# **@Scheduled中fixedDelay和fixedRate的区别**

先介绍下Schedule的使用方法  
1.首先新建使用Springboot的Maven项目，引入相关引用如下在pom.xml里添加如下：

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

2.新建MySchedule的Java文件，用@Component来注释该类，定时任务用@Scheduled来注释如下：

@Scheduled(fixedRate=2000)  
public void fixedRate(){System.***out***.println("fixedRate:\t"+new Date());  
}

该代码会定时2秒钟执行一次在这里插入代码片，执行结果如下  
fixedRate: Thu Nov 05 11:05:52 CST 2020

fixedRate: Thu Nov 05 11:05:54 CST 2020

fixedRate: Thu Nov 05 11:05:56 CST 2020

fixedRate: Thu Nov 05 11:05:58 CST 2020

fixedRate: Thu Nov 05 11:06:00 CST 2020

但是@Scheduled中还有一种属性fixedDelay，两者有什么区别，区别就是如果执行任务中有阻塞的情况，fixedDelay执行间隔会加上阻塞的时间代码如下：

@Scheduled(fixedDelay=2000)  
public void fixedDelay(){  
 try  
 {  
 Thread.currentThread().sleep(1000);*//毫秒* }  
 catch(Exception e){}  
 System.***out***.println("fixedDelay:\t"+new Date());  
}

执行结果如下

fixedDelay: Thu Nov 05 11:09:17 CST 2020

fixedDelay: Thu Nov 05 11:09:20 CST 2020

fixedDelay: Thu Nov 05 11:09:23 CST 2020

fixedDelay: Thu Nov 05 11:09:26 CST 2020

fixedDelay: Thu Nov 05 11:09:29 CST 2020

**可以看出后一次执行的时间是fixedDelay的时间加阻塞时间；**  
而fixedRate则是按其规定的时间来执行的，犹如时刻表一般，当阻塞结束后按时刻表来执行，代码如下：

@Scheduled(fixedRate=2000)  
public void fixedRate(){  
 try  
 {  
 Thread.currentThread().sleep(1000);*//毫秒* }  
 catch(Exception e){}  
 System.***out***.println("fixedRate:\t"+new Date());  
}

执行结果如下

fixedRate: Thu Nov 05 11:10:46 CST 2020

fixedRate: Thu Nov 05 11:10:48 CST 2020

fixedRate: Thu Nov 05 11:10:50 CST 2020

fixedRate: Thu Nov 05 11:10:52 CST 2020

fixedRate: Thu Nov 05 11:10:54 CST 2020

**执行结果中时间间隔为fixedRate定的时间间隔**

总结：

fixedRate下一次执行时间是本次开始时间加间隔时间

fixedDelay下一次执行时间是本次结束时间加间隔时间