



链滴

# Quartz 与 spring 整合之数据库存储

作者: [zml2015](#)

原文链接: <https://ld246.com/article/1511143614368>

来源网站: [链滴](#)

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

更多详细内容请参考[官网文档](#)

## 只使用quartz.properties

```
#①集群的配置，这里不使用集群
org.quartz.scheduler.instanceName = DefaultQuartzScheduler
org.quartz.scheduler.rmi.export = false
org.quartz.scheduler.rmi.proxy = false
org.quartz.scheduler.wrapJobExecutionInUserTransaction = false
## 任务超时时间，默认为60s
org.quartz.jobStore.misfireThreshold=60000
## 一次性最多执行多少个超时任务
org.quartz.jobStore.maxMisfiresToHandleAtATime=20
# Cluster集群
org.quartz.jobStore.isClustered = false
#ID设置为自动获取 每一个必须不同
org.quartz.scheduler.instanceId = AUTO
#②配置调度器的线程池
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
## 并行执行最大线程数
#org.quartz.threadPool.threadCount=5
##线程优先级
org.quartz.threadPool.threadPriority = 5
org.quartz.threadPool.threadsInheritContextClassLoaderOfInitializingThread = true
#③配置任务调度现场数据保存机制
## org.quartz.simpl.RAMJobStore为使用内存机制，将job和trigger信息保存到内存中，缺点，jv
停止后，内存信息丢失
#org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
## org.quartz.impl.jdbcjobstore.JobStoreTX完全符合JDBC的驱动程序,org.quartz.impl.jdbcjobst
re.oracle.OracleDelegate,这里使用数据库驱动方式，将trigger和job信息保存到数据库中
org.quartz.jobStore.class = org.quartz.impl.jdbcjobstore.JobStoreTX
#quartz数据库表名前缀,默认值为QRTZ_
org.quartz.jobStore.tablePrefix = QRTZ_
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.oracle.OracleDelegate

## dataSource
##库的别名
org.quartz.jobStore.dataSource = myDS
org.quartz.dataSource.myDS.driver = oracle.jdbc.driver.OracleDriver
org.quartz.dataSource.myDS.URL = jdbc:oracle:thin:@localhost:1521:orcl
org.quartz.dataSource.myDS.user = SCOTT
org.quartz.dataSource.myDS.password = mingliang
org.quartz.dataSource.myDS.validationQuery= select 1 from dual
org.quartz.dataSource.myDS.maxConnections = 20

org.quartz.jobListener.NAME.class=com.dtsz.scheduletask.quartz.job.CountJobListener
org.quartz.triggerListener.NAME.class=com.dtsz.scheduletask.quartz.job.CountJobListener
```

## 与Spring整合

可以将所有配置都配置到spring的配置文件中，仅使用spring的配置，但也可以使用部分spring配置和部分quartz.properties中的配置，如果两个配置文件中均有相应配置，则优先使用spring中的配

## 信息

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf
amework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/con
ext http://www.springframework.org/schema/context/spring-context.xsd">
  <context:component-scan base-package="com.dtsz.scheduletask.quartz"/>

  <bean id="customJobFactory" class="com.dtsz.scheduletask.quartz.job.CustomJobFactory"
  >
  <bean id="scheduler" class="org.springframework.scheduling.quartz.SchedulerFactoryBea
">
  <property name="jobFactory" ref="customJobFactory"/>
  <property name="dataSource">
    <ref bean="dataSource"/>
  </property>
  <property name="applicationContextSchedulerContextKey" value="applicationContextK
y"/>
  <!--<property name="configLocation" value="classpath:quartz.properties"/>-->
  <property name="quartzProperties">
    <props>

      <prop key="org.quartz.scheduler.instanceName">quartzScheduler</prop>
      <prop key="org.quartz.scheduler.instanceId">AUTO</prop>
      <prop key="org.quartz.scheduler.skipUpdateCheck">true</prop>
      <prop key="org.quartz.threadPool.class">org.quartz.simpl.SimpleThreadPool</pro
>
      <prop key="org.quartz.threadPool.threadCount">15</prop>
      <prop key="org.quartz.threadPool.threadPriority">5</prop>
      <prop key="org.quartz.jobStore.misfireThreshold">600000</prop>
      <prop key="org.quartz.jobStore.maxMisfiresToHandleAtATime">20</prop>
      <!--注意：如果数据源采用spring管理，则一定要注释掉下面的jobstore的配置-->
      <!--<prop key="org.quartz.jobStore.class">org.quartz.impl.jdbcjobstore.JobStoreT
</prop>-->
      <!--<prop key="org.quartz.jobStore.driverDelegateClass">org.quartz.impl.jdbcjobs
ore.oracle.OracleDelegate</prop>-->
      <!--userProperties设置为true时，trigger或者job中的dataMap的类型必需为String类
，false可以设置为自定义类型，但是有时候会出现异常，目前异常未能复现-->
      <prop key="org.quartz.jobStore.useProperties">false</prop>
      <!--指定job覆盖，如果这个覆盖配置为false，quartz启动以后将以数据库的数据为准，
置文件的修改不起作用-->
      <prop key="overwriteExistingJobs">true</prop>

      <prop key="org.quartz.jobStore.tablePrefix">QRTZ_</prop>
      <prop key="org.quartz.jobListener.NAME.class">com.dtsz.scheduletask.quartz.job.
ountJobListener</prop>
      <prop key="org.quartz.triggerListener.NAME.class">com.dtsz.scheduletask.quartz.j
b.CountJobListener</prop>

    </props>
  </property>
```

```
</bean>
```

```
</beans>
```