

## 100\_阶段测试：回归测试任务调度执行后，教师教学天数是否更新？

---

**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](http://www.ruyuan2020.com)（建议 PC 端访问）

---

### 1、开篇

上节课介绍了基于 XXL-Job 优化后的教学天数的计算功能，该功能会由 XXL-Job 控制器进行触发，具体任务执行的工作交给了 Spring Batch 完成。本节课我们会将本节课实施的代码进行测试，看看计算教学天数的任务是如何执行的。今天课程的内容包括以下几个部分：

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
- 测试计划任务执行功能

### 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。

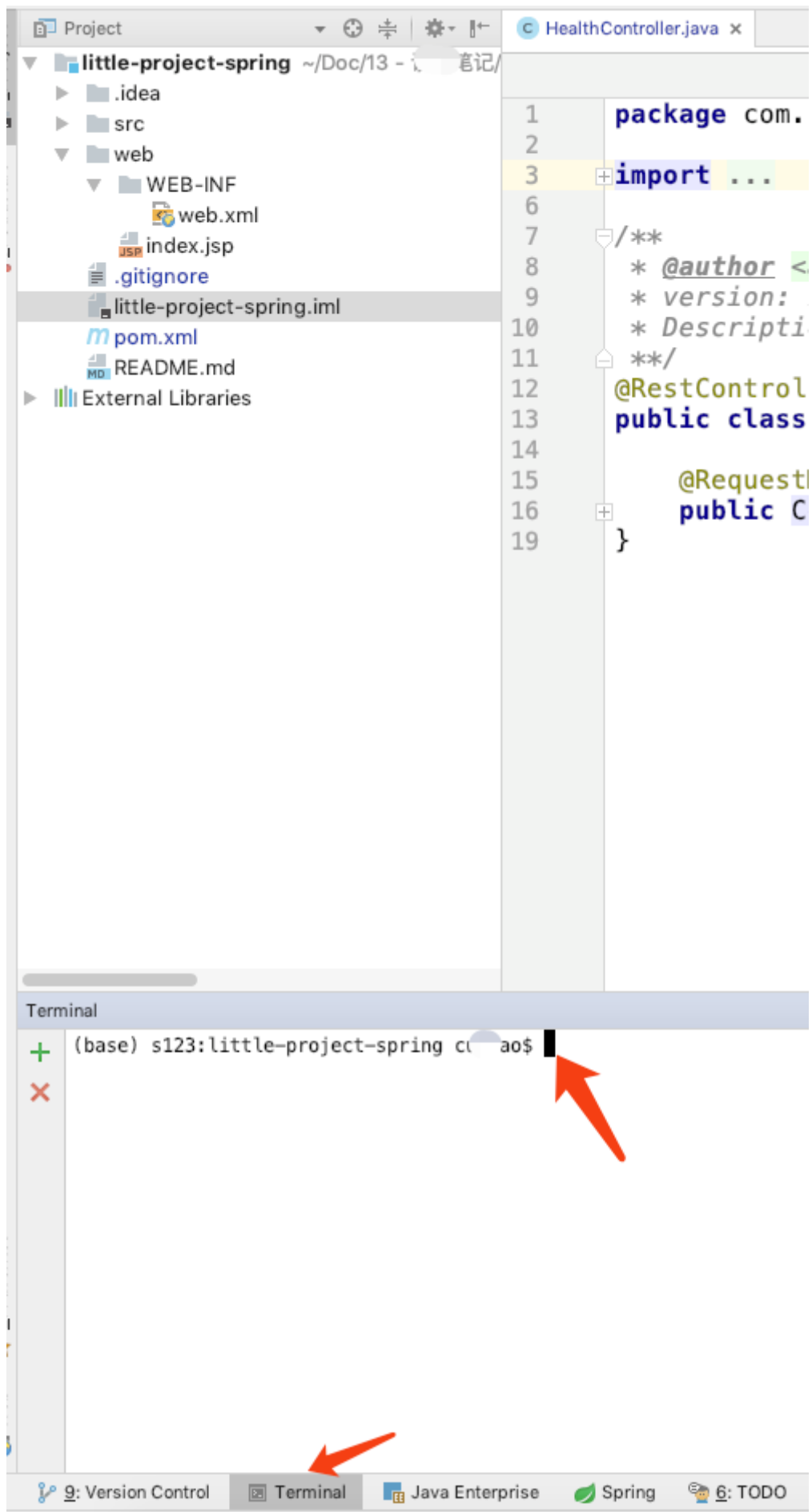


图 1 打开命令行

如图 2 所示，这里输入“mvn clean package”的命令。

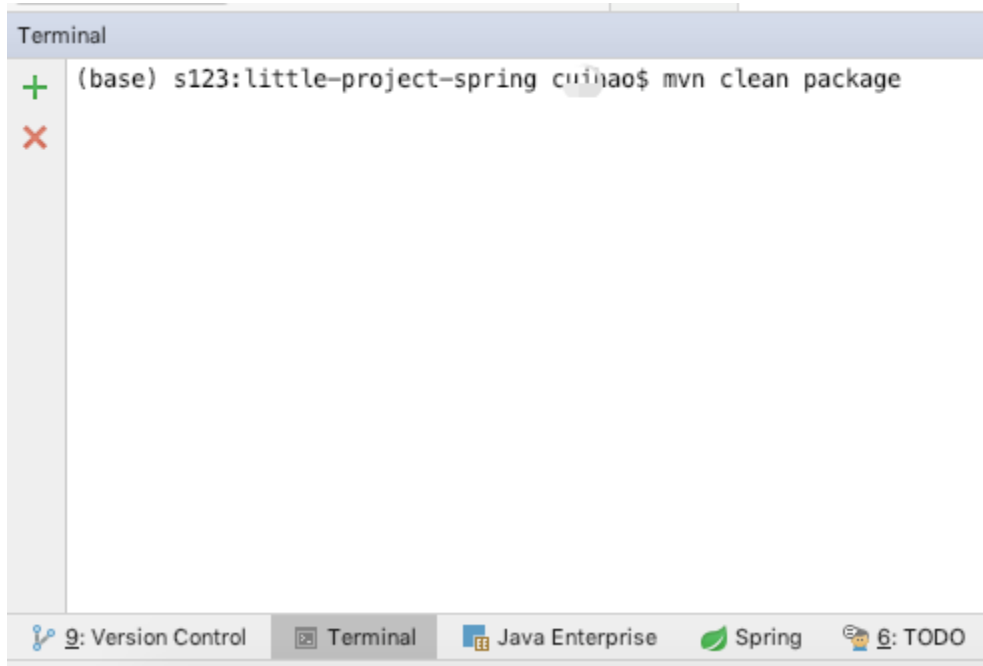


图 2 输入打包命令

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

这里启动服务需要特别说明一下，要先启动 `xxl-job` 的服务，然后再启动我们的项目服务，先启动 `xxl-job` 服务，才能让我们的服务启动时找到对应的 `xxl-job` 服务了。

接下来就是启动 `xxl-job` 的平台服务了，由于 `xxl-job` 的安装部署以及数据库相关设置都由儒猿团队在 ECS 中完成了，这里只需要大家通过命令行启动对应的服务即可。

通过以下命令进入到“xxl-job-deploy.sh”和“deploy.sh”文件所在的目录。

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

然后通过如下命令启动 xxl-job 服务

```
sh xxl-job-deploy.sh start
```

如图 4 所示，输入命令以后出现 started java process 的字样说明 xxl-job 服务已经启动。



```
[root@iZuf62jr8hyu2ppo8eituxZ little-project-spring]# sh xxl-job-deploy.sh start
starting java process
started java process
```

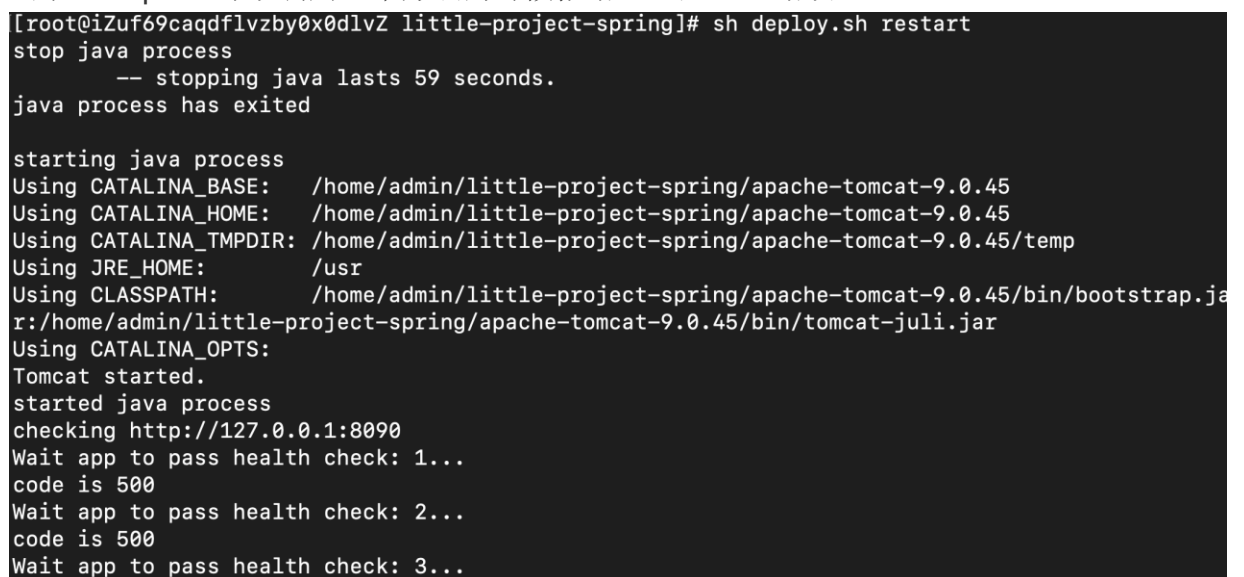
图 4

然后使用如下命令启动项目服务。

```
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 5 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
r:/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 5 启动服务成功。

## 5、测试计划任务执行和缓存过期功能

然后再通过 <http://106.14.175.22:8091/xxl-job-admin> 进入调度中心，对执行器和任务进行配置。注意这里的 URL 需要将 106.14.175.22 换成你所申请的 ECS 服务器的地址。

如图 6 所示，在登陆页面输入账号：admin，密码：123456，并且点击登陆按钮即可登陆。



图 6

登陆以后，从左边的菜单中选择“执行器管理”，看到中间页面显示的执行器中有一个是“xxl-job-executor-spring”的执行器，这个是我们的团队已经为大家设置好的，后面就由这个执行器去执行任务了。



图 7

如图 8 所示，接着在“任务管理”-“新增”页面填写如下信息：

执行器选择“spring 实战执行器”（团队预设好的）

路由策略选择“第一个”

运行模式选择“BEAN”

阻塞处理策略“单机串行”

负责人：XXL

任务描述：更新教师评分信息

Cron：\* \* \* \* \* ? \*

JobHandler: updateTeacherScoreAndTeacherCount，这个值的设置需要和上节课中 XxlJobHandler 类中的 updateTeacherScore 方法上的注释

updateTeacherScoreAndTeacherCount 名称完全相同。也就是通过 XXL-Job 平台来执行这个方法将其变成定时任务。

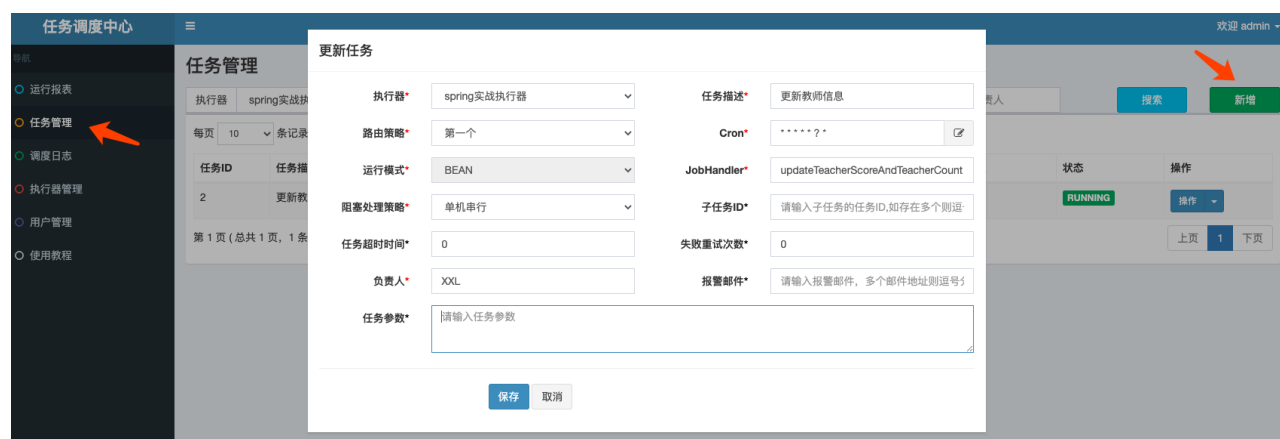


图 8

完成上述操作以后，在选择刚才创建的任务，选择“操作”-“启动”该任务。





图 9

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 10 所示，可以看到 XXL-Job 注册了 `updateTeachingDays` 方法，这个就是 XXL-Job 任务执行的功劳。

```
[2021-07-19 19:27:21.528] INFO com.xxl.job.core.executor.XxlJobExecutor 168 - >>>>>>>>> xxl-job register jobhandler success,
name:expireCouponJobHandler, jobHandler:com.xxl.job.core.handler.impl.MethodJobHandler@2b05fc82[class com.ruyuan.little.project.spring
.xxljob.XxlJobHandler$$EnhancerBySpringCGLIB$$a3d527a2#expireCouponJobHandler]
[2021-07-19 19:27:21.530] INFO com.xxl.job.core.executor.XxlJobExecutor 168 - >>>>>>>>> xxl-job register jobhandler success,
name:updateTeachingDays, jobHandler:com.xxl.job.core.handler.impl.MethodJobHandler@4df88987[class com.ruyuan.little.project.spring
.xxljob.XxlJobHandler$$EnhancerBySpringCGLIB$$a3d527a2#updateTeachingDays]
[2021-07-19 19:27:21.531] INFO com.xxl.job.core.executor.XxlJobExecutor 168 - >>>>>>>>> xxl-job register jobhandler success,
name:updateTeacherScoreAndTeacherCount, jobHandler:com.xxl.job.core.handler.impl.MethodJobHandler@7137dc40[class com.ruyuan.little
.project.spring.xxljob.XxlJobHandler$$EnhancerBySpringCGLIB$$a3d527a2#updateTeacherScore]
[2021-07-19 19:27:21.635] INFO com.xxl.job.core.server.EmbedServer 86 - >>>>>>>>> xxl-job remoting server start success, nettype =
class com.xxl.job.core.server.EmbedServer, port = 9999
```

图 10

通过如下命令查看 `xxl-job` 下面的日志。

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

由于每执行一次任务就会生成一个以 `log` 为后缀名的文件。用 `cat` 命令随便打开一个刚刚执行的 `log` 文件。打开文件如图 11 所示，可以看到开始更新教师教学天数信息，以及对应的完成信息。

```
2021-07-19 19:31:12 [com.xxl.job.core.thread.JobThread#run]-[124]-[Thread-22] <br>----- xxl-job job execute start -----<br>----- Param:
2021-07-19 19:31:12 [com.ruyuan.little.project.spring.xxljob.XxlJobHandler#updateTeachingDays]-[143]-[Thread-22] 开始更新教师教学天数信息
2021-07-19 19:31:12 [com.ruyuan.little.project.spring.xxljob.XxlJobHandler#updateTeachingDays]-[150]-[Thread-22] 更新教师教学天数信息完成
2021-07-19 19:31:12 [com.xxl.job.core.thread.JobThread#run]-[164]-[Thread-22] <br>----- xxl-job job execute end(finish) -----<br>----- ReturnT:ReturnT [code=200, msg=
null, content=null]
```

图 11

至此我们定义的 XXL-Job 调用 `updateTeachingDays` 执行的计划任务的测试就完成了。

## 6、总结

本节课带大家将教学天数更新的任务调度跑了一遍，通过 XXL-Job 调用 Spring batch 定时任务的运行，更新教师的教学天数。作为本周的最后一节课，这里讲本周的课程做一个总结。本周课程从更新教师教学天数的业务逻辑入手，通过 Spring Batch 的技术实现了业务需求，在过程中教授了 Spring Batch 的架构和对应的组件，还使用 XXL-Job 对 Spring Batch 功能进行了改造。

下周的课程会给大家服务整个互联网教学系统，看看我们通过这 14 周的课程学到哪些知识点。下期见，拜拜。