

80_阶段测试：提交代码后，验证一下改造后的登录逻辑是否走通

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 替代 Spring Event 发送登陆消息，针对 RocketMQ 的消息生产者、消费者和监听器进行了定义和使用等操作。本节课会测试一下 RocketMQ 改造之后的代码，看看执行效果。下期见，拜拜。

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
- 测试登陆功能

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。

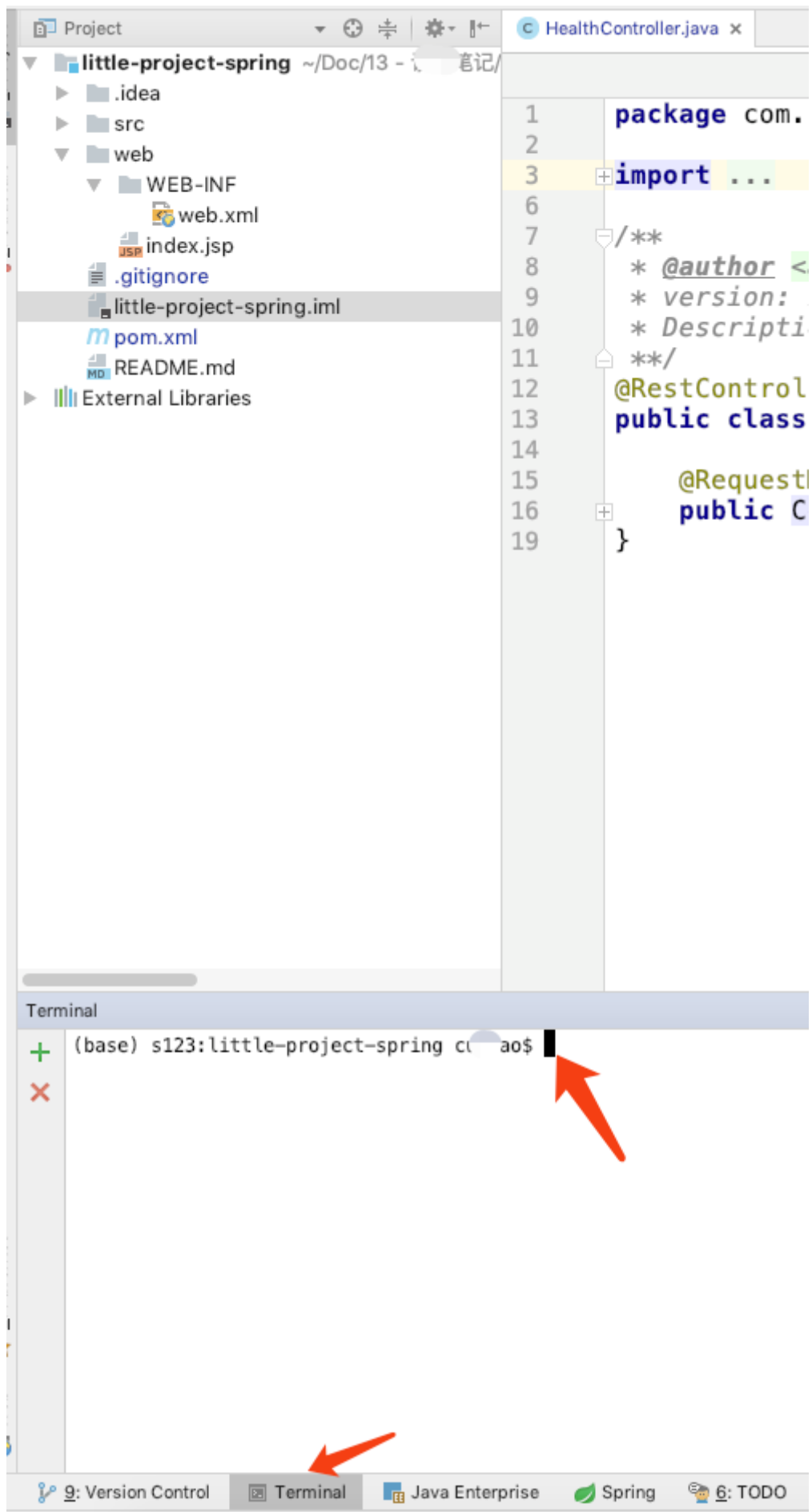


图 1 打开命令行

如图 2 所示，这里输入“mvn clean package”的命令。

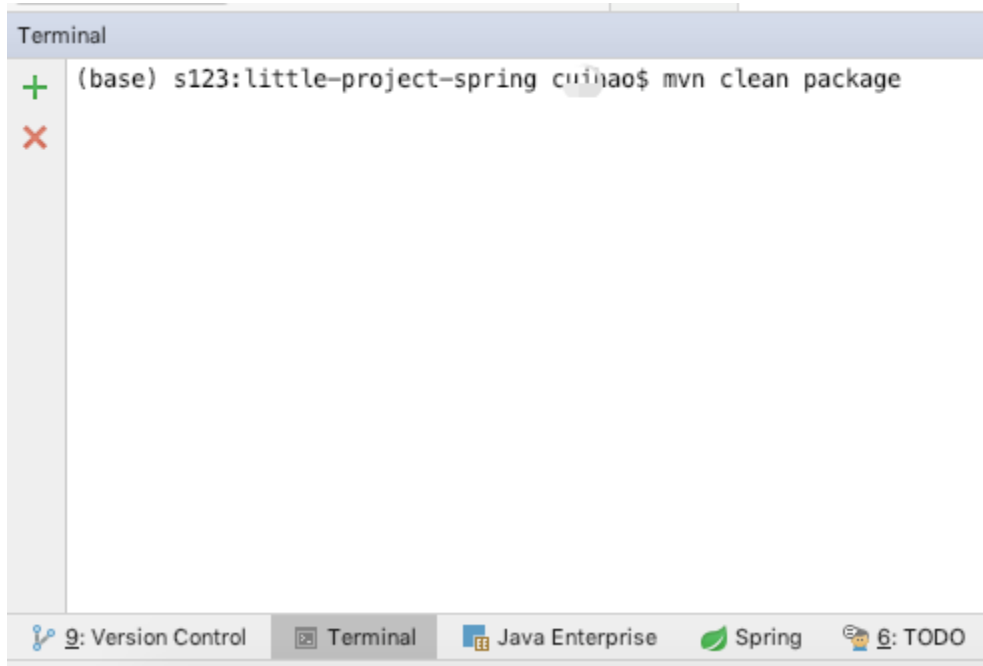


图 2 输入打包命令

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。

Project Explorer:

- little-project-spring ~/Doc/13 - 读书笔记/
 - .idea
 - src
 - target
 - classes
 - generated-sources
 - little-project-spring
 - maven-archiver
 - maven-status
 - little-project-spring.war**
 - web
 - .gitignore
 - little-project-spring.iml
 - pom.xml
 - README.md
- External Libraries

HealthController.java:

```
1 package com.ruyuan.lit
2
3 import ...
4
5
6
7 /**
8  * @author <a href="ma
9  * version: 1.0
10 * Description:健康检查的
11 */
12 @RestController
13 public class HealthCon
14
15     @RequestMapping(va
16     public CommonRespo
17
18
19 }
```

Terminal:

```
[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 记/石杉/
[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/ 战/Spri
[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“started java process”字样的时候，就说明服务启动成功了。但是随后的健康检查服务一直在重试，由于我们在 health 方法里面强行抛出了 exception 导致调用这个方法的时候报错，返回 500 错误。

```
[root@iZuf69caqdf1vzby0x0dlvZ little-project-spring]# sh deploy.sh restart
stop java process
    -- stopping java lasts 59 seconds.
java process has exited

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 500
Wait app to pass health check: 2...
code is 500
Wait app to pass health check: 3...
```

图 4 启动服务成功。

5、测试登陆功能

经过打包与部署之后来测试一下登陆功能是否生效，由于这里我们使用了 RocketMQ 去替代 Spring Event 从用户使用角度没有影响，只是将登陆的实现方式进行了调整。因此，在成功登陆小程序以后，需要到 ECS 上查看对应的日志，从而证明 RocketMQ 执行成功了。

登陆小城以后回到 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 5 所示，在系统启动的时候就会启动消息生产者和消费者的服务。

```
root@izuf6gwwj1uabccctaz092:~# cat little-project-spring.log
[2021-07-17 15:37:36.228] INFO com.ruyuan.little.project.spring.jms.LoginProducerConfiguration 39 - 开始启动消息生产者服务...
[2021-07-17 15:37:36.390] INFO com.ruyuan.little.project.spring.jms.LoginProducerConfiguration 46 - 消息生产者服务启动成功.
[2021-07-17 15:37:36.475] INFO com.ruyuan.little.project.spring.jms.MqConsumerConfiguration 62 - 开始启动消息消费者服务...
[2021-07-17 15:37:36.701] INFO com.ruyuan.little.project.spring.jms.MqConsumerConfiguration 80 - 消息消费者服务启动成功.
```

图 5

如图 6 所示，在登陆完成以后会通过 `loginEventManagerImpl` 发送一条消费者登陆的消息。

```
[2021-07-17 15:38:55.440] INFO com.ruyuan.little.project.spring.service.impl.LoginEventManagerImpl 54 - send login message finished consumer:Consumer{id=14876, consumerName='微信用户', phone='15827218376', credits=0, sex='0', birthday='null', orderTimes=null, openId='null', email='null', address='null', totalPrices=null, loginTimes=null}
```

图 6

紧接着就是消息消费者定义的监听器监听到对应的消息，如图 7 所示，

`LoginMessageListener` 接受到消费者登陆的消息，并进行后续的业务处理。

```
[2021-07-17 15:38:58.053] INFO com.ruyuan.little.project.spring.jms.LoginMessageListener 67 - received login message:{"consumerName":"微信用户","credits":0,"id":14876,"phone":"15827218376","sex":"0"}
[2021-07-17 15:39:00.057] INFO com.ruyuan.little.project.spring.jms.LoginMessageListener 78 - 消费者首次登录，发放优惠券，消费者信息Consumer{id=14876, consumerName='微信用户', phone='15827218376', credits=0, sex='0', birthday='null', orderTimes=null, openId='null', email='null', address='null', totalPrices=null, loginTimes=null}
[2021-07-17 15:39:00.064] INFO com.ruyuan.little.project.spring.jms.LoginMessageListener 111 - 发放优惠券ConsumerCoupon{id=null, consumerId=14876, couponId=1, consumerName='微信用户', couponName='首次登录优惠券', status='0', amount=20.00, createTime='2021-07-17 15:39:00', expireTime='2021-07-27 23:59:59'}
[2021-07-17 15:39:00.084] INFO com.ruyuan.little.project.spring.jms.LoginMessageListener 98 - 登录事件监听完成
```

图 7

6、总结

本节课把 **RocketMQ** 实现的登陆功能进行了测试，通过查看日志的方式将消息发送、接受、处理后续逻辑的步骤展示给大家。这里依旧对一周的课程进行总结，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。