1.   首先，说说 quartz.config 文件

1)   先将上一篇中的配置贴出来

# You can configure your scheduler in either<quartz> configuration section

# or in quartz properties file

# Configuration section has precedence

quartz.scheduler.instanceName = TopshelfAndQuartz

# configure thread pool info

quartz.threadPool.type = Quartz.Simpl.SimpleThreadPool, Quartz

quartz.threadPool.threadCount = 10

quartz.threadPool.threadPriority = Normal

# job initialization plugin handles our xml reading, without it defaults are used

quartz.plugin.xml.type = Quartz.Plugin.Xml.XMLSchedulingDataProcessorPlugin, Quartz

quartz.plugin.xml.fileNames = ~/quartz\_jobs.xml

# export this server to remoting context

#quartz.scheduler.exporter.type = Quartz.Simpl.RemotingSchedulerExporter, Quartz

#quartz.scheduler.exporter.port = 555

#quartz.scheduler.exporter.bindName = QuartzScheduler

#quartz.scheduler.exporter.channelType = tcp

#quartz.scheduler.exporter.channelName = httpQuartz

2)   以上标注为红色的配置修改的可能性相对较大。

1.   quartz.scheduler.instanceName 指定调度器的实例名称。

2.   quartz.threadPool.threadCount 设置线程池的最大线程数量。

3.   quartz.threadPool.threadPriority 设置作业中每个线程的优先级，可取 System.Threading.ThreadPriority 中的枚举。

4.   quartz.plugin.xml.fileNames 指定 Quartz 的作业配置文件路径及名称，以上指定为 quartz\_jobs.xml，默认情况下命名为该名称即可。

3)   其他的没什么说的，因为以上配置是 Quartz 的标准配置，如没有特殊要求是不需要修改的。所以，这里一带而过，以后有必要时再来研究它。

2.   然后，是 quartz\_jobs.xml文件

1)   同样，以上篇的配置为例：

<?xml version="1.0" encoding="UTF-8"?>

<!-- This file contains job definitions in schema version 2.0 format -->

<job-scheduling-data xmlns="http://quartznet.sourceforge.net/JobSchedulingData" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0">

  <processing-directives>

    <overwrite-existing-data>true</overwrite-existing-data>

  </processing-directives>

  <schedule>

    <!--该作业用于定时更新商品库存-->

    <job>

      <name>UpdateInventoryJob</name>

      <group>Update</group>

      <description>定时更新商品库存</description>

      <job-type>TopshelfAndQuartz.UpdateInventoryJob,TopshelfAndQuartz</job-type>

      <durable>true</durable>

      <recover>false</recover>

    </job>

    <trigger>

      <cron>

        <name>UpdateInventoryTrigger</name>

        <group>Update</group>

        <job-name>UpdateInventoryJob</job-name>

        <job-group>Update</job-group>

        <start-time>2017-12-01T00:00:00+08:00</start-time>

        <cron-expression>0 0/1 \* \* \* ?</cron-expression>

      </cron>

    </trigger>

  </schedule>

</job-scheduling-data>

2)   这个配置文件相当重要，所有作业的调度都是由该文件定义的。其中最重要的节点是 cron-expression 和 repeat-interval 节点，下面是每个节点的具体说明：

1.   job 作业节点，每个作业则对应一个 job 节点。

1)   name（必填）        作业名称，同一个 group 中作业名称不能相同。

2)   group（选填）       作业分组名称，表示该作业所属分组。

3)   description（选填） 作业描述，用于描述该作业的具体功能。

4)   job-type（必填）    指定作业将调用的作业实现类，格式为：命名空间.类名,程序集名称。

5)   durable（选填）     表示该作业是否长久的，具体作用不详，默认为true。

6)   recover（选填）     字面意思是恢复，具体作用不详，默认为false。

2.   trigger 作业触发器节点，用于定义指定的作业以何种方式触发，一个作业可以有多个触发器，而每个触发器都独立执行调度。

Ø  触发器可以为以下三种类型：

1)   **calendar-interval**这种触发器类型使用较少，此处省略。

2)   **simple** 简单触发器，可用于调度重复执行的作业。

1.   name（必填）        触发器名称，同一个 group 中作业名称不能相同。

2.   group（选填）       触发器分组名称，表示该触发器所属分组。

3.   job-name（必填）    要调度的作业名称，必须与 job 节点中的 name 相同。

4.   job-group（选填）   要调度的作业分组名称，必须与 job 节点中的 group 相同。

5.   start-time（选填）  开始作业的 utc 时间，北京时间需要+08:00，例如：<start-time>2017-12-01T08:00:00+08:00</start-time>，表示北京时间2017年12月1日上午8:00开始执行。注意：服务启动或重启时都会检测此属性。若没有设置此属性或者 start-time 的时间小于当前时间，服务启动后会立即执行一次调度；若大于当前时间，服务会等到当前时间等于 start-time 的时间才开始执行。如果没有特殊要求，可以忽略该属性。

6.   repeat-count（选填）表示作业重复执行次数，-1表示无限制重复执行，3 表示作业重复执行三次。

7.   repeat-interval（必填）表示作业触发间隔时间，10000 表示十秒触发一次，单位：毫秒。

8.   示例：

<simple>

  <name>UpdateInventoryTrigger</name>

  <group>Update</group>

  <job-name>UpdateInventoryJob</job-name>

  <job-group>Update</job-group>

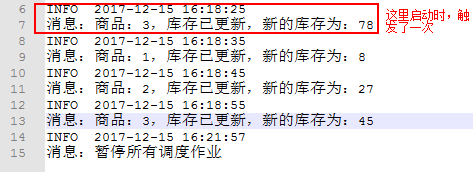
  <start-time>2017-12-01T00:00:00+08:00</start-time>

  <repeat-count>3</repeat-count>

  <repeat-interval>10000</repeat-interval>

</simple>

以上配置将在服务启动后，每十秒触发一次作业，共触发三次，如图：

[](https://img2018.cnblogs.com/blog/654920/201809/654920-20180903141940007-1388006148.png)

1)   **cron** 复杂触发器，使用 cron-expression 设置触发器的行为。

1.   name                与 simple 相同。

2.   group               与 simple 相同。

3.   job-name            与 simple 相同。

4.   job-group           与 simple 相同。

5.   start-time          开始作业的 utc 时间，北京时间需要+08:00，例如：<start-time>2017-12-01T08:00:00+08:00</start-time>，表示北京时间2017年12月1日上午8:00开始执行。注意：服务启动或重启时都会检测此属性。若没有设置此属性，服务会根据 cron-expression 的规则执行作业调度；若 start-time 的时间小于当前时间，服务启动后会忽略 cron-expression 的设置，立即执行一次调度，之后再根据 cron-expression 执行作业调度；若大于当前时间，服务会等到当前时间等于 start-time 的时间才开始执行，并根据 cron-expression 执行作业调度。如果没有特殊要求，可以忽略该属性。

6.   cron-expression（必填）cron 表达式，例如：<cron-expression>0 0/1 \* \* \* ?</cron-expression>，表示间隔1分钟触发一次。注意：该节点为必须，如果省略整个服务将不能正常运行！

3.   详解 cron-expression 节点

Ø  这个节点的设置相对复杂很多，因为所有的触发需求都是由该节点来控制，比如（未经测试）：

1.   每月1号1点触发一次，表达式：0 0 1 1 \* \*

2.   每周日12点触发一次，表达式：0 0 12 ? \* SUN

3.   每天9点到18点每小时触发一次，表达式：0 0 9-18 \* \* ?

Ø  详细说明

1.   一个cron表达式有至少6个（或7个）由空格分隔的时间元素，按顺序依次为：[秒] [分] [小时] [日] [月] [周] [年]

元素    必填        取值范围                通配符

秒     是          0~59                    , - \* /

分     是          0~59                    , - \* /

时      是          0~23                    , - \* /

日     是          1~31                    , - \* ? / L W

月     是          0-11 or JAN-DEC         , - \* /

周     是           1-7 or SUN-SAT          , - \* ? / L #

年     否           empty or 1970-2099      , - \* /

其中，周取值：1=周日~7=周六，1=SUN，2=MON，3=TUE，4=WED，5=THU，6=FRI，7=SAT。

2.   通配符说明：

1)   **\*** 表示所有值，例如：分设置为"\*"，表示每一分钟都会触发。

2)   **?** 表示不指定值，使用场景不需要关心当前的这个元素。例如：要在每月的10号触发一个操作，但不关心是周几，所以周设置为"?"，具体设置为 0 0 0 10 \* ?。

3)   **-**表示区间，例如：在小时上设置"10-12"，表示10,11,12点都会触发。**注意：**这里的区间是包头包围的，比如配置是：0 0/15 9-22 \* \* ?，则表示9:00~22:59，每15分钟触发一次。

4)   **,** 表示指定多个值（列表），例如：在周字段上设置"MON,WED,FRI"，表示周一、周三、周五都会触发。

5)   **/** 用于递增触发，例如：在秒上面设置"5/15"，表示从5秒开始，每增15秒触发一次(5,20,35,50)。

6)   **L** 表示最后的意思。在日字段设置上，表示当月的最后一天（依据当前月份，如果是二月还会依据是否是润年[leap]）；在周字段上表示星期六，相当于"7"或"SAT"。如果在"L"前加上数字，则表示该数据的最后一个。例如：在周字段上设置"6L"这样的格式，则表示“本月最后一个星期五”。

7)   **W** 表示离指定日期的最近那个工作日(周一至周五)。例如：在日字段上设置"15W"，表示离每月15号最近的那个工作日触发。如果15号正好是周六，则找最近的周五(14号)触发，如果15号是周未，则找最近的下周一(16号)触发。如果15号正好在工作日(周一至周五)，则就在该天触发。如果指定格式为"1W"，它则表示每月1号往后最近的工作日触发。如果1号正是周六，则将在3号下周一触发。（注："W"前只能设置具体的数字,不允许区间"-"）。

8)   **#** 序号，表示每月的第几个周几。例如：在周字段上设置"6#3"表示在每月的第三个周六。注意如果指定"#5"，正好第五周没有周六，则不会触发该配置（用在母亲节和父亲节再合适不过了）；小提示：'L'和'W'可以一组合使用。如果在日字段上设置"LW"，则表示在本月的最后一个工作日触发；周字段的设置，若使用英文字母是不区分大小写的，即MON与mon相同。

3.   常用示例：

0 0 12 \* \* ?            每天12点触发

0 15 10 ? \* \*           每天10点15分触发

0 15 10 \* \* ?           每天10点15分触发

0 15 10 \* \* ? \*         每天10点15分触发

0 0 12 ? \* WED          每周三12点触发

0 0/30 9-17 \* \* ?       每天九点到十五店，每半小时触发

0 10,14,16 \* \* ?        每天上午10点，下午2点、4点触发

0 15 10 \* \* ? 2005      2005年每天10点15分触发

0 \* 14 \* \* ?            每天下午的2点到2点59分每分触发

0 0/5 14 \* \* ?          每天下午的2点到2点59分（整点开始，每隔5分触发）

0 0/5 14,18 \* \* ?       每天下午的 2点到2点59分、18点到18点59分（整点开始，每隔5分触发）

0 0-5 14 \* \* ?          每天下午的 2点到2点05分每分触发

0 10,44 14 ? 3 WED      3月分每周三下午的 2点10分和2点44分触发

0 15 10 ? \* MON-FRI     从周一到周五每天上午的10点15分触发

0 15 10 15 \* ?          每月15号上午10点15分触发

0 15 10 L \* ?           每月最后一天的10点15分触发

59 59 23 L \* ?          每月最后一天的23点59分59秒触发（经测试）

0 15 10 ? \* 6L          每月最后一周的星期五的10点15分触发

0 15 10 ? \* 6L 2002-2005    从2002年到2005年每月最后一周的星期五的10点15分触发

0 15 10 ? \* 6#3         每月的第三周的星期五开始触发

0 0 12 1/5 \* ?          每月的第一个中午开始每隔5天触发一次

0 11 11 11 11 ?         每年的11月11号 11点11分触发（光棍节）

Ø  总结

本篇主要详细的学习了 Quartz 的相关配置，这应该是 Quartz 的重要知识点，要一下子记住这些配置是不太可能的。所以，还需要在以后的工作中多多运用才行。

<https://www.cnblogs.com/abeam/p/8044460.html>