



链滴

# SpringBoot 整合 Quartz

作者: [liumapp](#)

原文链接: <https://ld246.com/article/1515576606103>

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是一个相当优秀的任务调度项目，在SpringBoot中对其进行集成也是一件不算难也不算容易事情。然而，该有的坑总是会来的。

## 前言

废话不多说，先上Demo。

Github : [quartz-schedule-demo](#)

这个项目是我将Quartz官方案例在SpringBoot上的一个Demo实现，同时配上了必要的注释和适当调整。

对于新接触的同学来说，应该是一个不错的上手项目。

## 案例

[quartz-schedule-demo](#)项目中目前一共包含8个Demo，后期添加新的Demo时，我会更新这篇博文。

分别是:

### Demo1

简单的循环任务，每隔10秒一次输出

- 启动Test下SimpleTest的demo1Test（请将其ignore注释）
- SimpleJob将每隔10秒钟运行一次。

### Demo2

传递参数到Jobs中，直接从context中读取参数。

- 启动Test下DataParseTest的parameterTest（请将其ignore注释）
- DataParseJob将每隔10秒运行一次。

### Demo3

同样传递参数到Jobs中，但将参数写入Jobs的属性。

- 启动Test下DataParseTest的parameterAutoWriteTest（请将其ignore注释）
- HighLevelParameterJob将每隔10秒运行一次。

### Demo4

延时五秒后执行任务

- 启动Test下SimpleTest的demo4Test（请将其ignore注释）
- 五秒后控制台可以看到输出。

## Demo5

延时五秒后启动一个任务，该任务每隔10秒执行一次，直到执行了10次为止。

- 启动Test下SimpleTest的demo5Test（请将其ignore注释）
- 五秒后控制台可以看到输出。（一共会看到11行输出）

## Demo6

立即启动一个任务，然后每隔5分钟执行一次，直到当天下午4点为止。

- 启动Test下SimpleTest的demo6Test（请将其ignore注释）
- 检查结果

## Demo7

每个偶数小时整（14:00，16:00）执行一次任务。

- 启动Test下SimpleTest的demo7Test（请将其ignore注释）
- 检查结果

## Demo8

通过配置，将Spring的Bean对象自动加载到Quartz的JobFactory中。

Demo8检测了Quartz的job在执行的过程中，能否正常使用Spring的Bean对象

- 启动Test下HelloTest的Demo8Test（请将其ignore注释）
- 检查运行结果

## 排坑

在读官方文档的过程中，基本上从Demo1一直跑到Demo7都没什么问题，唯一的大问题出现在Quartz的SimpleThreadPool上，这个线程池在执行Jobs的过程中，如果缺少对JobFactory配置Spring的Context，那么执行的Job将无法解析Spring的Bean对象。

以下是详细的解决办法：

- 引入spring-context-support，在pom.xml中加下面的依赖

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
  <version>4.3.13.RELEASE</version>
</dependency>
```

- 配置applicationContext.getAutowireCapableBeanFactory()，加载到JobFactory中即可。（详情代码请见项目中的config）