

jobschedule

i 摘要
任务调度管理：

- 1、提供任务调度接口，其它插件可以直接使用。
- 2、管理系统中所有调度任务，调度历史等。

使用开发

列表

类名	任务名	执行计划	描述	有效?	上次执行时间	下次执行时间	状态
com.jxtech.jxprojectstat.comm...		0 20 7 1-3 ...	1~3号7:20统计上月项目成本。设...	YES		2015-12-01...	正常
com.jxtech.project.job.Abnorm...		0 0 1 ? * 6	项目异常通知	YES		2015-11-27...	正常
com.jxtech.jxdatasync.commo...		0 5/20 * ? ...	数据同步-增量	YES	2015-11-26...	2015-11-26...	正常
com.jxtech.jxdatasync.commo...		0 0 3 * * ?	数据同步-全量	NO			未启动
com.jxtech.email.common.Ema...		0 5/10 * ? ...	定时发送邮件，每10分钟一次	YES	2015-11-26...	2015-11-26...	正常

主界面

状态: 正常 上次执行时间: 2015-11-26 17:05:00 下次执行时间: 2015-11-26 17:25:00

表达式字段定义

位置	时间域名	允许值	允许的特殊字符
1	秒	0-59	, - * /
2	分钟	0-59	, - * /

i 摘要
演示任务调度的开发，本例以数据同步插件为例说明开发方法。

Java类SQL

实现job接口

```
package com.jxtech.jxdatasync.common;
import org.quartz.Job;
import org.quartz.JobExecutionContext;
import org.quartz.JobExecutionException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.jxtech.jbo.util.JxException;
/**
 * 增量同步
 *
 * @author wmzsoft@gmail.com
 * @date 2014.12
 *
 */
public class SyncIncJob implements Job {
    private static Logger LOG = LoggerFactory.getLogger(SyncIncJob.class);
    @Override
    public void execute(JobExecutionContext context) throws JobExecutionException {
        Sync sync = new Sync();
        try {
            // 增量同步
            sync.syncData(false);
        } catch (JxException e) {
            LOG.error(e.getMessage(), e);
        }
    }
}
```

在插件的SQL脚本中添加脚本

--插件安装之后，自动将Job任务添加到管理器中，以方便管理。

```
delete from JXCRONTASK where CRONTASKNUM='JXDATASYNCINC';
insert into JXCRONTASK (JXCRONTASKID, CHANGEBY, CHANGEDATE, CRONTASKNUM, DESCRIPTION, CRONTASKNAME,
CLASSNAME, APP, ACCESSLEVEL, TASKPLAN, ACTIVE)
values (Jxcrontaskidseq.Nextval, null, sysdate, 'JXDATASYNCINC', '数据同步-增量', '',
'com.jxtech.jxdatasync.common.SyncIncJob', '', '', '0 5/10 * ? * * *', 0);
```