

[🏠 / 开发指南 / 微服务手册](#)[👤 芋道源码](#) [📅 2022-04-03](#)

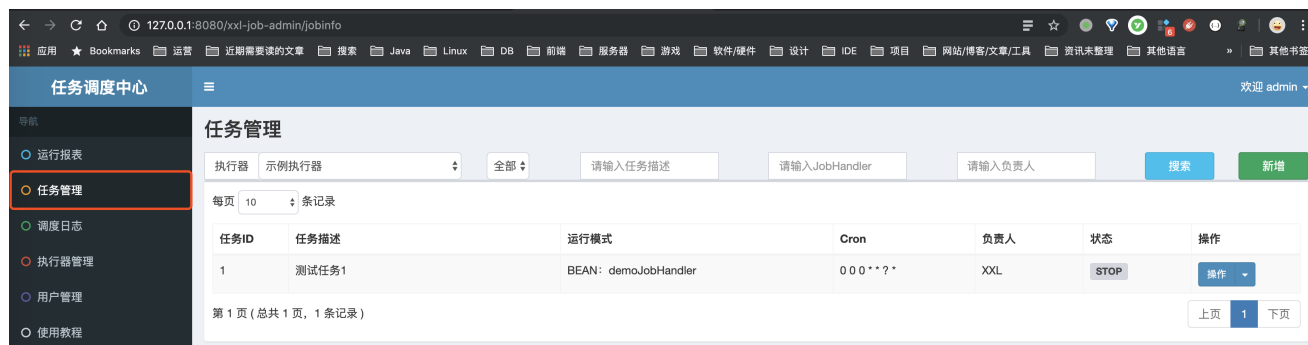
## 🎯 定时任务 XXL Job

定时任务的使用场景主要如下：

- 时间驱动处理场景：每分钟扫描超时支付的订单，活动状态刷新，整点发送优惠券。
- 批量处理数据：按月批量统计报表数据，批量更新短信状态，实时性要求不高。

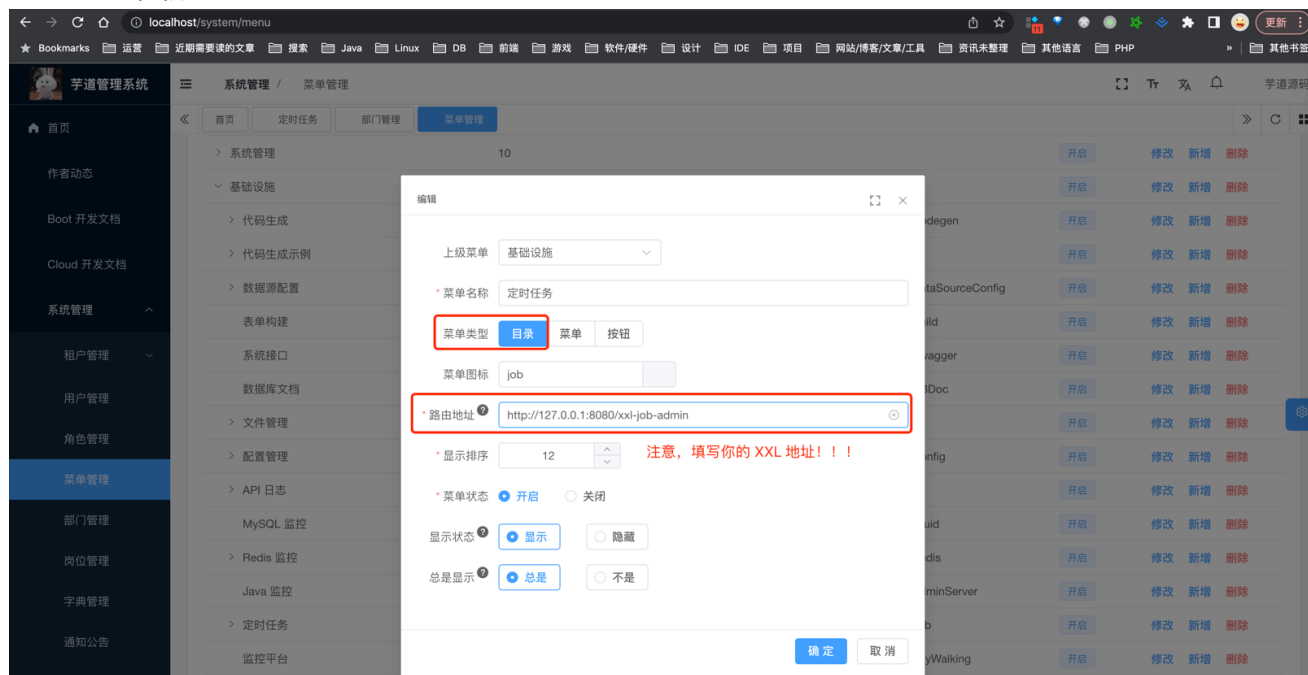
**年度最佳定时任务：每个月初的工资单的推送！！**

项目基于 XXL Job 实现分布式定时任务，支持动态控制任务的添加、修改、开启、暂停、删除、执行一次等操作。



## 1. 如何搭建 XXL Job 调度中心

- ① 参见《芋道 XXL-Job 极简入门》文档的「4. 搭建调度中心」部分。
- ② 搭建完成后，需要修改管理后台的 [基础设施 -> 定时任务] 菜单，指向你的 XXL-Job 地址。如下图所示：



## 2. 如何编写 XXL Job 定时任务

友情提示：以 yudao-module-system 服务为例子。

### 2.1 引入依赖

在 yudao-module-system-biz 模块的 pom.xml 中，引入 yudao-spring-boot-starter-job 技术组件。如下所示：

```
<dependency>
  <groupId>cn.iocoder.cloud</groupId>
  <artifactId>yudao-spring-boot-starter-job</artifactId>
</dependency>
```

该组件基于 XXL Job 框架的封装，实现它的 Spring Boot Starter 配置。

### 2.2 添加配置

① 在 application.yaml 中，添加 xxl.job 配置。如下所示：

```
--- ##### 定时任务相关配置 #####

xxl:
  job:
    executor:
      appname: ${spring.application.name} # 执行器 AppName
      logpath: ${user.home}/logs/xxl-job/${spring.application.name} # 执行器运行日志
      accessToken: default_token # 执行器通讯TOKEN
```

• 注意，xxl.job.accessToken 配置，需要改成你的 XXL Job 调度中心的访问令牌。

② 在 application-local.yaml 中，添加 xxl.job 配置。如下所示：

```
--- ##### 定时任务相关配置 #####

xxl:
  job:
    admin:
```

addresses: http://127.0.0.1:9090/xxl-job-admin # 调度中心部署跟地址

## 2.3 创建 Job 定时任务

参见《[芋道 Spring Boot 定时任务入门](#)》[文章](#)的「5. 快速入门 XXL-JOB」部分。

常用的 Cron 表达式如下：

```
0 0 10,14,16 * * ? 每天上午 10 点，下午 2 点、4 点
0 0/30 9-17 * * ? 朝九晚五工作时间内，每半小时
0 0 12 ? * WED 表示每个星期三中午 12 点
0 0 12 * * ? 每天中午 12 点触发
0 15 10 ? * * 每天上午 10:15 触发
0 15 10 * * ? 每天上午 10:15 触发
0 15 10 * * ? * 每天上午 10:15 触发
0 15 10 * * ? 2005 2005 年的每天上午 10:15 触发
0 * 14 * * ? 在每天下午 2 点到下午 2:59 期间，每 1 分钟触发
0 0/5 14 * * ? 在每天下午 2 点到下午 2:55 期间，每 5 分钟触发
0 0/5 14,18 * * ? 在每天下午 2 点到 2:55 期间和下午 6 点到 6:55 期间，每 5 分钟触发
0 0-5 14 * * ? 在每天下午 2 点到下午 2:05 期间，每 1 分钟触发
0 10,44 14 ? 3 WED 每年三月的星期三的下午 2:10 和 2:44 触发
0 15 10 ? * MON-FRI 周一至周五的上午 10:15 触发
0 15 10 15 * ? 每月15日上午 10:15 触发
0 15 10 L * ? 每月最后一日的上午 10:15 触发
0 15 10 ? * 6L 每月的最后一个星期五上午 10:15 触发
0 15 10 ? * 6L 2002-2005 2002 年至 2005 年，每月的最后一个星期五上午 10:15 触发
0 15 10 ? * 6#3 每月的第三个星期五上午 10:15 触发
```

### 疑问：为什么 Job 查询数据库时，报多租户的错误？

需要声明 `@TenantJob` [注解](#)在 Job 类上，实现并行遍历每个租户，执行定时任务的逻辑。

更多多租户的内容，可见《[开发指南 —— SaaS 多租户](#)》文档。

← [消息队列 RocketMQ](#)

[分布式事务 Seata](#) →



Theme by **Vdoing** | Copyright © 2019-2023 芋道源码 | MIT License