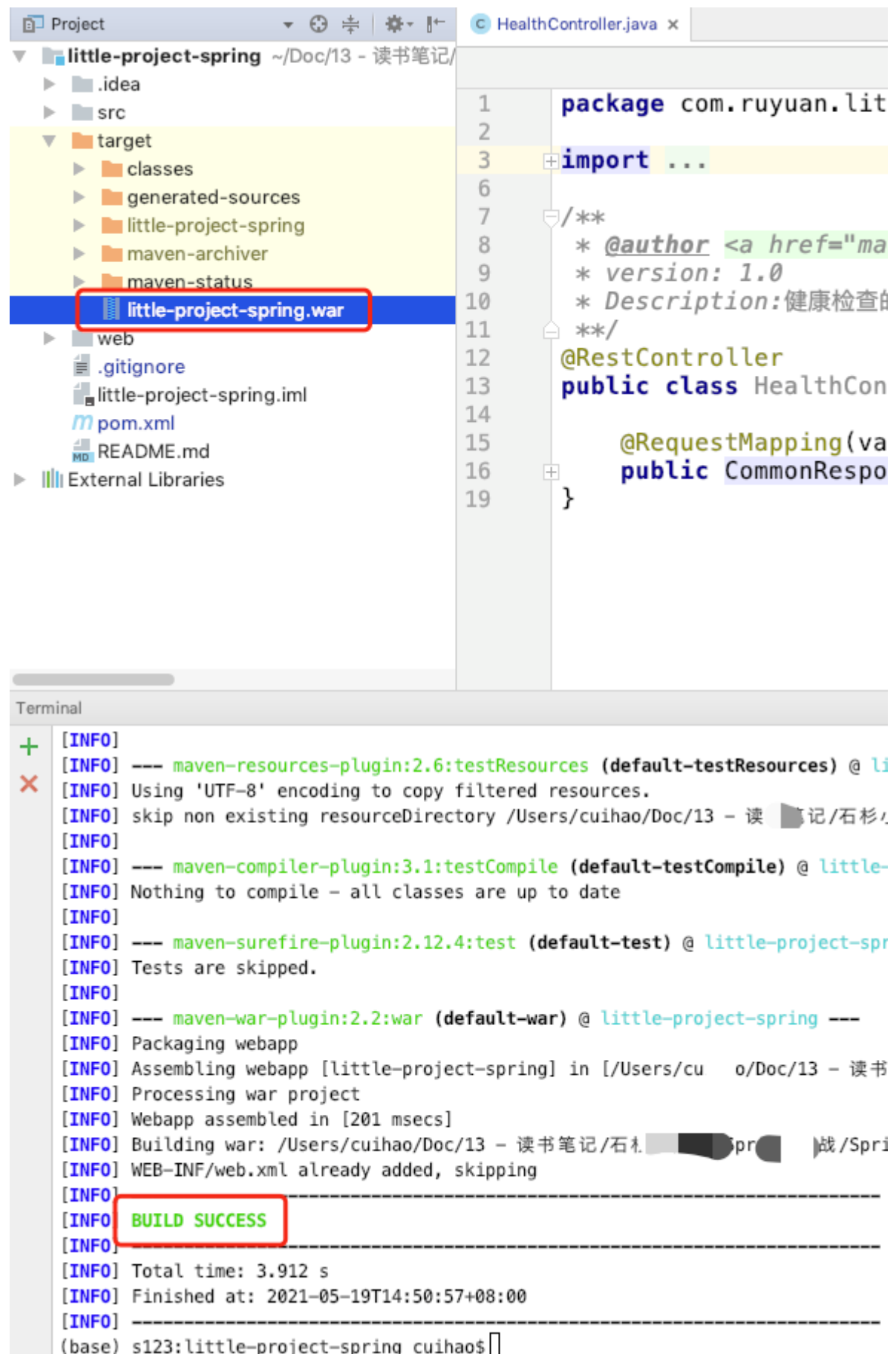


图 2 输入打包命令

在输入命令以后回车，如图 3 所示，会在输出信息中看到“BUILD SUCCESS”的字样表示，打包成功。同时在项目文件的 **target** 目录下面会看到一个“little-project-spring.war”的文件，这就是我们将要发布的 **war** 包。后面的操作会用到它。



The screenshot displays an IDE interface with three main sections:

- Project Explorer (Left):** Shows the project structure for 'little-project-spring'. The 'target' directory is expanded, and 'little-project-spring.war' is highlighted with a red box.
- Code Editor (Right):** Displays the 'HealthController.java' file. The code includes package declarations, imports, and annotations like '@RestController' and '@RequestMapping'. Line 3, 'import ...', is highlighted with a yellow background.
- Terminal (Bottom):** Shows the output of the Maven build process. The output includes information about the build phases (resources, compile, test, war) and the final result: 'BUILD SUCCESS'. The text 'BUILD SUCCESS' is highlighted with a red box.

```
package com.ruyuan.lit

import ...

/**
 * @author <a href="ma
 * version: 1.0
 * Description:健康检查的
 */
@RestController
public class HealthCon

@RequestMapping(va
public CommonRespo
}
```

```
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ li
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/cuihao/Doc/13 - 读
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ little-
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ little-project-spr
[INFO] Tests are skipped.
[INFO] --- maven-war-plugin:2.2:war (default-war) @ little-project-spring ---
[INFO] Packaging webapp
[INFO] Assembling webapp [little-project-spring] in [/Users/cu   o/Doc/13 - 读书
[INFO] Processing war project
[INFO] Webapp assembled in [201 msecs]
[INFO] Building war: /Users/cuihao/Doc/13 - 读书笔记/石杉
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.912 s
[INFO] Finished at: 2021-05-19T14:50:57+08:00
[INFO] -----
(base) s123:little-project-spring cuihao$
```

图 3 打包成功

3、上传服务

在上传服务（jar 包）之前需要保证，在对应的 ECS 中建立服务运行的目录如下：

```
/home/admin/little-project-spring/
```

这个目录已经由儒猿团队在 ECS 的镜像中创建好了，大家不用手动创建，我们后面的发布就基于这个目录。其中“/home/admin/little-project-spring”是用来存放部署脚本的，注意我们这里使用的 `deploy.sh` 脚本已经由儒猿团队上传了。

“/home/admin/little-project-spring/”目录是存放服务的 war 包的，这个包是需要我们自己上传的。

在 ECS 上建立好目录结构以后，再回到本地的项目中，依旧是在 IntelliJ IDEA 的命令行中输入以下命令：

```
scp target/little-project-spring.war root@47.117.120.102:/home/admin/little-project-spring /
```

命令的意思是通过 `scp` 命令将刚才打包的“little-project-spring.war”文件 copy 到对应 ECS 服务器的“/home/admin/little-project-spring/”目录中。这里的

“47.117.120.102”是我的测试地址，大家可以更换为自己申请的 ECS 的 IP 地址。

这里需要特别说明一下，由于我在执行命令的根目录在“little-project-spring”项目下面，如果你在其他的地方执行上述两条命令，需要指定好源文件的目录。同时在使用 `scp` 命令以后会让大家输入服务器的密码，该密码可以从实战云平台上获取。完成上面两个命令以后，服务就已经部署到 ECS 了。

4、启动服务

完成部署以后，需要到 ECS 服务器上面启动部署的酒店管理后台的服务。还是登录 ECS 服务器，通过以下命令进入到“`deploy.sh`”文件所在的目录。

```
cd /home/admin/little-project-spring
```

`deploy.sh` 文件是儒猿团队为大家生成的发布的脚本文件，通过 Linux shell 脚本完成发布的参数配置和启动命令。包括启动应用等待的时间、应用端口号、健康检查

的 URL 以及 jar 包的目录和日志信息。有兴趣的同学可以打开看看，这里就不做展开的介绍了。

保证当前目录下面存在“deploy.sh”文件，使用如下命令启动服务。

```
sh deploy.sh restart
```

命令使用了“restart”作用与“start”是一致的，用“restart”的目的以免在重复发布过程中，学员忘记是否启动过服务。因此使用“restart”，这样即便是已经启动过服务，也会重新加载服务。

运行命令在看到图 4 所示的“success”字样的时候，就说明服务启动成功了。

```
[root@iZuf64qotsm6gzy9jnkW8cZ little-project-spring]# sh deploy.sh restart
no java process
starting java process
Using CATALINA_BASE:   /home/admin/little-project-spring/apache-tomcat-9.0.45
Using CATALINA_HOME:   /home/admin/little-project-spring/apache-tomcat-9.0.45
Using CATALINA_TMPDIR: /home/admin/little-project-spring/apache-tomcat-9.0.45/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/admin/little-project-spring/apache-tomcat-9.0.45/bin/bootstrap.jar:/home/admin/little-project-spring/apache-tomcat-9.0.45/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
started java process
checking http://127.0.0.1:8090
Wait app to pass health check: 1...
code is 200
check http://127.0.0.1:8090 success
```

图 4 启动服务成功

5、测试支付和完成订单功能

打包部署完毕以后就开始今天的代码测试了，主要有两个场景需要测试，第一个是主动完成订单，另外一个延迟完成订单。

先从主动完成订单开始，如图 5 所示，选择王五老师的课程，点击“购买”生成对应的订单。

14:44



教师详情



王五

英语

专注于6-10岁儿童教育

销量 55 评分 4.41

价格: **¥190**

教学天数: 5

详细: 常年做儿童教育, 经验丰富

购买

图 5 教师详情

接下来就是确认订单的信息了，如图 6 所示，在确认订单页面中填入家教开始和结束时间，如果有可能也可以填入积分和优惠券的信息。然后，点击“立即付款”。

14:45

确认订单

王五

英语

销量 55 评分 4.41

专注于6-10岁儿童教育

价格 **¥190/天**

教学天数：5

详细：常年做儿童教育，经验丰富

家教开始时间

2021-06-11

▼

家教结束时间

2021-06-13

▼

家教间隔天数

3

手机号

15827218376

使用积分

优惠券

▼

合计： **¥ 570.00**

立即付款

图 6 确认订单

如图 7 所示，确认订单以后就可以看到订单详情页面，点击下方的“家教完成”按钮来完成订单。



图 7 家教完成

执行完上述操作以后去 ECS 上面通过日志看看消息的发送情况。运行如下命令查看日志信息，`cat /home/admin/little-project-spring/logs/little-project-spring.log`。

如图 8 所示，在点击“家教完成”的时候，小程序会调用 `OrderController` 中的 `finishOrder` 方法完成订单，同时会发送订单完成的消息。

```
x":0,"status":2,"statusName":"已支付","teacherId":3,"teacherName":"王五","totalPrices":570.00,"message":"成功"}
[2021-06-11 14:45:48.643] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 57 - 请求处理方法:com.ruyuan.little.project.spring.controller.OrderController.f
inishOrder
[2021-06-11 14:45:48.643] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["orderId"], 请求参数值:[2]
[2021-06-11 14:45:48.676] INFO com.ruyuan.little.project.spring.service.impl.OrderEventManagerImpl 90 - send order message finished messageTypeEnum:WX_FINISHED_ORDER, orde
rId:2
[2021-06-11 14:45:48.680] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 69 - 请求处理时间差:37, 响应结果:{"code":200,"data":1,"message":"成功"}
[2021-06-11 14:45:48.826] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 57 - 请求处理方法:com.ruyuan.little.project.spring.controller.OrderController.f
indById
```

图 8 发送订单完成消息

看完了消息发送方，再看看消息接收方，依旧是在日志中找到消息接受的部分。如图 9 所示，可以看到 `received order message` 的字样，这里就是接受到订单完成的消息，并且对应进行积分增加、优惠券发放的操作。

```
[2021-06-11 14:45:49.988] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 49 - received order message:{"content":{"betweenDay":3,"consumerId":14876,"
consumerName":"微信用户","counted":0,"couponAmount":0.00,"course":"02","deductCredits":0,"endDate":"2021-06-13 23:59:59","endIndex":10,"id":2,"pa
ge":1,"payAmount":570.00,"payChannel":1,"payTime":"2021-06-11 14:45:39","priceOfDay":190.00,"receiveCredits":570,"size":10,"startDate":"2021-06-11 00:
00:00","startIndex":0,"status":3,"teacherId":3,"teacherName":"王五","totalPrices":570.00,"messageType":"WX_FINISHED_ORDER"}
[2021-06-11 14:45:49.997] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 69 - 订单完成，积分增加
[2021-06-11 14:45:50.008] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 74 - 订单完成，优惠券发放
[2021-06-11 14:45:50.008] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 79 - 订单消息处理完成
[2021-06-11 14:45:51.076] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 57 - 请求处理方法:com.ruyuan.little.project.spring.controller.OrderController.f
indById
```

图 9 接受订单完成消息

上面描述的是主动完成订单的操作，接下来就来完成超时完成订单的操作，这里我们支付订单以后，不主动点击“家教完成”按钮，尝试等待一段时间（和取消订单一样设置为 5 分钟），看看订单是否会因为延迟消息而自动完成。

如图 10 所示，选择钱老师的课程，进入“教师详情”页面。

15:03



教师详情



钱老师

化学

专注于6-10岁儿童教育

销量 73 评分 4.98

价格: **¥ 205**

教学天数: 2

详细: 常年做儿童教育, 经验丰富

购买

图 10 教师详情

如图 11 所示，确认家教开始和结束时间，以及相关信息，然后点击“立即付款”。

15:03

确认订单

钱老师

化学

销量 73 评分 4.98

专注于6-10岁儿童教育

价格 **¥205/天**

教学天数：2

详细：常年做儿童教育，经验丰富

家教开始时间

2021-06-11

▼

家教结束时间

2021-06-13

▼

家教间隔天数

3

手机号

15827218376

使用积分

优惠券

首次登录优惠券

▼

合计： **¥595.00**

立即付款

图 11 确认订单

如图 12 所示，付款以后进入“订单详情”页面，此时不点击“家教完成”的按钮，只是等待 5 分钟，然后去 ECS 上面查看日志。

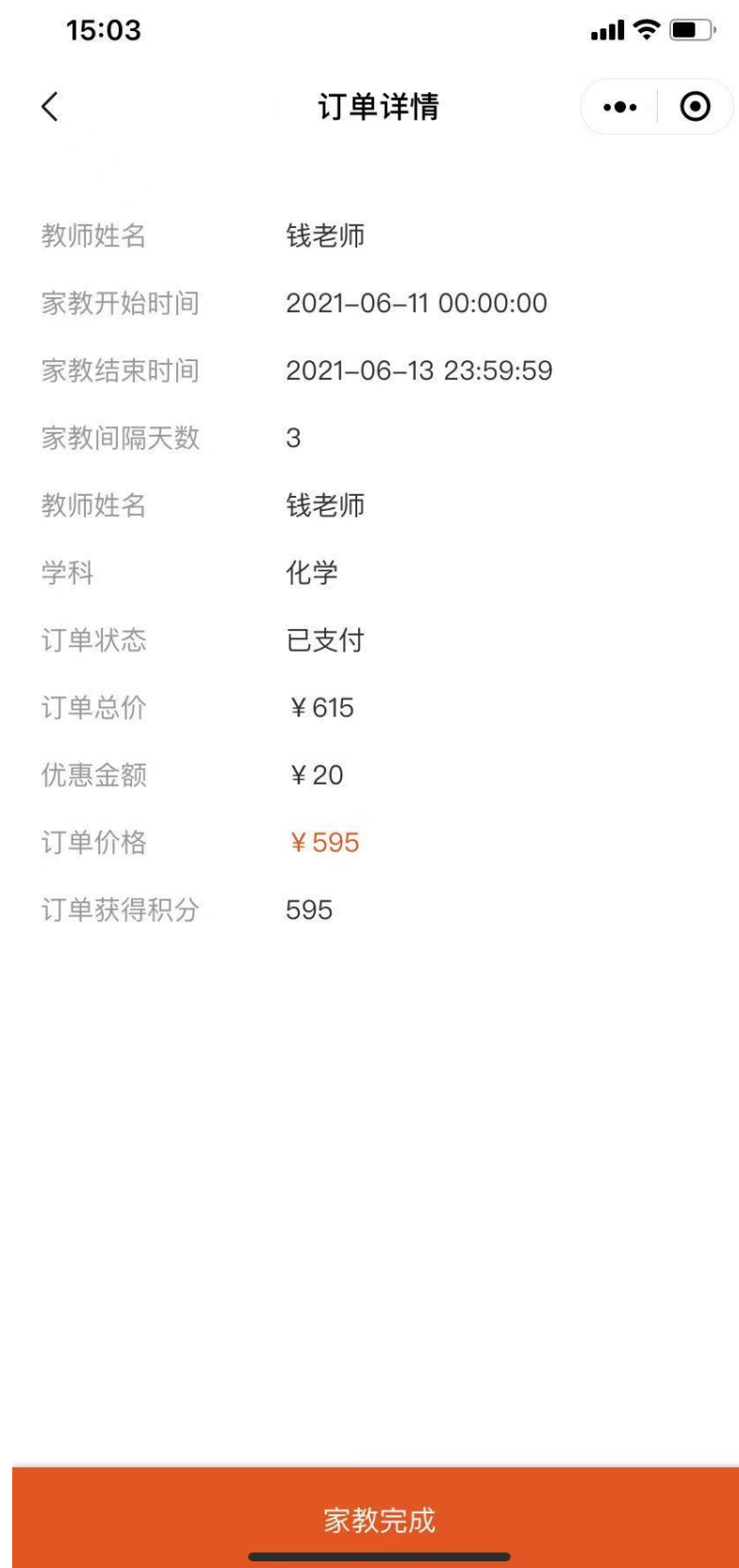


图 12 订单详情

依旧是通过日志查看消息的发送与接收是否正确。如图 13 所示，在 ECS 上的日志中我们发现，当支付订单的时候发送了“send order delay pay/finish message finished orderId: 3”的消息。此后，监听器在 5 分钟后接收到这个消息，紧接着完成后续操作，包括：积分增加、优惠券发放。

```

payOrder
[2021-06-11 15:03:40.317] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["orderId"], 请求参数值:[3]
[2021-06-11 15:03:40.333] INFO com.ruyuan.little.project.spring.service.impl.OrderEventManagerImpl 117 - send order delay pay/finish message finished orderId:3 delayTimeLe
val:9 delayMessageType:完成延时
[2021-06-11 15:03:40.337] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 69 - 请求处理时间差:20, 响应结果:{code:200,data:{code:1,message:成功}}
[2021-06-11 15:03:40.497] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 57 - 请求处理方法:com.ruyuan.little.project.spring.controller.OrderController.f
indById
[2021-06-11 15:03:40.497] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["id"], 请求参数值:[3]
[2021-06-11 15:03:40.502] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 69 - 请求处理时间差:4, 响应结果:{code:200,data:{betweenDay:3,consumerCou
ponId:1,consumerId:14876,consumerName:微信用户,counted:0,couponAmount:20.00,course:04,courseName:化学,deductCredits:0,endDate:2021-06-13 23:59:59,
,endIndex:10,id:3,page:1,payAmount:595.00,payChannel:1,payTime:2021-06-11 15:03:40,priceOfDay:205.00,receiveCredits:595,size:10,startDate:2021-06-1
1 00:00:00,startIndex:0,status:2,statusName:已支付,teacherId:5,teacherName:钱老师,totalPrices:615.00,message:成功}}
[2021-06-11 15:08:37.601] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 58 - 延时消息队列消息总数1
[2021-06-11 15:08:37.601] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 60 - received order delay message:{content:{betweenDay:3,consumerCo
uponId:1,consumerId:14876,consumerName:微信用户,couponAmount:20,course:04,deductCredits:0,endDate:2021-06-13 23:59:59,endIndex:10,id:3,page:1,payAmount:595.00,payChannel:1,priceOfDay:205,receiveCredits:595,size:10,startDate:2021-06-11,startIndex:0,status:2,statusName:待付款,teacherId:5,teacherName:钱老师,totalPrices:615.00,delayMessageType:PAY_DELAY}}
[2021-06-11 15:08:37.601] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 68 - 订单支付超时, 现在开始取消订单, 订单信息Order{id=3, consumerId=14876, c
onsumerName=微信用户, teacherId=5, teacherName=钱老师, course=04, status=0, statusName=待付款, priceOfDay=205, startDate=2021-06-11, endDate=2021-06-13, creat
eTime=null, payTime=null, cancelTime=null, finishTime=null, betweenDay=3, totalPrices=615.00, deductCredits=0, consumerCouponId=1, couponAmount=20, creditsA
mount=null, payAmount=595.00, payChannel=1, receiveCredits=595, receiveCouponId=null, commentId=null, comment=null}
[2021-06-11 15:08:37.608] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 114 - 订单已经取消, 无需再取消
[2021-06-11 15:08:40.330] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 58 - 延时消息队列消息总数1
[2021-06-11 15:08:40.330] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 60 - received order delay message:{content:{betweenDay:3,consumerCo
uponId:1,consumerId:14876,consumerName:微信用户,counted:0,couponAmount:20.00,course:04,deductCredits:0,endDate:2021-06-13 23:59:59,endIndex:10,id:3,page:1,payAmount:595.00,payChannel:1,priceOfDay:205.00,receiveCredits:595,size:10,startDate:2021-06-11,startIndex:0,status:0,teacherId:5,teacherName:钱老师,totalPrices:615.00,delayMessageType:FINISH_DELAY}}
[2021-06-11 15:08:40.339] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 100 - 订单完成, 积分增加
[2021-06-11 15:08:40.344] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 105 - 订单完成, 优惠券发放
[2021-06-11 15:08:40.345] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 111 - 订单延时消息处理完成

```

图 13 发送接收订单完成延迟消息

6、总结

本节课带大家把订单完成的两种场景的代码测试了一下，分别是主动完成订单和超时完成订单。作为本周的最后一节课这里将一周的课程做一个总结，本周从支付订单、完成订单以及评论订单的业务逻辑入手，整理出整个订单流程的业务逻辑图。先通过同步的方式实现以上逻辑，然后发现该方式会引起系统性能问题，因此提出使用 RocketMQ 的方式将支付订单和完成订单进行异步处理，并且付诸于实践。下周课程将围绕教师评分功能展开，会使用到 Spring 任务调度的功能。下期见，拜拜。