Spring任务调度：quartz使用；---4.1版，实现方式；

·任务执行和线程池；

多定时任务；*SchedulerFactoryBean*

定时触发；*CronTriggerFactoryBean*

*实现细节：*

*专门写一个spring扩展配置文件；applicationContext-quartz.xml*

spring配置，

<!-- 定时调度任务触发列表-->

<bean name=*"quartzScheduler"* class=*"org.springframework.scheduling.quartz.SchedulerFactoryBean"*>

<property name=*"triggers"*>

<list>

<!-- 债权转让系统自动流标 -->

<ref bean=*"transferFlowUpdateHanlderTime"*/>

</list>

</property>

<property name=*"configLocation"* value=*"classpath:/quartz.properties"*/>

</bean>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*债权转让流标 定时任务开始\*\*\*\*\*\*\*\*\*\* -->

<bean id=*"transferUpdateFlowStatuService"* class=*"com.cxdai.face.quartz.service.TransferUpdateFlowStatuService"*>

</bean>

<!-- 定义调用对象和调用对象的方法 -->

<bean id=*"transferUpdateFlowStatutask"*

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

<!-- 调用的类 -->

<property name=*"targetObject"*>

<ref bean=*"transferUpdateFlowStatuService"* />

</property>

<!-- 调用类中的方法 -->

<property name=*"targetMethod"*>

<value>updateTransferFlowStatue</value>

</property>

<!-- 并发设置 -->

<property name=*"concurrent"* value=*"false"* />

</bean>

<!-- 定义触发时间,每天23点00分00秒时触发 -->

<bean id=*"transferFlowUpdateHanlderTime"* class=*"org.springframework.scheduling.quartz.CronTriggerFactoryBean"*>

<property name=*"jobDetail"*>

<ref bean=*"transferUpdateFlowStatutask"* />

</property>

<!-- cron表达式 -->

<property name=*"cronExpression"*>

<value>00 00 23 \* \* ?</value>

</property>

</bean>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*债权转让流标 定时任务结束\*\*\*\*\*\*\*\*\*\*\*\*\* -->

Quartz.properties，

//

org.quartz.scheduler.instanceName = DefaultQuartzScheduler

org.quartz.scheduler.rmi.export = false

org.quartz.scheduler.rmi.proxy = false

org.quartz.scheduler.wrapJobExecutionInUserTransaction = false

//

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool

org.quartz.threadPool.threadCount = 30

org.quartz.threadPool.threadPriority = 5

org.quartz.threadPool.threadsInheritContextClassLoaderOfInitializingThread = true

//

org.quartz.jobStore.misfireThreshold = 60000

//

org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore

//

#org.quartz.jobStore.class = org.quartz.impl.jdbcjobstore.JobStoreTX

#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.HSQLDBDelegate

#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.oracle.OracleDelegate

//service类：

**public** **class** TransferUpdateFlowStatuService {

@Autowired

TransferUpdateFlowStatuMapper transferUpdateFlowStatuMapper;

**public** **void** updateTransferFlowStatue(){

//调用存储过程

transferUpdateFlowStatuMapper.updateFlowTransferStatus(map);

//xml，---调用存储过程，或自己写sql语句的实现方式；

<select id=*"updateFlowTransferStatus"* statementType=*"CALLABLE"* parameterType=*"java.util.Map"* resultType=*"java.lang.String"* >

<![CDATA[ { call transferCancel(#{id},#{cancelUser},#{cancelIP},#{cancelRemark},#{msg,mode=OUT,jdbcType=VARCHAR}) } ]]>

</select>

------------------

//每天22时30分30秒；

<property name=*"cronExpression"*>

<value>30 30 22 \* \* ?</value>

</property>

//////////////////////////////////////////////