

JavaNoob

Never Stand Still

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

SpringBoot之旅 -- 定时任务两种(Spring Schedule 与 Quartz 整合)实现

相关文章

[Spring Boot 相关文章目录](#)

前言

最近在项目中使用到定时任务，之前一直都是使用Quartz 来实现，最近看Spring 基础发现其实Spring 提供 Spring Schedule 可以帮助我们实现简单的定时任务功能。

下面说一下两种方式在Spring Boot 项目中的使用。

Spring Schedule 实现定时任务

Spring Schedule 实现定时任务有两种方式 1. 使用XML配置定时任务， 2. 使用 @Scheduled 注解。 因为是Spring Boot 项目 可能尽量避免使用XML配置的形式，主要说注解的形式。

Spring Schedule 提供三种形式的定时任务：

固定等待时间 @Scheduled(fixedDelay = 时间间隔)

```
@Component
public class ScheduleJobs {
    public final static long SECOND = 1 * 1000;
    FastDateFormat fdf = FastDateFormat.getInstance("yyyy-MM-dd HH:mm:ss");

    @Scheduled(fixedDelay = SECOND * 2)
    public void fixedDelayJob() throws InterruptedException {
        TimeUnit.SECONDS.sleep(2);
        System.out.println("[FixedDelayJob Execute]+"fdf.format(new Date()));
    }
}
```

固定间隔时间 @Scheduled(fixedRate = 时间间隔)

```
@Component
public class ScheduleJobs {
    public final static long SECOND = 1 * 1000;
    FastDateFormat fdf = FastDateFormat.getInstance("yyyy-MM-dd HH:mm:ss");

    @Scheduled(fixedRate = SECOND * 4)
    public void fixedRateJob() {
        System.out.println("[FixedRateJob Execute]+"fdf.format(new Date()));
    }
}
```

Corn表达式 @Scheduled(cron = Corn表达式)

```
@Component
public class ScheduleJobs {
    public final static long SECOND = 1 * 1000;
    FastDateFormat fdf = FastDateFormat.getInstance("yyyy-MM-dd HH:mm:ss");

    @Scheduled(cron = "0/4 * * * * ?")
    public void cronJob() {
        System.out.println("[CronJob Execute]+"fdf.format(new Date()));
    }
}
```

坐标：北

昵称：Java
园龄：11
粉丝：8
关注：18
[+加关注](#)

最新随笔

1. Lombok
2. SpringB
3. SpringB
4. JUC学习
5. Spring
6. Dubbo
7. 一个字
8. JUC学习
9. Google
10. Google

我的标签

java(9)
Dubbo(5)
SOA(4)
juc(3)
guava(2)
quartz(1)
Elasticsearch(1)
spring(1)
spring boot
开源项目(1)
[更多](#)

随笔分类

Dubbo(5)
Elastic Sta
IntelliJ IDE
Java基础(4)
JUC学习(4)
Spring(1)
Spring Boc
开源框架(1)

随笔档案

2017年4月
2017年3月
2017年2月
2017年1月
2016年12月

阅读排行

1. SpringB
2. SpringB
3. 一个字
4. Lombok
5. Dubbo

Spring Boot 整合 Quartz 实现定时任务

添加Maven依赖

```
<dependency>
  <groupId>org.quartz-scheduler</groupId>
  <artifactId>quartz</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context-support</artifactId>
</dependency>
```

Spring Boot 整合 Quartz

Spring 项目整合 Quartz 主要依靠添加 SchedulerFactoryBean 这个 FactoryBean ，所以在maven 依赖中添加 spring-context-support

首先添加 QuartzConfig 类 来声明相关Bean

```
@Configuration
public class QuartzConfig {
    @Autowired
    private SpringJobFactory springJobFactory;

    @Bean
    public SchedulerFactoryBean schedulerFactoryBean() {
        SchedulerFactoryBean schedulerFactoryBean = new SchedulerFactoryBean();
        schedulerFactoryBean.setJobFactory(springJobFactory);
        return schedulerFactoryBean;
    }

    @Bean
    public Scheduler scheduler() {
        return schedulerFactoryBean().getScheduler();
    }
}
```

这里我们需要注意 我注入了一个 自定义的JobFactory ，然后 把其设置为SchedulerFactoryBean 的 JobFactory。其目的是因为我在具体的Job 中 需要Spring 注入一些Service。

所以我们要自定义一个jobfactory， 让其在具体job 类实例化时 使用Spring 的API 来进行依赖注入。

SpringJobFactory 具体实现：

```
@Component
public class SpringJobFactory extends AdaptableJobFactory {

    @Autowired
    private AutowireCapableBeanFactory capableBeanFactory;

    @Override
    protected Object createJobInstance(TriggerFiredBundle bundle) throws Exception {
        Object jobInstance = super.createJobInstance(bundle);
        capableBeanFactory.autowireBean(jobInstance);
        return jobInstance;
    }
}
```

具体使用 (摘自项目代码)：

```
@Service
public class QuartzEventServiceImpl implements QuartzEventService {
    private static final String JOB_GROUP = "event_job_group";
    private static final String TRIGGER_GROUP = "event_trigger_group";
    @Autowired
    private Scheduler scheduler;

    @Override
    public void addQuartz(Event event) throws SchedulerException {
        JSONObject eventData = JSONObject.parseObject(event.getEventData());
        Date triggerDate = eventData.getDate("date");
        JobDetail job = JobBuilder.newJob(EventJob.class).withIdentity(event.getId().toString(),
```

```
JOB_GROUP).usingJobData(buildJobDateMap(event)).build();
    Trigger trigger = TriggerBuilder.newTrigger().withIdentity(event.getId().toString(),
    TRIGGER_GROUP).startAt(triggerDate).build();
    scheduler.scheduleJob(job, trigger);
}
```

分类: [Spring Boot](#)

标签: [spring](#), [spring boot](#), [quartz](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

JavaNoob

关注 - 18

粉丝 - 8

[+加关注](#)

« 上一篇: [SpringBoot之旅 -- SpringBoot 项目健康检查与监控](#)

» 下一篇: [Lombok : 让你写 Java代码像C#一样爽](#)

1

0

posted @ 2017-03-12 21:43 [JavaNoob](#) 阅读(4862) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻:

- [PC玩家疯狂抛弃Win10：Win7惊天反弹](#)
 - [谷歌无人车首次公开测试，但结果令人担忧](#)
 - [人类进入太空后：零重力下大脑位置居然上移](#)
 - [美的与天猫就“暂停双11空调预售”联合辟谣：已恢复销售](#)
 - [iPhone X专属新铃声《Reflection》试听下载](#)
- » [更多新闻...](#)

最新知识库文章:

- [NASA的10条代码编写原则](#)
 - [为什么你参加了那么多培训，却依然表现平平？](#)
 - [写给初学前端工程师的一封信](#)
 - [实用VPC虚拟私有云设计原则](#)
 - [如何阅读计算机科学类的书](#)
- » [更多知识库文章...](#)

Copyright ©2017 JavaNoob

[站长统计](#) | [今日IP\[31\]](#) | [今日PV\[38\]](#) | [昨日IP\[87\]](#) | [昨日PV\[107\]](#) | [当前在线\[1\]](#)