1. 在ServletContext中初始化Scheduler；
2. 将Scheduler赋值到CflowHelper静态变量 CflowHelper.scheduler；
3. startRepeater中。
   1. 新建Job
      1. JobName = USERID+ “-“ + work.sessionId;
      2. JobGroup = Process ID
      3. JobDetail job = new JobDetail( jobName, jobGroup, doRepeater.class);
      4. job.getJobDataMap().put(“USRID”, usrID);
      5. job.getJobDataMap().put(“PRCID”, Process ID);
      6. job.getJobDataMap().put(“SESSID”, work.sessionId);
      7. job.getJobDataMap().put(“NODEID”, work.nodeId);
      8. CflowHelper.scheduler.addJob(job, false);
      9. DbAdmin.addJob(JobName, JobGroup);
   2. 新建CronTrigger.
      1. TriggerName = USERID + “-“ + work.sessionId;
      2. TriggerGroup = Process ID
      3. Trigger trigger = new CronTrigger(TriggerName, TriggerGroup, jobName, jobGroup, startTime, endTime, cronExpression, TimeZone);
      4. CflowHelper.scheduler.scheduleJob(trigger);
4. 进程结束时, 暂停时， 重启时，中断时
   1. DbAdmin.foreach(JobName of Job whose GroupName == Process ID){
      1. CflowHelper.sceduler.deleteJob(jobName, Process ID);
   2. }
5. Cron结束时。

org.quartz.SchedulerListener Interface

CflowHelper.scheduler.AddSchedulerListener();

SchedulerListner.triggerFinalized(Trigger trigger){

JobDataMap dataMap = trigger.getJobDataMap();

doRepeatedTask(dataMap);

}

1. DoReapter:

Public void execute(JobExecutionContext context){

JobDataMap dataMap = context.getJobDetail().getJobDataMap();

DoRepeatedJob(dataMap);

}

7. AddSchedulerListener

SchedulerListner.triggerFinalized(Trigger trigger)

8. DoRepeatedJob(JobDataMap dataMap){

String usrid = dataMap.getString(“USRID”);

String prcid = dataMap.getString(“PRCID”);

String sessid = dataMap.getString(“SESSID”);

String nodeid = dataMap.getString(“NODEID”);

doTask();

}