## spring框架搭建（略：具体看ssm框架搭建）

## 配置quartz

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!-- ~ #{copyright}# -->  <beans xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd*  *http://www.springframework.org/schema/context [http://www.springframework.org/schema/context/spring-context-4.0.xsd"](http://www.springframework.org/schema/context/spring-context-4.0.xsd\">)*[>](http://www.springframework.org/schema/context/spring-context-4.0.xsd\">)  <bean id=*"scheduler"* class=*"org.springframework.scheduling.quartz.SchedulerFactoryBean"/*>  </beans> |

## 建立保存任务的实体类

|  |
| --- |
| **public** **class** SchedulerDto {  **private** String jobId;//任务ID  **private** String jobName;//任务名称  **private** String jobGroup;//任务分组  **private** String jobClassName;//任务类  **private** String triggerName;  **private** String triggerGroup;  **private** **int** jobStatus;//任务状态 0禁用 1启用 2删除  **private** String cronExpression;//任务运行时间表达式  **private** String jobDesc;//任务描述  Getter/setter方法省略  } |

## 建立service接口

|  |
| --- |
| package tt.job.service;  import java.util.List;  import org.quartz.SchedulerException;  import tt.job.dto.SchedulerDto;  /\*\*  \*  \* @name SchedulerService  \* @description  \* @author xupf  \* @version 1.0  \* @date 2017年7月7日  \*/  public interface SchedulerService {  /\*\*  \* 创建一个任务  \* @param schedulerDto  \*/  void createScheduler(SchedulerDto schedulerDto) throws SchedulerException ;    /\*\*  \* 暂停任务  \* @param list  \* @throws SchedulerException  \*/  void pauseJobs(List<SchedulerDto> list) throws SchedulerException;    /\*\*  \* 继续执行任务  \* @param list  \* @throws SchedulerException  \*/  void resumeJobs(List<SchedulerDto> list) throws SchedulerException;    /\*\*  \* 删除Job  \* @param list  \* @throws SchedulerException  \*/  void deleteJobs(List<SchedulerDto> list) throws SchedulerException;    /\*\*  \* 暂停触发器  \* @param list  \* @throws SchedulerException  \*/  void pauseTriggers(List<SchedulerDto> list) throws SchedulerException;    /\*\*  \* 继续执行触发器  \* @param list  \* @throws SchedulerException  \*/  void resumeTriggers(List<SchedulerDto> list) throws SchedulerException;  } |

## service实现类

|  |
| --- |
| **package** tt.job.service.impl;  **import** java.util.List;  **import** org.quartz.CronScheduleBuilder;  **import** org.quartz.CronTrigger;  **import** org.quartz.JobBuilder;  **import** org.quartz.JobDetail;  **import** org.quartz.JobKey;  **import** org.quartz.Scheduler;  **import** org.quartz.SchedulerException;  **import** org.quartz.TriggerBuilder;  **import** org.quartz.TriggerKey;  **import** org.slf4j.Logger;  **import** org.slf4j.LoggerFactory;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.scheduling.quartz.SchedulerFactoryBean;  **import** org.springframework.stereotype.Service;  **import** tt.job.dto.SchedulerDto;  **import** tt.job.service.SchedulerService;  /\*\*  \*  \* **@name** SchedulerServiceImpl  \* **@description**  \* **@author** xupf  \* **@version** 1.0  \* **@date** 2017年7月7日  \*/  @Service  **public** **class** SchedulerServiceImpl **implements** SchedulerService {  **private** Logger logger = LoggerFactory.*getLogger*(SchedulerServiceImpl.**class**);    @Autowired  **private** SchedulerFactoryBean schedulerFactory;    @Override  **public** **void** createScheduler(SchedulerDto schedulerDto) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();    String jobClassName = schedulerDto.getJobClassName();  Class forName = **null**;    **try** {  forName = Class.*forName*(jobClassName);  } **catch** (ClassNotFoundException e) {  **if**(logger.isErrorEnabled()){  logger.error(e.getMessage(),e);  }  }    //创建Jobdetail  JobDetail jobDetail = JobBuilder.  *newJob*(forName).  withIdentity(schedulerDto.getJobName(),schedulerDto.getJobGroup()).  build();    CronTrigger trigger = TriggerBuilder.  *newTrigger*().  withIdentity(schedulerDto.getTriggerName(),schedulerDto.getTriggerGroup()).  forJob(jobDetail).  withSchedule(CronScheduleBuilder.*cronSchedule*(schedulerDto.getCronExpression())).  build();    scheduler.scheduleJob(jobDetail,trigger);    }  @Override  **public** **void** pauseJobs(List<SchedulerDto> list) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();  **for** (SchedulerDto schedulerDto : list) {  scheduler.pauseJob(JobKey.*jobKey*(schedulerDto.getJobName(), schedulerDto.getJobGroup()));  }  }  @Override  **public** **void** resumeJobs(List<SchedulerDto> list) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();  **for** (SchedulerDto schedulerDto : list) {  scheduler.resumeJob(JobKey.*jobKey*(schedulerDto.getJobName(), schedulerDto.getJobGroup()));  }  }  @Override  **public** **void** deleteJobs(List<SchedulerDto> list) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();  **for** (SchedulerDto schedulerDto : list) {  scheduler.deleteJob(JobKey.*jobKey*(schedulerDto.getJobName(), schedulerDto.getJobGroup()));  }  }  @Override  **public** **void** pauseTriggers(List<SchedulerDto> list) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();  **for** (SchedulerDto schedulerDto : list) {  scheduler.pauseTrigger(TriggerKey.*triggerKey*(schedulerDto.getJobName(), schedulerDto.getJobGroup()));  }  }  @Override  **public** **void** resumeTriggers(List<SchedulerDto> list) **throws** SchedulerException {  Scheduler scheduler = schedulerFactory.getScheduler();  **for** (SchedulerDto schedulerDto : list) {  scheduler.resumeTrigger(TriggerKey.*triggerKey*(schedulerDto.getJobName(), schedulerDto.getJobGroup()));  }    }  } |

## 建立controller

|  |
| --- |
| **package** tt.job.controller;  **import** java.util.ArrayList;  **import** java.util.List;  **import** org.quartz.SchedulerException;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Controller;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.ResponseBody;  **import** tt.job.dto.SchedulerDto;  **import** tt.job.service.SchedulerService;  **import** tt.system.controller.BaseController;  /\*\*  \*  \* **@name** SchedulerController  \* **@description**  \* **@author** xupf  \* **@version** 1.0  \* **@date** 2017年7月7日  \*/  @Controller  **public** **class** SchedulerController **extends** BaseController {  @Autowired  **private** SchedulerService schedulerService;    /\*\*  \* 创建任务  \* **@throws** SchedulerException  \*/  @RequestMapping("/tt/job/test")  @ResponseBody  **public** **void** test() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  schedulerService.createScheduler(schedulerDto);  }    /\*\*  \* 暂停任务  \* **@throws** SchedulerException  \*/  @RequestMapping("/tt/job/pauseJobs")  @ResponseBody  **public** **void** pauseJobs() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  List<SchedulerDto> list = **new** ArrayList<>();  list.add(schedulerDto);  schedulerService.pauseJobs(list);  }    /\*\*  \* 继续执行任务  \* **@throws** SchedulerException  \*/  @RequestMapping("/tt/job/resumeJobs")  @ResponseBody  **public** **void** resumeJobs() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  List<SchedulerDto> list = **new** ArrayList<>();  list.add(schedulerDto);  schedulerService.resumeJobs(list);  }    /\*\*  \* 删除job  \* **@throws** SchedulerException  \*/  @RequestMapping("/tt/job/deleteJobs")  @ResponseBody  **public** **void** deleteJobs() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  List<SchedulerDto> list = **new** ArrayList<>();  list.add(schedulerDto);  schedulerService.deleteJobs(list);  }    /\*\*  \* 暂停触发器  \* **@throws** SchedulerException  \*/  @RequestMapping("/tt/job/pauseTriggers")  @ResponseBody  **public** **void** pauseTriggers() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  List<SchedulerDto> list = **new** ArrayList<>();  list.add(schedulerDto);  schedulerService.pauseTriggers(list);  }    @RequestMapping("/tt/job/resumeTriggers")  @ResponseBody  **public** **void** resumeTriggers() **throws** SchedulerException{  SchedulerDto schedulerDto = **new** SchedulerDto();  schedulerDto.setJobName("新建任务");  schedulerDto.setJobClassName("tt.job.test.CronTest");  schedulerDto.setCronExpression("0/10 \* \* \* \* ?");  schedulerDto.setJobGroup("DEFAULT");  schedulerDto.setTriggerGroup("DEFAULTT");  schedulerDto.setTriggerName("tigger");  List<SchedulerDto> list = **new** ArrayList<>();  list.add(schedulerDto);  schedulerService.resumeTriggers(list);  }      } |

## 建立实体类中的classname

|  |
| --- |
| package tt.job.test;  import org.quartz.JobExecutionContext;  import tt.job.AbstractJob;  /\*\*  \*  \* @name CronTest  \* @description  \* @author xupf  \* @version 1.0  \* @date 2017年7月7日  \*/  public class CronTest extends AbstractJob {  @Override  public void safeExecute(JobExecutionContext context) throws Exception {  System.out.println("===========会执行吗？=============");  }    } |

## 编写需要的AbstractJob

|  |
| --- |
| package tt.job;  import org.quartz.Job;  import org.quartz.JobExecutionContext;  import org.quartz.JobExecutionException;  import org.quartz.JobListener;  /\*\*  \*  \* @name AbstractJob  \* @description  \* @author xupf  \* @version 1.0  \* @date 2017年7月7日  \*/  public abstract class AbstractJob implements Job, JobListener {  @Override  public String getName() {  return null;  }  @Override  public void jobExecutionVetoed(JobExecutionContext arg0) {    }  @Override  public void jobToBeExecuted(JobExecutionContext arg0) {    }  @Override  public void jobWasExecuted(JobExecutionContext arg0, JobExecutionException arg1) {    }  @Override  public void execute(JobExecutionContext context) throws JobExecutionException {  try {  safeExecute(context);  } catch (Exception e) {  e.printStackTrace();  }  }  public abstract void safeExecute(JobExecutionContext context) throws Exception;  } |