# XXL-JOB 作业调度系统文档

## 一、系统总览

本系统基于 [XXL-JOB](https://www.xuxueli.com/xxl-job/) 进行作业调度执行管理，支持按月/季/年/时间约定调用、作业互相依赖、并行/串行执行、失败重试、时间超限、执行日志等功能，并配合邮件通知和 DAG 依赖可视化实现强大的作业调度能力。

## 二、数据库设计

### 2.1 job\_def 作业定义表

CREATE TABLE job\_def (  
 job\_id BIGINT PRIMARY KEY AUTO\_INCREMENT,  
 job\_code VARCHAR(100) NOT NULL UNIQUE COMMENT '作业编码',  
 job\_name VARCHAR(200) NOT NULL COMMENT '作业名称',  
 proc\_name VARCHAR(200) COMMENT '存储过程名',  
 job\_type VARCHAR(50) COMMENT '作业类型（PROC/SCRIPT/SQL）',  
 schedule\_time TIME, -- 调度时间  
 status VARCHAR(20) DEFAULT 'active', -- 当前状态  
 timeout\_sec INT DEFAULT 1800 COMMENT '超时时间（秒）',  
 retry\_count INT DEFAULT 0 COMMENT '失败重试次数',  
 notify\_email VARCHAR(500) COMMENT '失败通知邮箱',  
 is\_depend TINYINT(1) DEFAULT 1 COMMENT '1 有依赖 0 没有依赖',  
 is\_active TINYINT(1) DEFAULT 1,  
 last\_run\_time DATETIME, -- 上次执行时间  
 next\_run\_time DATETIME, -- 下次执行时间  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,  
 update\_time DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
) COMMENT='作业定义表';

### 2.2 job\_dependency 作业依赖关系表

CREATE TABLE job\_dependency (  
 id BIGINT PRIMARY KEY AUTO\_INCREMENT,  
 job\_code VARCHAR(100) NOT NULL COMMENT '当前作业',  
 depends\_on VARCHAR(100) NOT NULL COMMENT '依赖的作业',  
 dependency\_order INT, -- 依赖顺序  
 UNIQUE KEY uq\_job\_dep (job\_code, depends\_on)  
) COMMENT='作业依赖关系表';

### 2.3 job\_queue 作业队列表

CREATE TABLE job\_queue (  
 queue\_id BIGINT PRIMARY KEY AUTO\_INCREMENT,  
 batch\_no VARCHAR(50) NOT NULL COMMENT '批次号',  
 job\_code VARCHAR(100) NOT NULL COMMENT '作业编码',  
 status VARCHAR(20) DEFAULT 'PENDING' COMMENT '状态',  
 try\_count INT DEFAULT 0 COMMENT '已重试次数',  
 error\_message TEXT,  
 create\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,  
 update\_time DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  
 UNIQUE KEY uq\_batch\_job (batch\_no, job\_code)  
) COMMENT='作业队列表';

### 2.4 job\_execution\_logs 执行日志表

CREATE TABLE job\_execution\_logs (  
 log\_id INT PRIMARY KEY AUTO\_INCREMENT,  
 queue\_id INT, -- 对应 job\_queue.queue\_id  
 start\_time DATETIME,  
 end\_time DATETIME,  
 status VARCHAR(20), -- 执行状态  
 error\_message TEXT,  
 duration INT, -- 执行耗时（秒）  
 notify\_status VARCHAR(20), -- NOTIFIED/UNNOTIFIED  
 retry\_count INT,  
 FOREIGN KEY (queue\_id) REFERENCES job\_queue(queue\_id)  
);

### 2.5 email\_notifications 邮件通知表

CREATE TABLE email\_notifications (  
 id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 recipient\_email VARCHAR(255) NOT NULL,  
 subject VARCHAR(255) NOT NULL,  
 body TEXT NOT NULL,  
 status ENUM('pending', 'sent', 'failed') NOT NULL DEFAULT 'pending',  
 send\_time DATETIME NULL,  
 created\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  
 fail\_reason TEXT NULL,  
 type VARCHAR(50) NULL,  
 retry\_count INT NOT NULL DEFAULT 0,  
 is\_system BOOLEAN NOT NULL DEFAULT TRUE,  
 INDEX idx\_recipient\_email (recipient\_email),  
 INDEX idx\_status (status),  
 INDEX idx\_send\_time (send\_time)  
);

## 三、执行流程设计

**处理流程**

1. xxl-job根据corn定义调度作业Handler  queueDispatcherHandler

2. queueDispatcherHandler 扫描job\_queue，找出所有依赖满足、状态为PENDING的作业

3. 程序可并发执行无依赖作业，串行执行有依赖作业.在这个处理过程中可以有两种方式，

   1.是根据队列中作业情况。符合条件的作业生成一条一次执行的任务交给xxl-job进行处理，

     queueDispatcherHandler只负责扫描job\_queue找出符合条件的任务进行执行。根据作业任务的状态更新日子，发送通知。

   2. 多任务任务的执行都在queueDispatcherHandler 进行处理，xxl-job只做触发

    推荐按照1进行处理

4. 每次作业状态变更写入job\_execution\_logs

5. 作业执行完成发送邮件通知或出发下一级信号

### 3.1 执行系统主体

* XXL-JOB 调度中心：解析调度 cron 表达式、传参、解析 jobHandler
* JobHandler(queueDispatcherHandler)：执行作业队列管理、作业分配
* WorkerHandler ：根据作业类型(PROC/SQL/SCRIPT) 调用对应操作

### 3.2 调度过程

1. XXL-JOB 根据 cron 调用 JobHandler: queueDispatcherHandler
2. queueDispatcherHandler:
   * 解析输入参数(如 batchNo)
   * 根据 job\_def 生成 job\_queue 队列
   * 循环扫描 job\_queue，找出所有 status=’PENDING’且依赖成功的作业
   * 为各个 job\_code 调用 WorkerHandler（形成一次性任务）
3. WorkerHandler 执行作业：
   * PROC: 调用 JDBC CALL 执行存储过程
   * SCRIPT: 执行 Shell/脚本
   * SQL: 执行指定 SQL 语句
4. 执行结果:
   * SUCCESS: 更新 job\_queue 状态 SUCCESS
   * FAILED: 重试或标记失败，记录 logs ，发送邮件
5. 通知:
   * 失败/超时 执行 email\_notifications

### 3.3 每次作业状态变更

* 写入 job\_execution\_logs

### 3.4 重试策略

* job\_queue.try\_count < job\_def.retry\_count
* 可重试则重新调用

## 四、调度任务配置（XXL-JOB 控制台）

| 配置项 | 示例值 |
| --- | --- |
| JobHandler | queueDispatcherHandler |
| Cron | 0 0 2 1 \* ? (每月1日2点) |
| 参数 | {"batchNo":"202407","type":"monthly"} |
| 失败重试 | 3 |
| 超时时间 | 3600 |
| 负责人 | [zxr@aia.com](mailto:zxr@aia.com) |

## 五、系统分工

| 功能 | 调度配置 | 程序处理 |
| --- | --- | --- |
| 调度频率 | Cron表达式 | - |
| 任务参数 | Web界面传入 | 进行解析生成 job\_queue |
| 失败重试 | 配置值 | 重试逻辑 |
| 邮件通知 | 配置负责人 | 失败后发送通知 |
| 依赖关系 | - | 根据 job\_dependency 判断 |
| 队列管理 | - | job\_queue 生成、状态转换 |
| 执行日志 | XXL-JOB 日志 | job\_execution\_logs 输出 |
| 可视化 | - | 生成 DAG 树给前端 |

## 六、延伸能力

* 支持 job\_group/job\_tag 分组
* job\_type 支持 spark/flink 执行类型
* 配合 DAG 图显示依赖链
* 依赖别名和 alias
* job\_template 作业模板设计

如需要开发 queueDispatcherHandler 和 WorkerHandler 的核心代码或系统源码模板，可再继续提供。