

78_阶段测试：验证一下系统预订超时订单是否自动取消？

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

上节课介绍了主动取消订单和延迟取消订单的代码实施，其中包括消息队列的配置：pom 文件和 application.properties 文件；消息队列的服务：消息生产者、消息消费者、消息监听器；消息队列的应用：OrderServiceImpl 和 OrderEventManagerImpl。本节课会带大家进行取消订单业务代码的测试。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
- 测试取消订单功能

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。

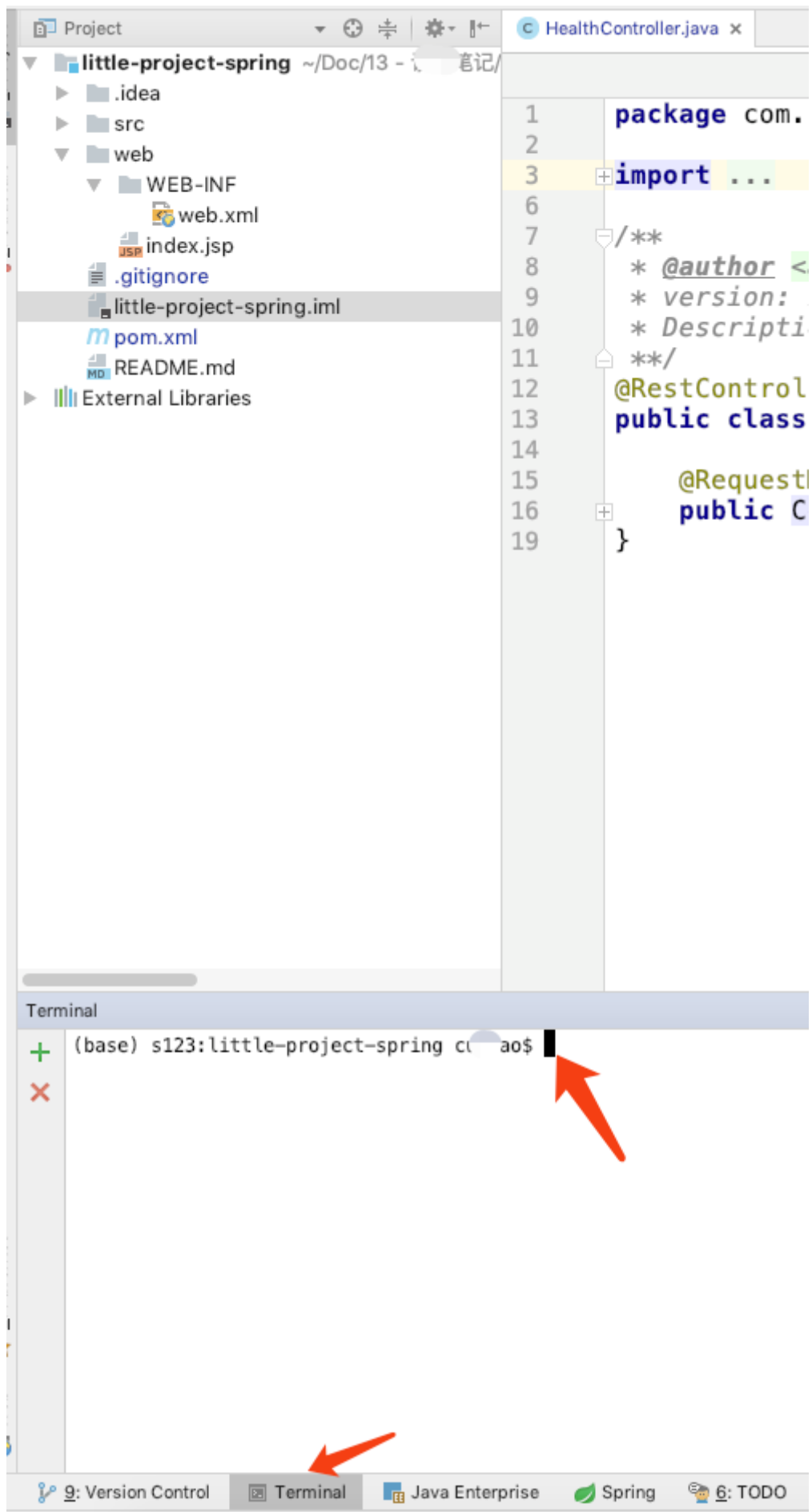


图 1 打开命令行

如图 2 所示，这里输入“mvn clean package”的命令。

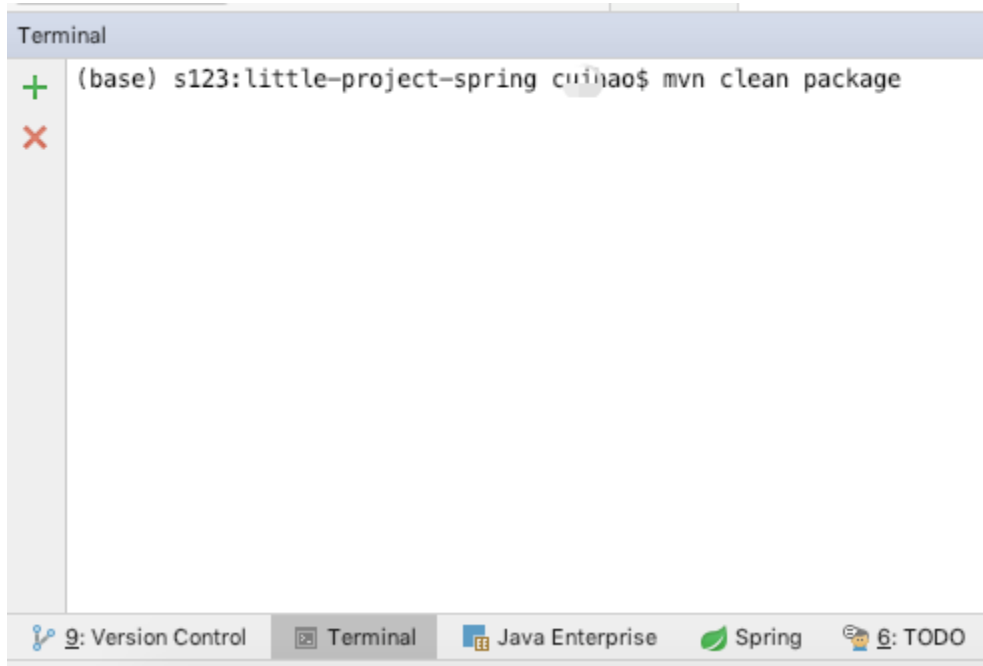
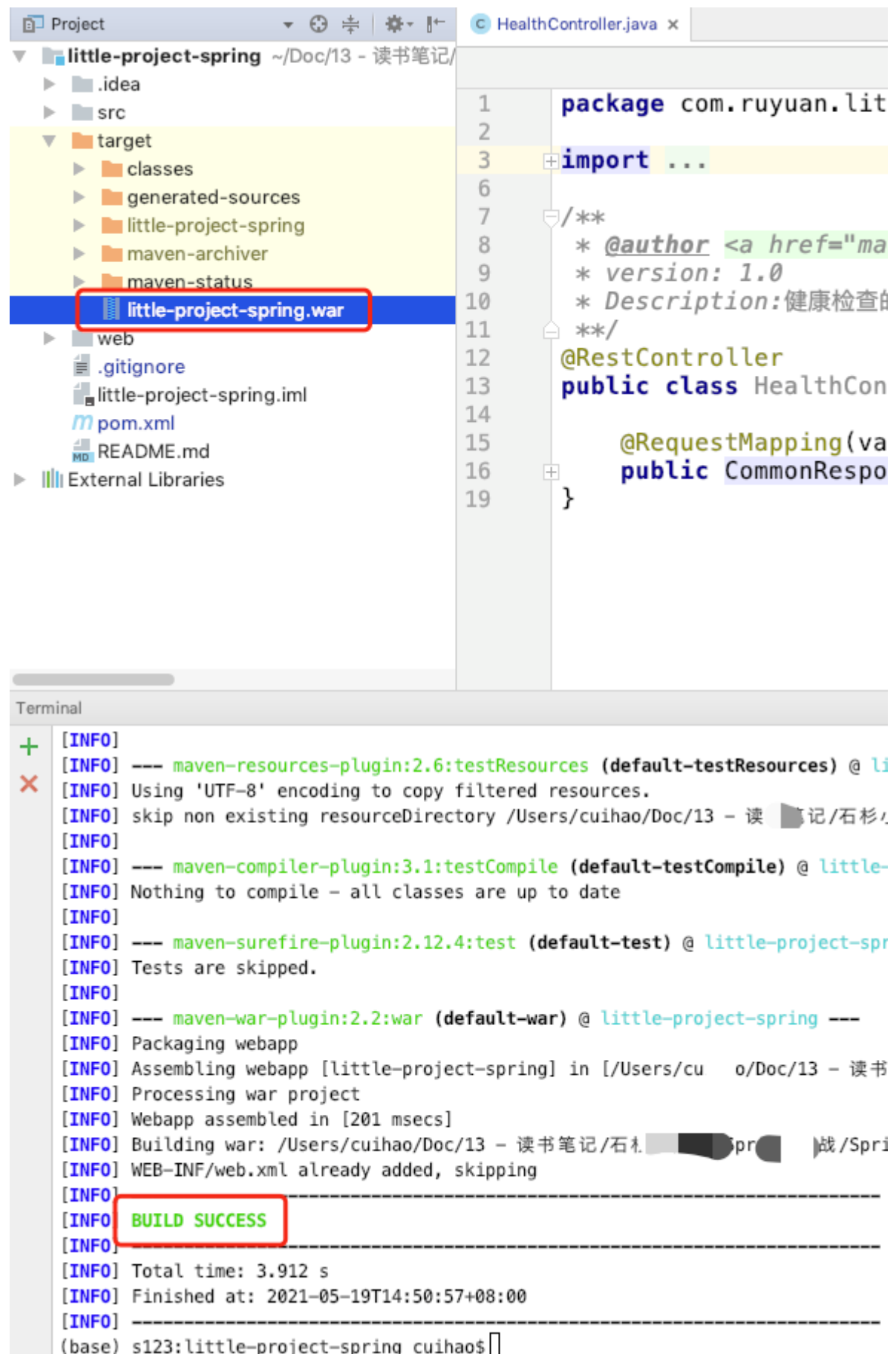


图 2 输入打包命令

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 'BUILD SUCCESS' text 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.测试取消订单功能

经过打包与部署之后来测试一下创建订单的功能，登陆微信小程序，如图 5 所示，在“我的优惠券”菜单查看“首次登陆优惠券”。

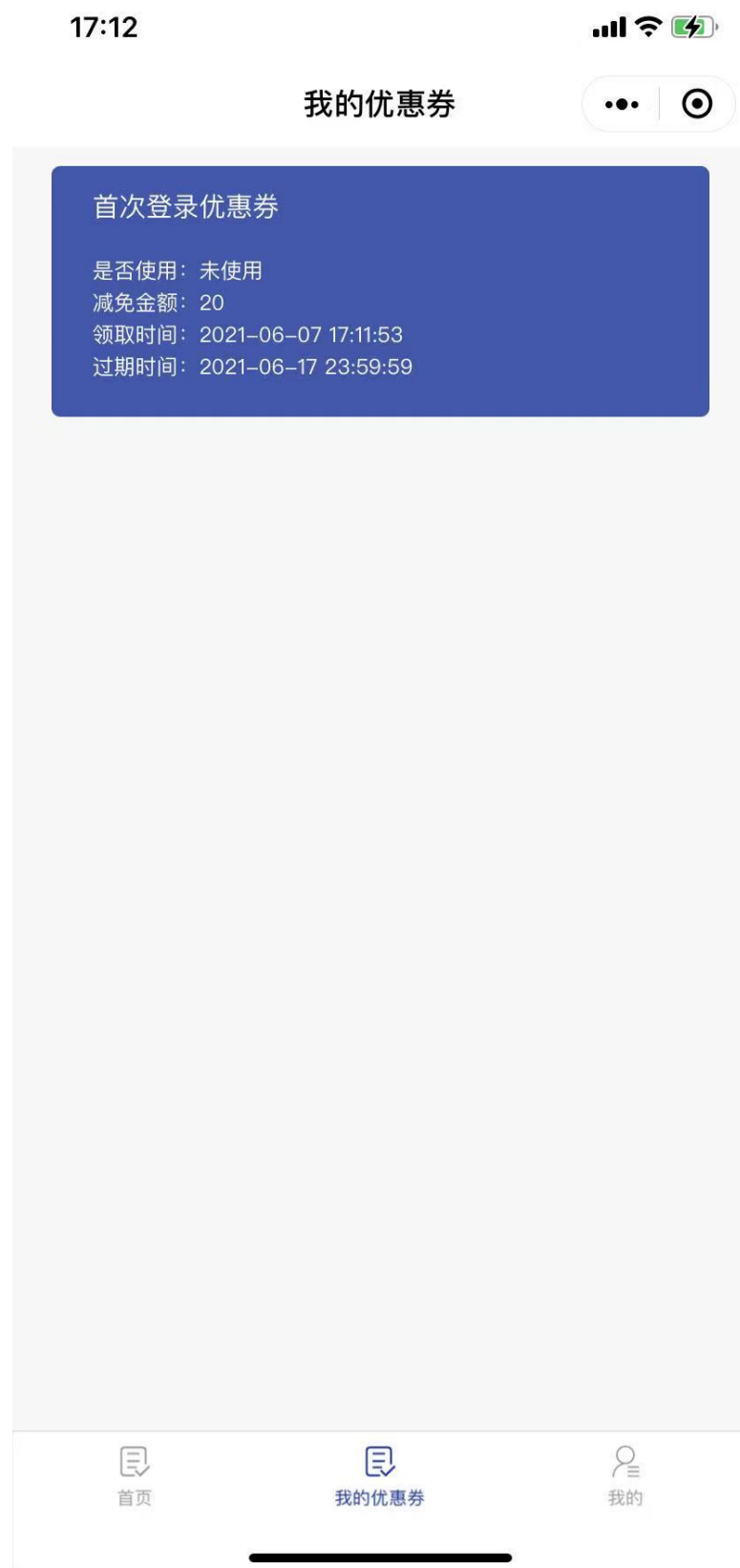


图 5 查看登陆优惠券

需要注意的是，如果你的账号登陆过多次了，这里无法收到优惠券，就需要修改数据库并且重新登陆获取这个优惠券。

在登陆 ECS 以后输入以下代码段用来登陆 MySQL 数据库。

```
mysql -uroot -p
```

输入登陆 MySQL 的登陆命令以后会弹出“Enter Password”的字样，随后输入“root”回车，我们给大家分配到 ECS 上的 MySQL 数据库的密码是“root”。

通过如下 SQL 语句重置首次登录：`update spring.t_consumer set login_times = 0 where phone = '手机号'`。这样就可以看到优惠券信息。

由于刚开始没有消费课程所以积分为 0，因此通过如下 SQL 语句添加积分信息：

```
update spring.t_consumer set credits=100 where phone='手机号'
```

这里我们添加 100 积分。

之后来到“我的”页面在个人中心里面查看积分信息，如图 6 所示，积分为 100。



图 6 积分信息

准备好优惠券和积分以后，就可以创建订单操作了，在“首页”的教师列表中选择教师。如图 7 所示，选择了教师“王五”，展开教师的详细介绍。

17:12



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 7 显示教师详细信息

在展开详细信息中点击“购买”，如图 8 所示，进入“确定订单”页面。在页面中选择家教开始时间为：2021-06-07，家教结束时间为：2021-06-09。由于我们有 100 积分，按照规则可以抵扣 100 元钱，因此在“使用积分”这一栏输入 100。在“优惠券”处下拉框中选择“首次登陆优惠券”。然后点击“立即付款”按钮。

17:12



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	2021-06-07	∨
家教结束时间	2021-06-09	∨
家教间隔天数	3	
手机号	15827218376	
使用积分	100	
优惠券	首次登录优惠券	∨

合计: **¥450.00**

立即付款

图 8 确定订单

点击“立即付款”以后，显示订单详情页面。如图 9 所示，我们选择了三天的课程，每节课是 190 元，订单总价是 570，另外使用了 20 元的首次登陆优惠券和 100 积分（抵扣 100 元），因此优惠金额为 120。 $570-120=450$ ，最终的订单价格为 450。



图 9 展示订单详情

经过以上步骤就完成订单的创建。

接下来演示主动取消订单的操作，在“我的”菜单中选择“全部订单”，在展开的订单列表中选择刚才下单的项目，点击对应的“取消订单”的按钮。如图 10 所示，会显示订单的状态变为“已取消”。

19:19



订单列表



教师名称：王五

学科：英语

价格/天：190

间隔天数：3

上课时间：2021-06-07 00:00:00-2021-06-09 23:59:59

总价 ¥ 450

优惠金额 ¥ 120

已取消

图 10 订单已经取消

取消订单以后回到 ECS 上查看日志。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 11 所示在取消订单的时候，订单消息的生产者会发送一定取消订单的消息，取消订单的监听器会接受该消息并且进行处理，具体处理的结果包括积分和优惠券的还原。

```
[2021-06-09 19:15:14.270] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["orderId"], 请求参数值:[1]
[2021-06-09 19:15:14.298] INFO com.ruyuan.little.project.spring.service.impl.OrderEventManagerImpl 81 - send order message finished messageTypeEnum:WX_CANCEL_ORDER orderId:1
[2021-06-09 19:15:14.301] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 69 - 请求处理时间差:31, 响应结果:{"code":200,"data":1,"message":"成功"}
[2021-06-09 19:15:14.399] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 57 - 请求处理方法:com.ruyuan.little.project.spring.controller.OrderController.getPage
[2021-06-09 19:15:14.399] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["order"], 请求参数值:[{"consumerId":14876,"deductCredits":0,"endIndex":10,"page":1,"receiveCredits":0,"size":10,"startIndex":0}]
[2021-06-09 19:15:14.410] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 69 - 请求处理时间差:10, 响应结果:{"code":200,"data":{"rows":[{"betweenDay":3,"cancelTime":"2021-06-09 19:15:14","consumerCouponId":1,"consumerId":14876,"consumerName":"微信用户","counted":0,"couponAmount":20.00,"course":"02","courseName":"英语","creditsAmount":100.00,"deductCredits":100,"endDate":"2021-06-09 23:59:59","endIndex":10,"id":1,"page":1,"payAmount":450.00,"payChannel":1,"priceOfDay":190.00,"receiveCredits":450,"size":10,"startDate":"2021-06-07 00:00:00","startIndex":0,"status":1,"statusName":"已取消","teacherId":3,"teacherName":"王五","totalPrices":570.00}], "total":1,"message":"成功"}
[2021-06-09 19:15:16.759] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 46 - received order message:{"content":{"betweenDay":3,"consumerCouponId":1,"consumerId":14876,"consumerName":"微信用户","counted":0,"couponAmount":20.00,"course":"02","creditsAmount":100.00,"deductCredits":100,"endDate":"2021-06-09 23:59:59","endIndex":10,"id":1,"page":1,"payAmount":450.00,"payChannel":1,"priceOfDay":190.00,"receiveCredits":450,"size":10,"startDate":"2021-06-07 00:00:00","startIndex":0,"status":1,"teacherId":3,"teacherName":"王五","totalPrices":570.00},"messageType":"WX_CANCEL_ORDER"}
[2021-06-09 19:15:16.766] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 56 - 订单取消, 积分还原
[2021-06-09 19:15:16.771] INFO com.ruyuan.little.project.spring.jms.OrderMessageListener 61 - 订单取消, 优惠券还原
```

图 11 主动取消订单日志

看完了主动取消订单的操作，再来看看超时取消订单的操作，为了方便演示我们将订单延迟的等级设置为 9，也就是 5 分钟之后如果没有支付，系统会自动取消订单。其原理就是利用 Rocket MQ 的延迟消息功能。

按照主动取消订单的套路还是创建一个订单，这里创建的过程就不再赘述了。如图 12 所示，在“我的”-“全部订单”中就可以看到订单列表了。其中我们可以看到一个订单状态为“去付款”的，这个就是我们要处理的延迟取消订单。

19:27



订单列表



教师名称：王五

学科：英语

价格/天：190

间隔天数：3

上课时间：2021-06-07 00:00:00-2021-06-09 23:59:59

总价 ¥ 450

优惠金额 ¥ 120

已取消

教师名称：王五

取消订单

学科：英语

价格/天：190

间隔天数：3

上课时间：2021-06-07 00:00:00-2021-06-09 23:59:59

总价 ¥ 450

优惠金额 ¥ 120

去付款

图 12 “去付款”状态的订单

在创建订单以后去查看日志会发现，如图 13 所示，下单以后系统会发送一个支付延时的消息。

```
[2021-06-09 19:26:49.140] INFO com.ruyuan.little.project.spring.aspect.ControllerLogAspect 60 - 请求参数名:["order"], 请求参数值:[{"betweenDay":3,"consumerCouponId":1,"consumerId":14876,"consumerName":"微信用户","couponAmount":20,"course":"02","creditsAmount":100,"deductCredits":100,"endDate":"2021-06-09","endIndex":10,"page":1,"payAmount":450.00,"payChannel":1,"priceOfDay":190,"receiveCredits":0,"size":10,"startDate":"2021-06-07","startIndex":0,"teacherId":3,"teacherName":"王五","totalPrices":570.00}]
[2021-06-09 19:26:49.157] INFO com.ruyuan.little.project.spring.service.impl.OrderEventManagerImpl 108 - send order delay pay message finished orderId:2 delayTimeLevel:9 delayMessageType:支付延时
```

图 13 发送延迟消息

在 5 分钟以后再次查看日志会发现，监听器接受到了这个延迟消息。如图 14 所示，延时消息队列消息总数 1，在接受到消息以后执行了订单取消的积分和优惠券还原的操作。

```
[2021-06-09 19:31:49.160] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 53 - 延时消息队列消息总数 1
[2021-06-09 19:31:49.160] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 55 - received order delay message:{content":{"betweenDay":3,"consumerCouponId":1,"consumerId":14876,"consumerName":"微信用户","couponAmount":20,"course":"02","courseName":"英语","creditsAmount":100,"deductCredits":100,"endDate":"2021-06-09","endIndex":10,"id":2,"page":1,"payAmount":450.00,"payChannel":1,"priceOfDay":190,"receiveCredits":450,"size":10,"startDate":"2021-06-07","startIndex":0,"status":"0","statusName":"待付款","teacherId":3,"teacherName":"王五","totalPrices":570.00},"delayMessageType":"PAY_DELAY"}
[2021-06-09 19:31:49.161] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 64 - 订单支付超时，现在开始取消订单，订单信息 Order{id=2, consumerId=14876, consumerName='微信用户', teacherId=3, teacherName='王五', course='02', status='0', statusName='待付款', priceOfDay=190, startDate='2021-06-07', endDate='2021-06-09', createTime='null', payTime='null', cancelTime='null', finishTime='null', betweenDay='3', totalPrices='570.00', deductCredits=100, consumerCouponId='1', couponAmount='20', creditsAmount='100', payAmount='450.00', payChannel='1', receiveCredits='450', receiveCouponId='null', commentId='null', comment=null}
[2021-06-09 19:31:49.169] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 76 - 订单取消，积分还原
[2021-06-09 19:31:49.174] INFO com.ruyuan.little.project.spring.jms.OrderDelayMessageListener 81 - 订单取消，优惠券还原
```

图 14 接受延迟消息

如图 15 所示，最后回到小程序中验证订单是否被取消，在“我的”-“全部订单”中查看订单信息，发现刚才“去付款”的订单现在变为“已取消”。

19:44



订单列表



教师名称：王五

学科：英语

价格/天：190

间隔天数：3

上课时间：2021-06-07 00:00:00-2021-06-09 23:59:59

总价 ¥ 450

优惠金额 ¥ 120

已取消

教师名称：王五

学科：英语

价格/天：190

间隔天数：3

上课时间：2021-06-07 00:00:00-2021-06-09 23:59:59

总价 ¥ 450

优惠金额 ¥ 120

已取消

图 15 超时订单被取消

至此就完成了主动取消订单和超时取消订单的操作。

6、总结

本节课介绍了主动取消订单和延迟取消订单的测试过程，在小程序上进行下单、取消、等待操作模拟了两个场景，通过 log 日志的查看正式了我们的想法。本节课是本周的最后一节课，这里给大家做一个总结。本周我们从取消订单的流程说起，引出了大量超时订单如何取消的问题，将传统方法和 RocketMQ 进行了对比发现 RocketMQ 的优点，并在现有的项目中进行实时。最后通过代码测试完成了取消订单，退回优惠券，退回积分的全过程。下周我们会继续订单部分的介绍，专注于订单的支付模块，依旧会提到 MQ 的应用。下期见，拜拜。