1. **概述**

本文主要说明如何在Spring boot项目中集成定时任务Quartz，适应数据库集群模式，任务相关配置将会持久到数据库中，确保集群部署时，任务只执行一次。

本文是建立在已经配置好了数据源的前提下，并且数据库为MySQL。

quart版本：2.2.3

1. **集成步骤**
   1. **添加相关依赖和配置**

在pom.xml中添加quartz相关依赖。

|  |
| --- |
| <!-- 定时任务 -->  <dependency>  <groupId>org.quartz-scheduler</groupId>  <artifactId>quartz</artifactId>  </dependency>  <dependency>  <groupId>org.quartz-scheduler</groupId>  <artifactId>quartz-jobs</artifactId>  <version>2.2.1</version>  </dependency> |

* 1. **初始配置**

添加config/quartz.properties配置文件，该文件主要描述持久化配置项，该文件为通用配置，通常不需要修订。



该配置源文件：



定义QuartzConfiguration.java加载该配置和进行其他初始设定。

该类中配置了两部分内容，第一部分是，重写了AutowiringSpringBeanJobFactory中createJobInstance()方法，设置在定时任务Job上下文可以自动装载Spring容器Bean。



第二部分，加载上述配置文件，设置数据源和设定自动启动定时任务等。



因为相关配置将持久到数据库中，下载Quartz完整包，解压后可在doc文件夹中获取MySQL建表脚本，总计11张表。

。

官网：<http://www.quartz-scheduler.org/downloads/>

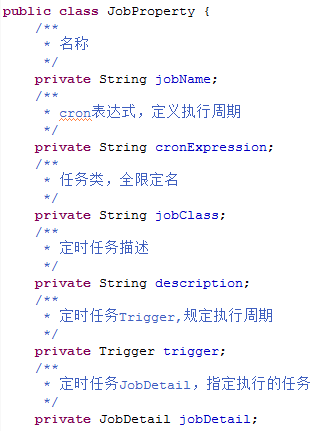
* 1. **定义任务**

在application-profile.yml的application节点配置任务相关属性，这里定义了多个任务，所以需要一个List<配置对象>来接收。

|  |
| --- |
| application:  jobs:  -  job-name: myJob  cron-expression: 0/5 \* \* \* \* ?  job-class: com.nercel.microservice.cyber.job.MyJob  description: myJob for flowerName  -  job-name: yourJob  cron-expression: 0/2 \* \* \* \* ?  job-class: com.nercel.microservice.cyber.job.YourJob  description: yourJob for color |

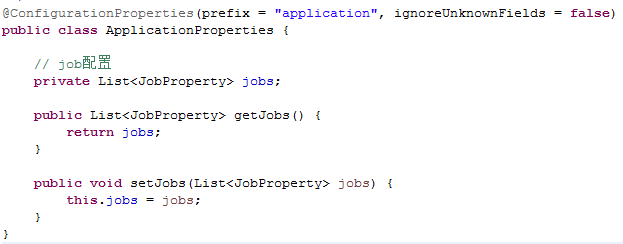
定义配置对象JobProperty接收每个Job的属性，该类中也有两部分内容，接收上述配置和初始化Trigger和JobDetail对象。

注意配置中的键名和类中属性名称的驼峰规则的对应格式。



首先接收application-profile(dev).yml中的配置。在ApplicationConfiguration类（该类默认生成）中定义List<JobProperty>接收配置。

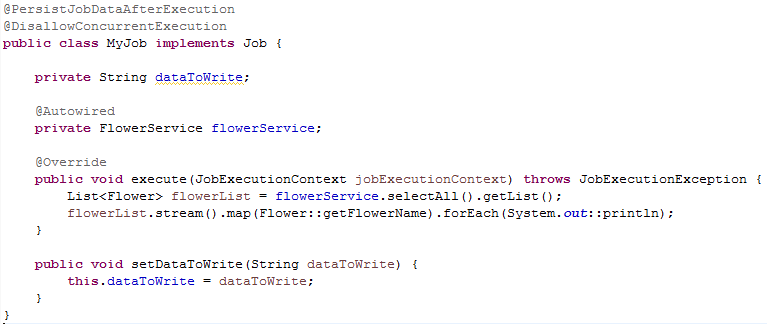
类中jobs自动接收配置中jobs键对应的节点内容。



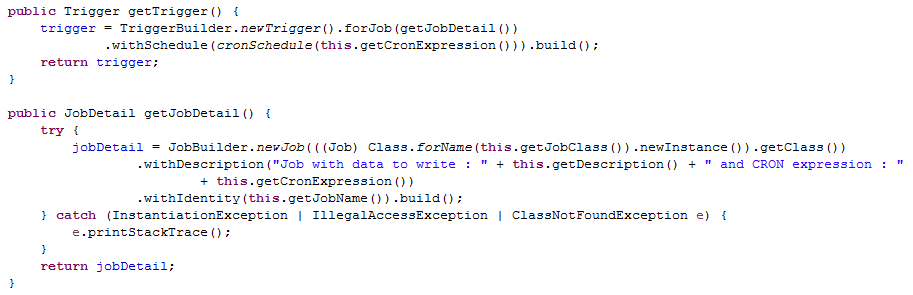
然后初始化JobDetail和Trigger对象，在初始化此两个对象需要指定具体执行任务的类。配置中指定了两个任务MyJob和YourJob。以下是MyJob部分源码，打印数据库中的数据。

其中dataToWrite展示如何给Job中传入数据。

FlowerService因初始配置中设定可以直接加载Spring容器中Bean而直接使用@Autowirer加载。

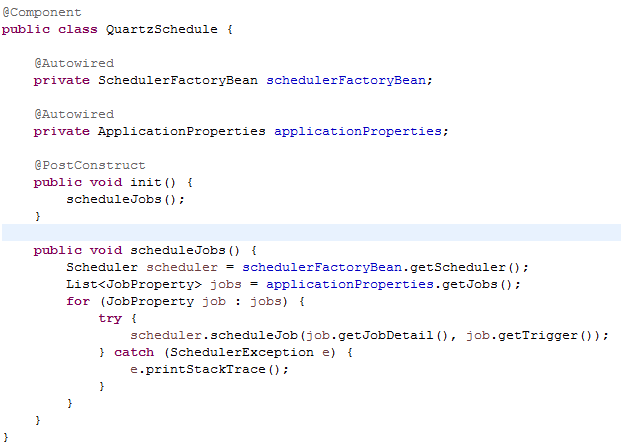


在JobProperty中，利用加载的配置初始化JobDetail和Trigger对象。注意指定JobDetail的唯一标识。



* 1. **将任务加入调度执行**

定义QuartzSchedule类，将所有的任务加入调度。



以上截图部分源文件包。



1. **总结**

以上的配置和实现，可以作为模板使用，也就是增加新的任务时，定义自己的定时任务，如，MyJob和YourJob，然后在application.profile.yml中添加该任务的配置即可。