spring整合quartz有两种方式:

一.常见是使用配置文件,将定时任务保存到内存中

<!-- 短信催还提醒任务调度 -->

<bean id="overdueRecall"

class="com.sursen.souba.ddlibserve.quartz.OverdueRecallTimerTask" />

<!--定义定时执行overdueRecallTimerTask 这个bean中的overdueRecall()方法-->

<bean id="overdueRecallTask"

class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">

<property name="targetObject">

<ref bean="overdueRecall" />

</property>

<property name="targetMethod">

<value>overdueRecall</value>

</property>

</bean>

<!--触发器的bean的设置，要触发的jobDetail是overdueRecallTask-->

<bean id="overdueRecallTrigger"

class="org.springframework.scheduling.quartz.CronTriggerBean">

<property name="jobDetail">

<ref bean="overdueRecallTask" />

</property>

<property name="cronExpression">

<!-- 每天17:00运行 -->

<value>0 0 17 \* \* ?</value>

<!--<value>0 05 18 \* \* ?</value>-->

</property>

</bean>

<!--管理触发器的总设置,管理我们的触发器列表,可以在bean的list中放置多个触发器。-->

<bean autowire="no"

class="org.springframework.scheduling.quartz.SchedulerFactoryBean">

<property name="triggers">

<list>

<ref local="overdueRecallTrigger" />

</list>

</property>

</bean>

这种方式配置简单,但是存在问题

1.定时任务信息都报错在内存中,服务器重启会丢失信息

2.每个定时任务都是一串配置,定时任务多了不好管理

3.任务时间修改后要重新发布项目

二.quartz持久化