# 【SpringBoot】@Scheduled 定時執行

在Spring Boot中，我□可以使用注□@Scheduled□定□□行方法。□注□可用于□方法□□□需要定期□行的方法，并提供□行□划。

@Scheduled注□支持以下属性：

1. fixedRate：以毫秒□□位的□行速率。
2. fixedDelay：以毫秒□□位的□行延□，即上一次□行□束后的延□□间。
3. initialDelay：第一次□行的延□□间，以毫秒□□位。
4. cron：CRON表达式，用于更高□的定□□划。

下面是一个□□的示例，□示例演示如何使用@Scheduled注□在Spring Boot中定期□行方法：

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

@Component
public class MyTask {

@Scheduled(fixedRate = 5000)
public void printMessage() {
System.out.println("Hello, world!");
}
}
```

在上述示例中，我□定义了一个名□MyTask的□件，其中包含一个使用@Scheduled注□□□的printMessage方法。此方法□在每隔5秒钟输出一条消息"Hello, world!"。

我□□可以使用fixedDelay属性□延□□行。例如，如果我□□fixedDelay设置□10000，那么□方法□在上一次□行□束后等待10秒才能□始下一次□行。

@Scheduled注□□支持CRON表达式，□使得我□可以更高□地安排任务。例如，如果我□想要在每天早上6点□行任务，我□可以使用以下CRON表达式：

```
0 0 6 * * *
```

在此示例中，我□使用六个星号□指定秒，分，□，日，月和星期几。因此，此表达式□在每天早上6点□行任务。

---

# 註解停用

如果你想□□停用@Scheduled注□□□的方法，你可以□□地注□掉它，或者□其□□□禁用，以便稍后重新启用。

要注□掉方法，□□其注□掉：

```
// @Scheduled(fixedRate = 5000)
public void printMessage() {
    System.out.println("Hello, world!");
}
```

要禁用方法，您可以使用@Scheduled注□的enabled属性：

```
// @Scheduled(fixedRate = 5000)
public void printMessage() {
    System.out.println("Hello, world!");
}
```

在□个例子中，我□使用@Scheduled注□的enabled属性□printMessage方法□□□已禁用。要重新启用□方法，只需□enabled属性设置□true即可。

---

# 使用命令停用

是的，您可以使用Spring Boot Actuator的端点□动□启用/禁用定□任务。Spring Boot Actuator是Spring Boot的一个附加模块，提供了很多有用的端点，可以在□用程序运行□□视和管理□用程序。

您需要在□目中添加Actuator依□□：

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

然后，您可以使用`/actuator/scheduledtasks`端点□□查和管理定□任务。要□□停用定□任务，您可以使用□端点□取消□划的任务。例如：

```
@Scheduled(fixedRate = 5000, name = "myTask")
public void printMessage() {
    System.out.println("Hello, world!");
}
```

```
$ curl -X POST http://localhost:8080/actuator/scheduledtasks/taskName
```

□□取消□划名□taskName的任务。要重新启用□任务，您可以使用□端点□重新启用任务：

```
$ curl -X DELETE http://localhost:8080/actuator/scheduledtasks/taskName
```

假设您的Spring Boot□用程序正在运行，并且您已经添加了Actuator依□□，那么您可以使用以下curl命令向`/actuator/scheduledtasks`端点发出HTTP GET□求□□取□前□用程序中所有定□任务的列表：

```
curl http://localhost:8080/actuator/scheduledtasks
```

□□返回一个JSON格式的响□，其中包含有□每个定□任务的信息，例如任务名□、任务是否启用、任务的CRON表达式等。您可以使用□响□□查看□用程序中所有的定□任务信息。

```
{
  "cron": {
    "myScheduledTask": {
      "scheduled": true,
      "expression": "0/30 * * * * ?"
    }
  },
  "fixedDelay": {},
  "fixedRate": {}
}
```

---

修訂版本 #1
★由 treeman 建立於 5 ���� 2023 10:19:35
🖉由 treeman 更新於 5 ���� 2023 10:38:31