Quartz是一款开源的定时任务调度框架，本文主要记录一下在工作中使用springboot整合quartz实现定时任务调度管理的用例。内容主要有：**springboot整合quartz相关配置、实现基于simpleTrigger的定时任务、实现基于cronTrigger的定时任务。**

Quartz官网： http://www.quartz-scheduler.org/

1.导入依赖

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.quartz.springboot</groupId>
    <artifactId>quartz-springboot</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.2.RELEASE</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-
thymeleaf</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
```

```
33
34          <dependency>
35              <groupId>org.springframework.boot</groupId>
36              <artifactId>spring-boot-starter-tomcat</artifactId>
37              <scope>provided</scope>
38          </dependency>
39          <dependency>
40              <groupId>org.springframework.boot</groupId>
41              <artifactId>spring-boot-starter-test</artifactId>
42              <scope>test</scope>
43          </dependency>
44
45          <!--Quartz任务调度的包 -->
46          <dependency>
47              <groupId>org.quartz-scheduler</groupId>
48              <artifactId>quartz</artifactId>
49              <version>2.2.1</version>
50          </dependency>
51          <dependency>
52              <groupId>org.quartz-scheduler</groupId>
53              <artifactId>quartz-jobs</artifactId>
54              <version>2.2.1</version>
55          </dependency>
56
57      </dependencies>
58
59      <build>
60          <plugins>
61              <plugin>
62                  <groupId>org.springframework.boot</groupId>
63                  <artifactId>spring-boot-maven-plugin</artifactId>
64              </plugin>
65          </plugins>
66      </build>
67  </project>
```

2.新建SpringBoot启动类

```java
package com.bruce;

import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class QuartzApplication {
    public static void main(String[] args){
        SpringApplication.run(QuartzApplication.class,args);
    }
}
```

3.新建quartz的作业类

```java
public class Job1 implements Job{

    @Override
    public void execute(JobExecutionContext context) throws
JobExecutionException {
        System.out.println("作业1:这里就是写业务逻辑的地方....");
    }

}
```

4.新建quartz的配置类

```java
package com.bruce;

import java.util.Date;

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.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;

import com.bruce.job.Job1;

@Configuration
```

```java
public class QuartzScheduler {

    @Autowired
    private Scheduler scheduler;

    /**
     * 要启动的任务1
     *
     * @param scheduler
     * @throws SchedulerException
     */
    private void startJob1(Scheduler scheduler) throws SchedulerException {
        // 通过JobBuilder构建JobDetail实例，JobDetail规定只能是实现Job接口的实例
        // JobDetail 是具体Job实例
        // 开发的时候你要改变的地方 也只有下面的class ---> Job1.class
        JobDetail jobDetail = JobBuilder.newJob(Job1.class).withIdentity("job1", "group1").build();
        // 基于表达式构建触发器
        // 下面的表达式很重要
        CronScheduleBuilder cronScheduleBuilder = CronScheduleBuilder.cronSchedule("0/5 * * * * ?");
        // CronTrigger表达式触发器 继承于Trigger
        // TriggerBuilder 用于构建触发器实例
        CronTrigger cronTrigger = TriggerBuilder.newTrigger().withIdentity("job1", "group1").withSchedule(cronScheduleBuilder).build();
        scheduler.scheduleJob(jobDetail, cronTrigger);
    }

    /**
     * 01-开始执行所有任务
     *
     * @throws SchedulerException
     */
    public void startJob() throws SchedulerException {
        startJob1(scheduler);
        scheduler.start();
    }

    /**
     * 02-获取Job信息
     */
    public String getJobInfo(String name, String group) throws SchedulerException {
```

```
59          TriggerKey triggerKey = new TriggerKey(name, group);
60          CronTrigger cronTrigger = (CronTrigger)
    scheduler.getTrigger(triggerKey);
61          return String.format("time:%s,state:%s",
    cronTrigger.getCronExpression(),
    scheduler.getTriggerState(triggerKey).name());
62      }
63
64      /**
65       * 03-暂停某个任务
66       */
67      public void pauseJob(String name, String group) throws
    SchedulerException {
68          JobKey jobKey = new JobKey(name, group);
69          JobDetail jobDetail = scheduler.getJobDetail(jobKey);
70          if (jobDetail == null)
71              return;
72          scheduler.pauseJob(jobKey);
73      }
74
75      /**
76       * 04-恢复所有任务
77       *
78       * @throws SchedulerException
79       */
80      public void resumeAllJob() throws SchedulerException {
81          scheduler.resumeAll();
82      }
83
84      /**
85       * 05-修改某个任务的执行时间
86       */
87      public boolean modifyJob(String name, String group, String
    time) throws SchedulerException {
88          Date date = null;
89          TriggerKey triggerKey = new TriggerKey(name, group);
90          CronTrigger cronTrigger = (CronTrigger)
    scheduler.getTrigger(triggerKey);
91          String oldTime = cronTrigger.getCronExpression();
92          if (!oldTime.equalsIgnoreCase(time)) {
93              CronScheduleBuilder cronScheduleBuilder =
    CronScheduleBuilder.cronSchedule(time);
94              CronTrigger trigger =
    TriggerBuilder.newTrigger().withIdentity(name,
    group).withSchedule(cronScheduleBuilder).build();
95              date = scheduler.rescheduleJob(triggerKey, trigger);
96          }
```

```
97            return date != null;
98        }
99
100       /**
101        * 06-删除某个任务
102        */
103       public void deleteJob(String name, String group) throws
      SchedulerException {
104           JobKey jobKey = new JobKey(name, group);
105           JobDetail jobDetail = scheduler.getJobDetail(jobKey);
106           if (jobDetail == null)
107               return;
108           scheduler.deleteJob(jobKey);
109       }
110
111       /**
112        * 07-暂停所有任务
113        * @throws SchedulerException
114        */
115       public void pauseAllJob() throws SchedulerException {
116           scheduler.pauseAll();
117       }
118   }
```

5.新建quartz的监听器类

```
1   package com.bruce;
2
3   import org.quartz.Scheduler;
4   import org.quartz.SchedulerException;
5   import org.quartz.SchedulerFactory;
6   import org.quartz.impl.StdSchedulerFactory;
7   import org.springframework.beans.factory.annotation.Autowired;
8   import org.springframework.context.ApplicationListener;
9   import org.springframework.context.annotation.Bean;
10  import org.springframework.context.annotation.Configuration;
11  import org.springframework.context.event.ContextRefreshedEvent;
12
13  @Configuration
14  public class ApplicationStartQuartzJobListener implements
    ApplicationListener<ContextRefreshedEvent> {
15
16      @Autowired
17      private QuartzScheduler quartzScheduler;
18
19      /**
20       * 初始启动quartz
```

```
     */
    @Override
    public void onApplicationEvent(ContextRefreshedEvent event) {
        try {
            quartzScheduler.startJob();
            System.out.println("任务已经启动...");
        } catch (SchedulerException e) {
            e.printStackTrace();
        }
    }

    /**
     * 初始注入scheduler
     * @return
     * @throws SchedulerException
     */
    @Bean
    public Scheduler scheduler() throws SchedulerException{
        SchedulerFactory schedulerFactoryBean = new
StdSchedulerFactory();
        return schedulerFactoryBean.getScheduler();
    }
}
```

6.控制器类

```
package com.bruce.controller;

import org.quartz.SchedulerException;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.bruce.QuartzScheduler;

@RestController
@RequestMapping("/quartz")
public class QuartzApiController {

    @Autowired
    private QuartzScheduler quartzScheduler;

    @RequestMapping("/start")
    public void startQuartzJob() {
        try {
            quartzScheduler.startJob();
        } catch (SchedulerException e) {
```

```
22              e.printStackTrace();
23          }
24      }
25
26      @RequestMapping("/info")
27      public String getQuartzJob(String name, String group) {
28          String info = null;
29          try {
30              info = quartzScheduler.getJobInfo(name, group);
31          } catch (SchedulerException e) {
32              e.printStackTrace();
33          }
34          return info;
35      }
36
37      @RequestMapping("/modify")
38      public boolean modifyQuartzJob(String name, String group,
    String time) {
39          boolean flag = true;
40          try {
41              flag = quartzScheduler.modifyJob(name, group, time);
42          } catch (SchedulerException e) {
43              e.printStackTrace();
44          }
45          return flag;
46      }
47
48      @RequestMapping(value = "/pause")
49      public void pauseQuartzJob(String name, String group) {
50          try {
51              quartzScheduler.pauseJob(name, group);
52          } catch (SchedulerException e) {
53              e.printStackTrace();
54          }
55      }
56
57      @RequestMapping(value = "/pauseAll")
58      public void pauseAllQuartzJob() {
59          try {
60              quartzScheduler.pauseAllJob();
61          } catch (SchedulerException e) {
62              e.printStackTrace();
63          }
64      }
65
66      @RequestMapping(value = "/delete")
67      public void deleteJob(String name, String group) {
```

```
68          try {
69              quartzScheduler.deleteJob(name, group);
70          } catch (SchedulerException e) {
71              e.printStackTrace();
72          }
73      }
74
75      @RequestMapping(value = "/resumeAllJob")
76      public void resumeAllJob() {
77          try {
78              quartzScheduler.resumeAllJob();
79          } catch (SchedulerException e) {
80              e.printStackTrace();
81          }
82      }
83  }
```