97_代码实战:基于 Spring Batch,编写代码实现教师教学天数的批量更新

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 (建议 PC 端访问)

1、开篇

上节课首先讲了 Spring Batch 的架构体系,包括 JobRepository、JobLauncher、Job、Step(ItemReader、ItemProcessor、ItemWriter)。然后再讲 Spring Batch 落地到系统中,设计处 TeacherJob 去完成计算教学天数的功能,并且描述了 Task、TeacherJob、数据库、缓存之间的关系。本节课会基于 Spring Batch,编写代码实现教师教学天数的批量更新。今天课程的内容包括以下几个部分:

- Spring-batch.xml 文件配置
- Spring Batch 中 Step 和 listener 的创建
- TaskService 中启动 TeacherJob

2、Spring-batch.xml 文件配置

如图 1 所示,在 resources 目录下面创建 spring-batch.xml 文件,这个文件会对 计划任务重 TeacherJob 进行描述。下面我们一个一个节点来看看:

- JobRepository 节点,由于这个节点用于 Job 对象实体的存储,因此这里会引用txManager,这个txManager 是来自 spring-jdbc.xml 文件的节点配置。它是用来连接 datasource 的数据源,并且实现了数据库的连接池。
- JobLauncher 节点,是用来启动 Job 的启动器,在 Spring Batch 架构中它是依赖
 JobRepository 的,因此这里需要引用 JobRepository 才能工作。
- 接下来就是主角 teatcherJob 登场了,在它的 step 节点中定义了两部分的内容, 首先是 listener,定义了 stepExecutionStatsListener ,定义它的目的是在 step 的 三个步骤执行完毕以后进行日志的记录。然后就是 teacherReader (教师信息读取

- 器)、teacherItemProcessor(教师信息处理器)以及 teacherWriter(教师信息写入器)。并且讲数据提交的条数设置为 5。
- 完成 teacherJob 的定义以后,进入 teacherReader 的定义,对于它而言由于需要获取数据库中教师信息,因此制定 dataSource 以及 SQL 语句,并且加入了rowMapper 的定义,这个 rowMapper 是用来将获取的教师信息映射到实体类中的。
- 在 teacherReader 后面跟着 teacherWrite、teacherRowMapper、
 jobParametersBuilder 的实体化定义,它们的具体实现会在后面提到。

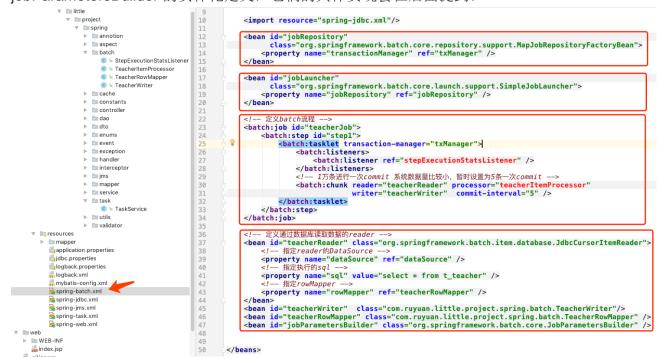


图 1 spring-batch.xml 配置

3、Spring Batch 中 Step 和 listener 的创建

看完了 xml 文件中定义的 JobRepository、JobLauncher 以及 TeacherJob 的相关 定义以后,再来看看具体 Job 中的实现步骤,也就是 Step 的定义。

上面提到了 TeacherReader 的实现,就是从数据库中获取教师信息,这里接着看 TeacherItemProcessor 做了哪些操作。

如图 2 所示,再 spring/batch 目录下创建 TeacherItemProcessor 类,它实现 ItemProcessor Override 了 process 方法,方法体对传入的 Teacher 对象进行操作,通过工具类 DateUtils 中的 betweenOfDay 方法获取教师开始教学时间与当前时间的差,也就是教师教学天数的信息,并且将该信息放入 Teacher 对象中保存,然后返回。

```
▼ Image spring

                                     10
   ▶ ■ annotion
                                              * @author <a href="mailto:little@163.com">little</a>
     aspect
                                                 version: 1.0
   ▼ 🛅 batch
                                               * Description:spring实战
         C & StepExecutionStatsListener
                                     14
15
© 🖫 TeacherItemProcessor
                                             @Component("teacherItemProcessor")
         © 🖆 TeacherRowMapper
                                     16
                                             public class TeacherItemProcessor implements ItemProcessor<Teacher, Teacher> {
         © 🖫 TeacherWriter
                                     17
18 1
                                                  @Override
   ▶ □ cache
                                                   public Teacher process(Teacher teacher) throws Exception {
     constants
                                                       String startTime = teacher.getStartTime();
int betweenDays = (int) DateUtils.betweenOfDay(startTime, LocalDateTime.now());
teacher.setTeachingDays(betweenDays);
                                     19
     controller
                                     20
21
   ▶ □ dao
     dto
                                                        return teacher;
   ▶ ■ enums
                                     23
24
                                                  }
   ▶ ■ event
```

图 2 TeacherItemProcessor 类

接在处理器后面的是写入器,如图 3 所示,在 spring/batch 目录下面创建 TeacherWriter 类,顾名思义它是用来执行教师信息写入的。它的动作很简单,实现 ItemWriter 之后 Override write 方法,执行 teacherMapper 的 updateTeachingDays 方法直接请求数据库,更新 Teacher 实体的信息。

```
package com.ruyuan.little.project.spring.batch;
uyuan
little 🗎
                                  3
                                        import ...
project
                                  9
  spring
                                 10
    annotion
                                         * @author <a href="mailto:little@163.com">little</a>
      aspect
                                           version: 1.0
    batch
                                         * Description:spring实战
        © > StepExecutionStatsListener
                                 14
        © = TeacherItemProcessor
                                 15
                                        public class TeacherWriter implements ItemWriter<Teacher> {
         💿 🖫 TeacherRowMapper
                                 16
                                 17
      cache
                                 18
                                             private TeacherMapper teacherMapper;
     constants
                                 19
      controller
                                 20
      🛅 dao
                                 21 📭
                                             public void write(List<? extends Teacher> list) throws Exception {
     🛅 dto
                                 22
                                                 for (Teacher teacher : list){
     enums enums
                                 23
                                                     int update = teacherMapper.updateTeachingDays(teacher);
    ▶ D event
                                 24
    exception
                                 25
                                             }
    ▶ lim handler
                                 26
27
                                        }
     interceptor
```

图 3 TeacherWriter 类

数据库对象转化为实体对象是由 TeacherRowMapper 类完成的,如图 4 所示,它通过实现 RowMapper Override mapRow 方法,将从数据库中查询的数据映射到

Teacher 实体中,方便处理器操作。

```
annotion
                                      * @author <a href="mailto:little@163.com">little</a>
                              10
 aspect
                              11
                                        version: 1.0

▼ lim batch

                                      * Description:spring实战
    StepExecutionStatsListener
    © № TeacherItemProcessor
                                     public class TeacherRowMapper implements RowMapper<Teacher> {
                              14
   © TeacherRowMapper
                              15
                                          @Override
    TeacherWriter
                                          public Teacher mapRow(ResultSet rs, int rowNum) throws SQLException {
                              16 🗊
▶ □ cache
                                              Teacher teacher = new Teacher();
teacher.setId(rs.getInt(columnLabel: "id"));
                              17
constants
                              18
▶ □ controller
                                              teacher.setTeacherName(rs.getString( columnLabel: "teacher_name"));
                              19
▶ □ dao
                                              teacher.setCourse(rs.getString(columnLabel: "course"));
                              20

    dto

                                              teacher.setScore(rs.getBigDecimal(columnLabel: "score"));
                              21
▶ ■ enums
                                              teacher.setTeachingDays(rs.getInt( columnLabel: "teaching_days"));
                              22
▶ ■ event
                                              teacher.setStatus(rs.getString( columnLabel: "status"));
                              23
exception
                                              teacher.setStartTime(rs.getString( columnLabel: "start_time"));
                              24
handler
                              25
                                              return teacher;
▶ interceptor
                              26
▶ 🛅 ims
                              27
▶ mapper mapper
                              28
service
```

图 4 TeacherRowMapper 类

这部分最后要说的是 StepExecutionStatsListener 类,它是用来在每个 Step 之后记录日志用的。如图 5 所示,它实现了 StepExecutionStatsListenerSupport,通过

Override afterStep 方法,从而在每个 step 执行之后加入日志,方面我们调试。

```
import ...
project
 spring
annotion
aspect
                                               10
11
12
13
14
15
16
17
18
19
20
21
22
23 •1
24
25
26
27
                                                             * @author <a href="mailto:little@163.com">little</a>
                                                                version: 1.0
Description:spring实战
  ▼ 🛅 batch
© 18 StepExecutionStatsListener
© 18 TeacherItemProcessor
© 18 TeacherRowMapper
© 18 TeacherWriter
                                                           public class StepExecutionStatsListener extends StepExecutionListenerSupport {
   Logger log = LoggerFactory.getLogger(getClass());
 cache
constants
controller
                                                                  @Override
                                                                  /**
* 记录batch更新数据条数
    dao
  ▶ 🛅 dto
                                                                  *//
public ExitStatus afterStep(StepExecution stepExecution){
    log.info("write " + stepExecution.getWriteCount() + " items in step: " + stepExecution.getStepName());
     event
     exception
    handler
interceptor
    ims jms
```

图 5 StepExecutionStatsListener 类

4、TaskService 中启动 TeacherJob

上面把 Spring Batch 中的要素都进行了定义,包括: JobRepository、

JobLauncher、Job、Step(ItemReader、ItemProcessor、ItemWriter)以及负责 实体映射的 Mapper 和日志记录的 StepExecutionStatsListener。 可谓是万事具备只欠东风了,这里我们将其启动。

如图 6 所示,回到 TaskService 类这个是我们上周对教师评价信息进行定义的地方,在该类中添加一个 updateTeachingDays 的方法,方法体中通过 JobParametersBuilder 加入日期作为参数,这样保证每个 Job 启动时由于启动的时间不一样而区别于其他 Job 实体。说白了这里用时间作为 Key 去定义每个 Job 实体。接下来就是通过 JobLauncher 的 run 方法,传入 TeacherJob 实体和参数,并对 TeacherJob 进行启动。

```
120
121
     ▶ interceptor
                                         122
                                                          * 定时任务更新教师
       ims jms
                                                         * 此处更新教师信息,将teacher缓存全部清除
     ▶ mapper mapper
     ▶ b service
▼ task
                                         124
125
                                                        @CacheEvict(value = "teacher", allEntries = true)
                                                        public void updateTeachingDays() throws Exception {
// 添加一个JobParameters构建类,以当前时间作为参数,保证其他参数相同的情况下却是不同的job instance
                                        126
127
          C TaskService
    ▶ 🖿 utils
                                                             LOGGER.info("开始更新教师教学天数信息");
jobParametersBuilder.addDate("date", new Date());
                                        128
129
                                                             jobLauncher.run(teacherJob, jobParametersBuilder.toJobParameters()); LOGGER.info("更新教师教学天数信息完成");
                                         130
ication.properties
properties
ack.properties
```

图 6 TaskService 修改

定义完 TaskService,通过 updateTeachingDays 方法启动 TeacherJob 之后,回到 spring-batch.xml 文件中,通过计划任务定义 TeacherJob 的执行实现。这里依旧使用 cron 表达式,定义 TaskService 中的 updateTeachingDays 方法在凌晨十

二点十分的时候执行。当然,下面有一个 remark 的配置,是给我们测试用的。

```
nttp://www.springtramework.org/scnema/task/spring-task.xsd
                 ijdbc.properties
                                                                                                              cbean id="taskService" class="com.ruyuan.little.project.spring.task.TaskService"></bean>
<!-- 线程个数(默认为1个,多个任务同一时间内只能串行执行,设置多个则可以并行执行) -->
<task:scheduler id="taskScheduler" pool-size="5"></task:scheduler>
                 👼 logback.xml
                 mybatis-config.xml
                spring-jdbc.xml
                                                                                                              <!-- 开启任务 --> <!-- 配置相关类中方法的执行时间-
                  눩 spring-jms.xm
                                                                                                              <!-- 配置相关集中方法的执行时间-->
<task:scheduled-tasks scheduler="taskScheduler">
<!-- 毎陽五分钟(5*60*1000)执行一次, fixed-rate単位为毫秒 -->
        <task:scheduled ref="taskService" method="updateTeacherScore" fixed-rate="300000"/>
<!-- 使用cron表达式, 每天凌晨十二点一分执行 -->
        <task:scheduled ref="taskService" method="expireCoupon" cron=" 0 1 0 * * ?"/>
        <!-- 使用cron表达式, 每天凌晨十二点十分执行 -->
        <task:scheduled ref="taskService" method="updateTeachingDays" cron=" 0 10 0 * * ?"/>

                spring-web.xm
      web
        index.jsp
    gitignore
   # little-project-spring.iml
pom.xml
README.md
                                                                                                                       <!-- 调试时可以将执行时间调整为1分钟 -
External Libraries
                                                                                                              - <task:scheduled ref='
</task:scheduled-tasks>
                                                                                                                                                                           taskService" method="updateTeachingDays" fixed-rate="60000"/>
                                                                                                      </beans>
```

图 7 spring-task.xml 修改

5、总结

本节课对 Spring Batch 中的要素都进行了定义,包括: JobRepository、
JobLauncher、Job、Step(ItemReader、ItemProcessor、ItemWriter)以及负责

实体映射的 Mapper 和日志记录的 StepExecutionStatsListener。同时通过修改 TaskService 加入 updateTeachingDays 方法启动 TeacherJob,在 springtask.xml 中修改了 TeacherJob 的执行时间。

下节课是代码验证的环节,验证一下任务调度执行后,教师教学天数是否更新。这 里将<u>代码</u>给大家,下期见,拜拜。

友情提示:本章讲述代码只是部分核心代码,完整代码请查阅文末链接中代码,谢谢。