

فی الـ @Scheduled

1. تفعیل تطبیق فی الـ @Scheduled

فی الـ `@EnableScheduling` تفعیل تطبیق فی الـ @Scheduled ذلک یتم بـ `@EnableScheduling`. اخاص الـ @Scheduled علیک الـ @EnableScheduling لاستخدام التکوین.

```
import org.springframework.context.annotation.Configuration;
import org.springframework.scheduling.annotation.EnableScheduling;

@Configuration
@EnableScheduling
public class SchedulerConfig {
    //
}
```

المحده. لـ `@Scheduled` بـ `@Scheduled` تم الـ `@Scheduled` اتفاقیه وفقاً لـ `@Scheduled` بـ `@Scheduled` يوضح هـ `@Scheduled`.

2. إنشاء مهمة تجديدها

مثـ `@Scheduled` بـ `@Service` أو `@Component` مثـ `@Component` أي فـ `@Component` طريقة تعريف مـ `@Scheduled`.

```
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@Component
public class MyScheduledTasks {

    //      5
    @Scheduled(fixedRate = 5000)
    public void performTask() {
        System.out.println("      : " + System.currentTimeMillis());
    }
}
```

الـ `@Scheduled` بـ `@Component` من الـ `@Component` الـ `@Scheduled` `fixedRate = 5000` مـ `@Scheduled` هذا فـ `@Component` 5000 ثوانیة. مـ `@Scheduled` 5000 ثوانیة.

الجدولة خيارات أنواع 3.

المهمة: تنفيذ يجلب متى لتحديد طرق عدة `@Scheduled` يوفر

ثابت معدل

المهمة. تنفيذ مدة عن الانظر بغض ثابت، بمعدل المهمة بتتنفيذ يقوم `@Scheduled(fixedRate = 5000)` كل 5 ثوان. مثال:

ثابت تأخير

التالي، بدأية واحد تنفيذ نهائية بين ثابت تأخير مع المهمة بتتنفيذ يقوم `@Scheduled(fixedDelay = 5000)`. المهمة انتهاء بعد ثوان 5 مثال:

طبعي ثبات

صباحاً 9 الـساعة في عمل يوم كل `@Scheduled` مثل تعقيذاً أكثر لموعيده `0 0 * * MON-FRI`. مثال:

أولي تأخير

مدمج `fixedRate` أو `fixedDelay` مع يمكن الأولي. المهمة تنفيذ يؤخر

ثوان 5 كل عمل ثم ثوان، 10 بعد `initialDelay = 10000` مثلاً (@Scheduled).

4. صيغة أساسيات

الصيغة: `0 15 * * ?` سريعاً: مرجع إلىك، `0 15 * * ?` تستخدم كنت إذا يوم. كل صباحاً 10:15 الـساعة في عمل ("?" * * ?

`@Scheduled(cron = "0 15` مثلاً:

الحقل	به المسموح القيم
الثانية	0 59
الدقائق	0 59
الـساعة	0 23
الـشـهـرـيـومـ	1 31
الـشـهـرـ	1 12 أو 0 12
الـأـحـدـ	0 0 أو 7 0
الـأـسـبـوـعـيـومـ	0 0 أو 7 0

زيادة / إنـطاـقـ - قـائـمـةـ ، جـمـيـعـ * مـثـلـ خـاصـةـ رـمـوزـ اـسـتـخـداـمـ يـمـكـنـ

5. الاتعلقات

بالنسبة `spring-boot-starter` تعليق افتراضي بكل مدرجة الجدول ميزة فإن، `Starter` تستخدم كنـت إذا
لديكـ أن من تأكـد، `Starter` غير `Starter` لمـشروع

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>6.1.3</version> >!--
</dependency>
```

6. مثال کامل مع

كامل: مثال إلیک

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Configuration;
import org.springframework.scheduling.annotation.EnableScheduling;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@SpringBootApplication
@EnableScheduling
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}

@Component
class ScheduledTasks {

    @Scheduled(fixedRate = 5000)
    public void runEveryFiveSeconds() {
        System.out.println("      5 : " + System.currentTimeMillis());
    }
}

@Scheduled(cron = "0 */1 * * *")
```

```

public void runEveryMinute() {
    System.out.println(" : " + System.currentTimeMillis());
}

```

٧. اختياري الـ `TaskScheduler`

تکوین یمکن مرتزامن، مهام تنفیذی لی تتحتاج کنت إذا واحد. خیط ذو زمنی جدول `TaskScheduler` استخدمو افتراضی، بالتعلیمات `TaskScheduler` مخصوص زمنی جدول:

```

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.scheduling.concurrent.ThreadPoolTaskScheduler;

@Configuration
public class SchedulerConfig {

    @Bean
    public ThreadPoolTaskScheduler taskScheduler() {
        ThreadPoolTaskScheduler scheduler = new ThreadPoolTaskScheduler();
        scheduler.setPoolSize(10); //
        scheduler.setThreadNamePrefix("my-scheduler-");
        return scheduler;
    }
}

```

مرتزامن. بشکل بالعمل مهام لعدة یسمح هذا

٨. اختبار مهام مجولة

متوقع. هنکما تعامل المهام أن لتأکید الأمامية الواجهة سجلات من وتحققت طبیعته بتشریح قم `MockMvc` بسرعة. النتائج لرؤية الاختبار أثناه ثوان `2` كل `MockMvc` مثل أقصر فترات استخدم