**定时计划设计文档**

# 编写目的

编写此文档的目的是；让最终用户了解桌面这一模块的功能从而熟练操作这一模块的功能。一般用户：熟练使用该模块的功能；开发者：掌握定时计划的设计思路和实现方法，可以根据开发需求做适当的修改和功能拓展。

# 系统概述

## 概述

定时计划模块的主要功能，使某些用户自定义的任务实现定时执行，并得到执行结果信息。这样会使得需要重复执行的任务需求（如流程催办），不需手动执行，任务会按照设置的定时计划执行。

## 功能



## 面向用户

1. 一般的业务人员。

2. 熟悉WEB开发的开发人员。

## 功能特色

1. 操作简单灵活，最终用户只需根据自己的需求定义自己的定时任务类。
2. 采用Quartz定时框架，该框架会管理相应的定时任务，定时计划表（触发器）无需我们创建新的管理数据表，我们只需创建一系统任务执行日志表记录任务执行情况即可。
3. 采用Quartz spring开源框架集成，优点如下：
   1. 配置文件可以直接配置在spring中，直接使用spring配置的数据源。
   2. 在启动系统时，定时框架自动启动。
4. 支持支持多种定时策略。
5. Quartz，spring开源框架集成过程：
   1. 下载quartz2.0以上的版本。
   2. 执行quartz目录下的创建数据库表脚本。数据库脚本包括各种主流的数据库脚本，ORACLE,MySql，MySql。
   3. 将quartz的jar文件拷贝至项目当中。
   4. 编辑spring的配置文件。

在app-resources.xml中添加如下代码：

/bpm/src/main/resources/conf/app-resources.xml

<!-- QUARTZ 2.0配置的问题 -->

<bean id=*"scheduler"* class=*"org.springframework.scheduling.quartz.SchedulerFactoryBean"*

lazy-init=*"false"* destroy-method=*"destroy"*>

<property name=*"autoStartup"* value=*"true"* />

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

<property name=*"dataSource"* ref=*"dataSource"* />

<property name=*"overwriteExistingJobs"* value=*"true"* />

<property name=*"jobFactory"*>

<bean class=*"org.quartz.simpl.SimpleJobFactory"*></bean>

</property>

<property name=*"quartzProperties"*>

<props>

<!--Job Store -->

<prop key=*"org.quartz.jobStore.driverDelegateClass"*>org.quartz.impl.jdbcjobstore.StdJDBCDelegate</prop>

<prop key=*"org.quartz.jobStore.class"*>org.quartz.impl.jdbcjobstore.JobStoreTX</prop>

<prop key=*"org.quartz.jobStore.tablePrefix"*>QRTZ\_</prop>

<prop key=*"org.quartz.jobStore.clusterCheckinInterval"*>20000</prop>

<prop key=*"org.quartz.scheduler.instanceId"*>AUTO</prop>

<prop key=*"org.quartz.scheduler.classLoadHelper.class"*>org.quartz.simpl.CascadingClassLoadHelper</prop>

<prop key=*"org.quartz.jobStore.lockHandler.class"*>org.quartz.impl.jdbcjobstore.UpdateLockRowSemaphore</prop>

<prop key=*"org.quartz.jobStore.lockHandler.updateLockRowSQL"*>UPDATE QRTZ\_LOCKS SET LOCK\_NAME = LOCK\_NAME WHERE LOCK\_NAME = ?</prop>

</props>

</property>

</bean>

# 各模块介绍

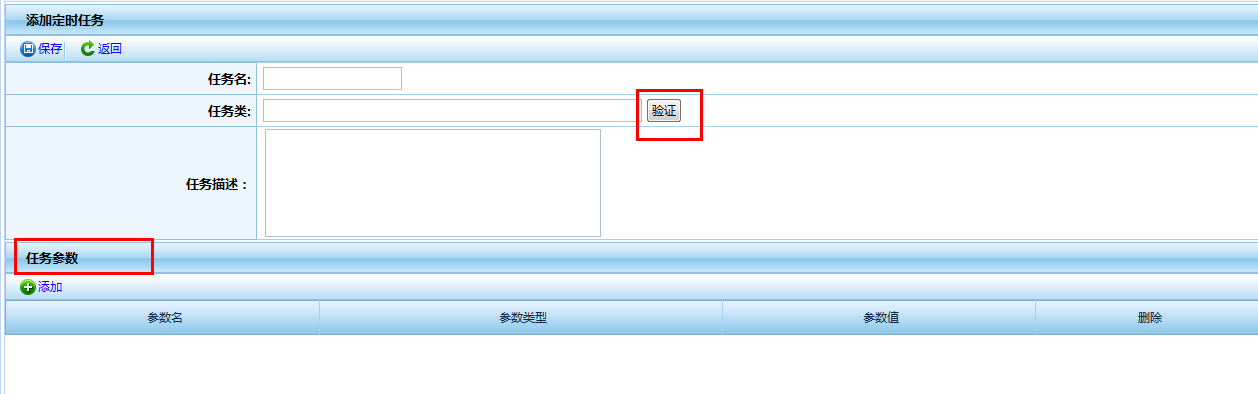
## 定时任务管理模块

### 概述

该模块的主要功能是对用户系统中需要进行定时执行的任务进行统一管理。是整个定时计划模块的主导。

例如：

添加定时任务：



任务类：

一般我们将在com.hotent.platform.job包下定义相关的任务类,该任务类需继承com.hotent.core.scheduler.BaseJob。

* 添加定时任务成功将会在Quartz自带的QRTZ\_JOB\_DETAILS 表中添加任务记录

相关代码：/com/hotent/core/scheduler/SchedulerService.java

**public** **void** addJob(String jobName,String className,String parameterJson,String description) **throws** SchedulerException, ClassNotFoundException{

Class<Job> cls = (Class<Job>)Class.*forName*(className);

JobBuilder jb =JobBuilder.*newJob*(cls);

jb.withIdentity(jobName, *schedGroup*);

**if**(StringUtil.*isNotEmpty*(parameterJson)){

setJobMap(parameterJson,jb);

}

jb.storeDurably();

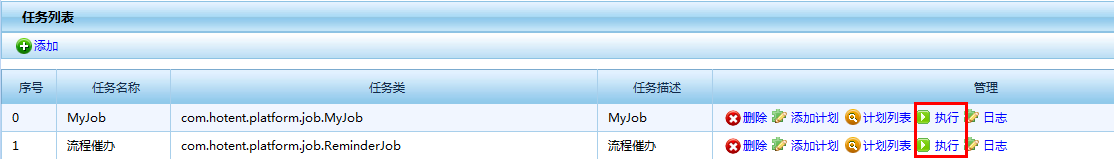
jb.withDescription(description);

JobDetail jobDetail= jb.build();

scheduler.addJob(jobDetail, **true**);

}

可以点击 做任务类的相关验证

定时任务列表：

添加计划，计划列表，日志会在下面模块中说明，这里就不做介绍。

### 用例图



### 用例说明

1. 在这里添加定时任务，删除定时任务，添加任务定时计划，查看任务定时计划都是Quartz对它自带的相应表的增加，删除，查看。

2．每次任务执行完成，我们就用我们系统中的任务执行日志表进行记录。

## 定时计划管理模块

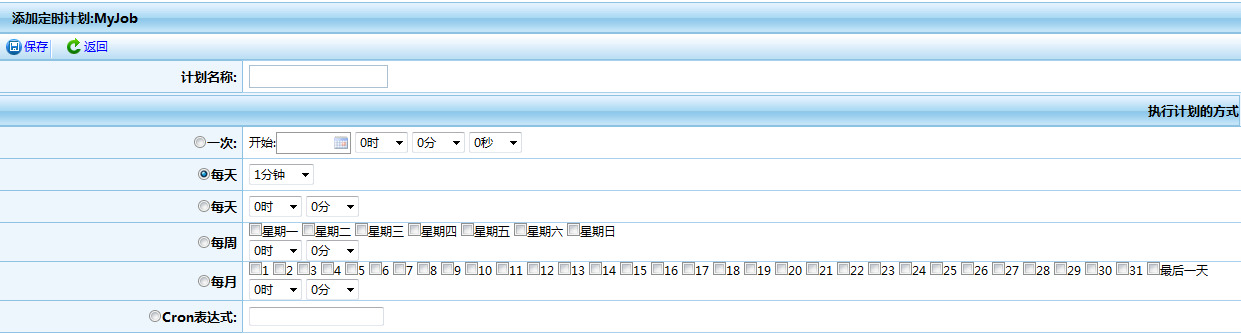
### 概述

该模块的主要功能是对定时任务中的定时情况进行管理，每个任务按照设置的定时计划（触发器）去执行。在这里执行计划主要分为如下方式：

1. xxx时刻执行一次。
2. 每xxx分钟执行。
3. 每天xxx时刻执行。
4. 每周的星期xxx，xxx时刻执行。
5. 每月的xxx日，xxx时刻执行。
6. 最灵活的方式是使用cron表达式。

表达式的写法请参考QUARTZ产品的相关文档。

例如：

 添加定时计划：

相关核心代码：/com/hotent/core/scheduler/SchedulerService.java

**public** **void** addTrigger(String jobName,String trigName, String planJson) **throws** SchedulerException, ParseException{

JobKey jobKey=**new** JobKey(jobName, *schedGroup*);

TriggerBuilder<Trigger> tb= TriggerBuilder.*newTrigger*();

tb.withIdentity(trigName,*schedGroup*);

//tb.withDescription(description);

setTrigBuilder(planJson,tb);

tb.forJob(jobKey);

Trigger trig=tb.build();

scheduler.scheduleJob(trig);

}

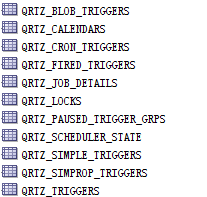
### 用例图



### 用例说明

添加定时计划成功，定时任务会立即执行一次，随后任务会按照设置的定时计划依次执行。这部分的功能都是通过Quartz控制的。

### Quartz相关数据表



## 任务执行日志管理模块

### 概述

该模块的主要功能是，记录定时任务执行情况。

### 产生日志记录流程图



### 数据表



### 涉及文件

|  |  |  |
| --- | --- | --- |
| 文件 | 类别 | 备注 |
| SysJobLog.java | Model | 与表SYS\_JOBLOG映射的实体 |
| SysJobLogDao.java | Dao | 实现对后台任务执行日志的增删改查 |
| SysJobLogService.java | Service | 处理业务逻辑 |
| SchedulerService.java | Service | 定时框架逻辑 |
| SchedulerController.java | Controller | 控制业务与页面之间的调用 |
| jobList.jsp | View | 定时任务列表展现页面 |
| triggerList.jsp | View | 计划列表展现页面 |
| triggerAdd.jsp | View | 添加计划页面 |
| sysJobLogList.jsp | View | 任务执行日志列表 |

其中SchedulerService.java在/com/hotent/core/scheduler下。