



链滴

java时间调度之SpringQuartz调度框架使用

作者: [terry](#)

原文链接: <https://ld246.com/article/1446648323554>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Quartz 是一个开源的作业调度框架，设计用于 J2SE 和 J2EE 应用中。提供了巨大的灵活性而不牺牲简单性。能够用它来为执行一个作业而创建简单的或复杂的度

Spring Quartz 是基于 Quartz 做的2次封装

Quartz分为2种调度器，简单调度和任务调度，核心就是3个bean

`org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean` -- 定制任务类

`org.springframework.scheduling.quartz.SimpleTriggerBean` -- 简单触发器

`org.springframework.scheduling.quartz.CronTriggerBean` -- 任务触发器

`org.springframework.scheduling.quartz.SchedulerFactoryBean` --调度工厂

一、调度的步骤

1、制定任务信息 bean 设置执行对象 设置执行对象中对应的执行方法 是否可以同步执行

2、制定任务执行时机（执行触发器） bean 2.1、简单触发器或任务调度器（SimpleTrigger, CronTriggerBean） 设置任务详细（二者都需要） 设置任务延迟执行时间（任务调度器不需要配置） 设置任务执行频率（任务调度器不需要配置） 设置任务执行时机，cron表达式（任务调度器必须用）

2.2、任务触发器（CronTrigger） 设置任务详细 设置执行时机（cronExpression） cronExpression:秒分日月周年（可选） 3、设置任务调度工厂 bean 设置触发器们（可能是一个或多个触发器）

二、调度器demo

1、简单触发器

第一步先定义一个普通类，新增需要执行的方法

```
import java.text.SimpleDateFormat;
import java.util.Date;
//简单触发器、打印出来当前时间
public class QuartzTask {
    public void doSimpleTriggerTask() {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        System.out.println("doing simpleTrigger task..." + sdf.format(new Date()));
    }
}
```

第二步编写spring配置文件

<!-- 注册一个普通bean -->
<bean id="quartzTask" class="cn.terry.nsfw.complain.QuartzTask"/>
<!-- 1、制定任务信息 -->
<bean id="jobDetail1" class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">
<!-- 设置执行对象 -->
<property name="targetObject" ref="quartzTask"/>
<!-- 设置执行对象中对应的执行方法 -->
<property name="targetMethod" value="doSimpleTriggerTask"/>
<!-- 是否可以同步执行；不可同步执行 -->
<property name="concurrent" value="false"/>
</bean>
<!-- 2、制定任务执行时机（执行简单触发器） -->

```

/><span style="color: #ff00ff;"> &lt;bean id="simplerTrigger" class="org.springframework.sc
eduling.quartz.SimpleTriggerBean"&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp;
&nbsp; &lt;!-- 设置任务详细 --&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp;
&nbsp; &lt;property name="jobDetail" ref="jobDetail1"&gt;&lt;/property&gt;</span> <br /> <span s
yle="color: #ff00ff;">&nbsp; &nbsp; &lt;!-- 设置任务延迟执行时间 ; 延迟2秒执行--&gt;</span
<br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;property name="startDelay" value="
000"&gt;&lt;/property&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;!--
设置任务执行频率; 执行频率为每2秒执行一下 --&gt;</span> <br /> <span style="color: #ff00ff;
">&nbsp; &nbsp; &lt;property name="repeatInterval" value="2000"&gt;&lt;/property&gt;</s
an> <br /> <span style="color: #ff00ff;"> &lt;/bean&gt;</span> <br /> <span style="color: #ff
00ff;"> &lt;!-- 3、设置调度工厂 --&gt;</span> <br /> <span style="color: #ff00ff;"> &lt;bean cla
s="org.springframework.scheduling.quartz.SchedulerFactoryBean"&gt;</span> <br /> <span s
yle="color: #ff00ff;">&nbsp; &nbsp; &lt;property name="triggers"&gt;</span> <br /> <span
tyle="color: #ff00ff;">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &lt;list&gt;</span> <br /> <span style="
olor: #ff00ff;">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &lt;ref bean="simplerTrigger"/
&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &lt;/list&gt;</
pan> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;/property&gt;</span> <br /> <s
an style="color: #ff00ff;"> &lt;/bean&gt;</span> </p>

```

<p>至此启动spring 2秒以后就会每2秒自动输出当前的时间</p>

<p>2、任务触发器</p>

<p>第一步先定义普通类，新增需要执行的方法</p>

```

<pre class="prettyprint">import java.text.SimpleDateFormat;
import java.util.Date;
//任务触发器、打印出来当前时间
public class QuartzTask {
    public void doCronTriggerTask() {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        System.out.println("doing cronTrigger task..." + sdf.format(new Date()));
    }
}</pre>

```

<p>第二步编写spring配置文件</p>

```

<p><span style="color: #ff00ff;">&lt;!-- 注册一个普通bean --&gt;</span> <br /> <span style=
color: #ff00ff;">&lt;bean id="quartzTask" class="cn.terry.nsfw.complain.QuartzTask"&gt;&lt;/
ean&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;!-- 1、制定任务信息
息 --&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;bean id="jobDetail2
class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean"&gt;</
pan> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;!-- 设置执行对象 --&gt;</span>
br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;property name="targetObject" ref="qu
rtzTask"&gt;&lt;/property&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;
!-- 设置执行对象中对应的执行方法 --&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp;
&nbsp; &lt;property name="targetMethod" value="doCronTriggerTask"&gt;&lt;/property&gt;<
span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;!-- 是否可以同步执行; 不可同步
行 --&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;property name="co
current" value="false"&gt;&lt;/property&gt;</span> <br /> <span style="color: #ff00ff;"> &lt;
bean&gt;</span> <br /> <span style="color: #ff00ff;"> &lt;!-- 2、制定任务执行时机 (任务执行
发器) --&gt;</span> <br /> <span style="color: #ff00ff;"> &lt;bean id="cronTrigger" class="
rg.springframework.scheduling.quartz.CronTriggerBean"&gt;</span> <br /> <span style="col
r: #ff00ff;">&nbsp; &nbsp; &lt;!-- 设置任务详细 --&gt;</span> <br /> <span style="color: #ff0
ff;">&nbsp; &nbsp; &lt;property name="jobDetail" ref="jobDetail2"&gt;&lt;/property&gt;</s
an> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;!-- 设置任务执行时机, cron表达式
--&gt;</span> <br /> <span style="color: #ff00ff;">&nbsp; &nbsp; &lt;property name="cronE
pression" value="* * * 18c * ?"&gt;&lt;/property&gt;</span> <br /> <span style="color: #ff00f
;"> &lt;/bean&gt;</span> <br /> <span style="color: #ff00ff;"> &lt;!-- 3、设置调度工厂 --&gt;
/pan> <br /> <span style="color: #ff00ff;"> &lt;bean class="org.springframework.scheduling.

```


<p> </p>