

第一支排程式

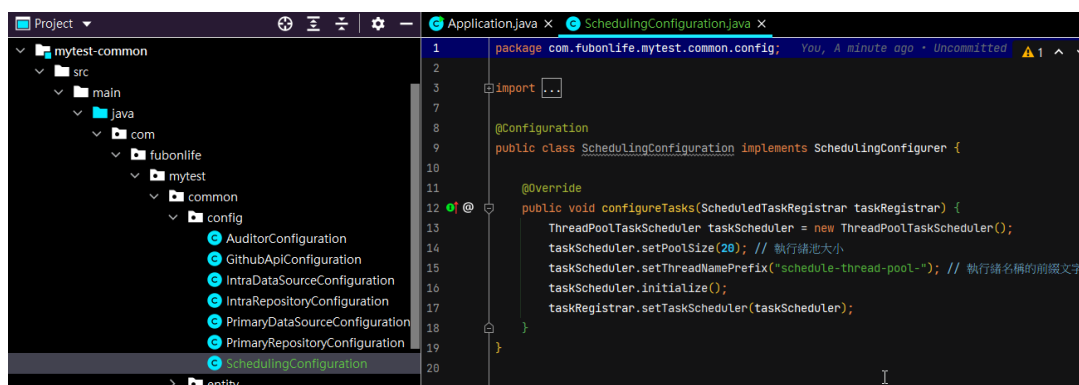
開發流程

步驟一：多執行緒配置

方法 1：Spring Boot 2.1及以後的版本

```
spring:
  task:
    scheduling:
      pool:
        size: 20
      thread-name-prefix: schedule-thread-pool-
```

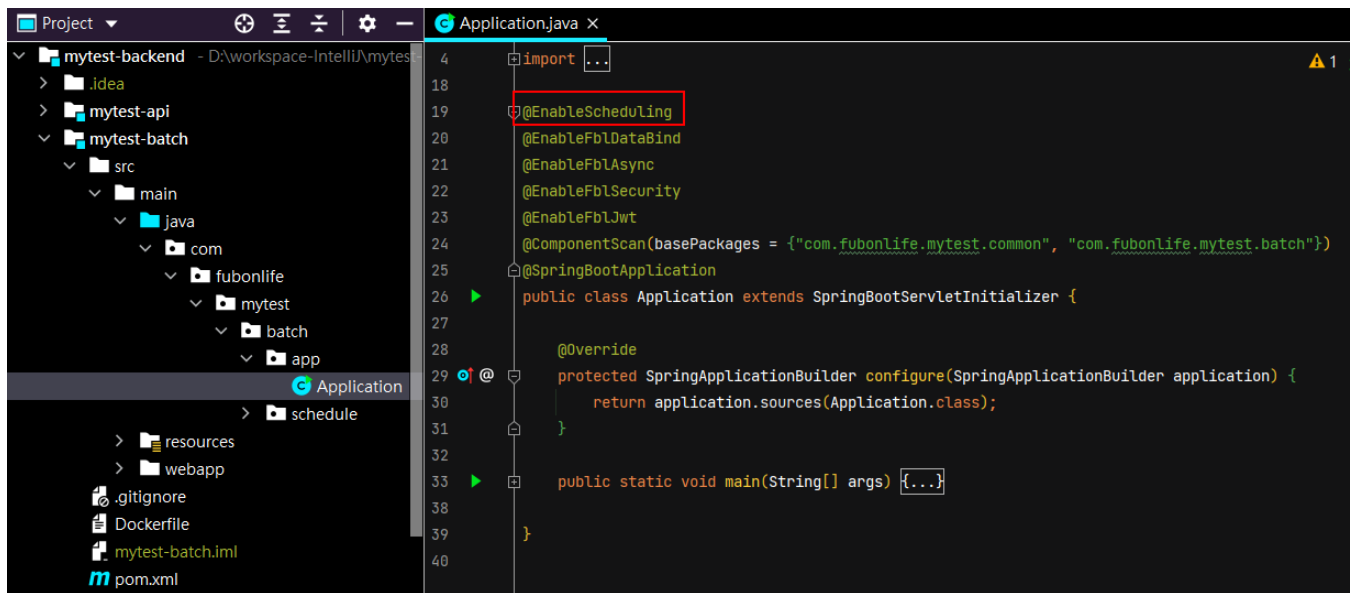
方法 2：Spring Boot 2.0及以前的版本



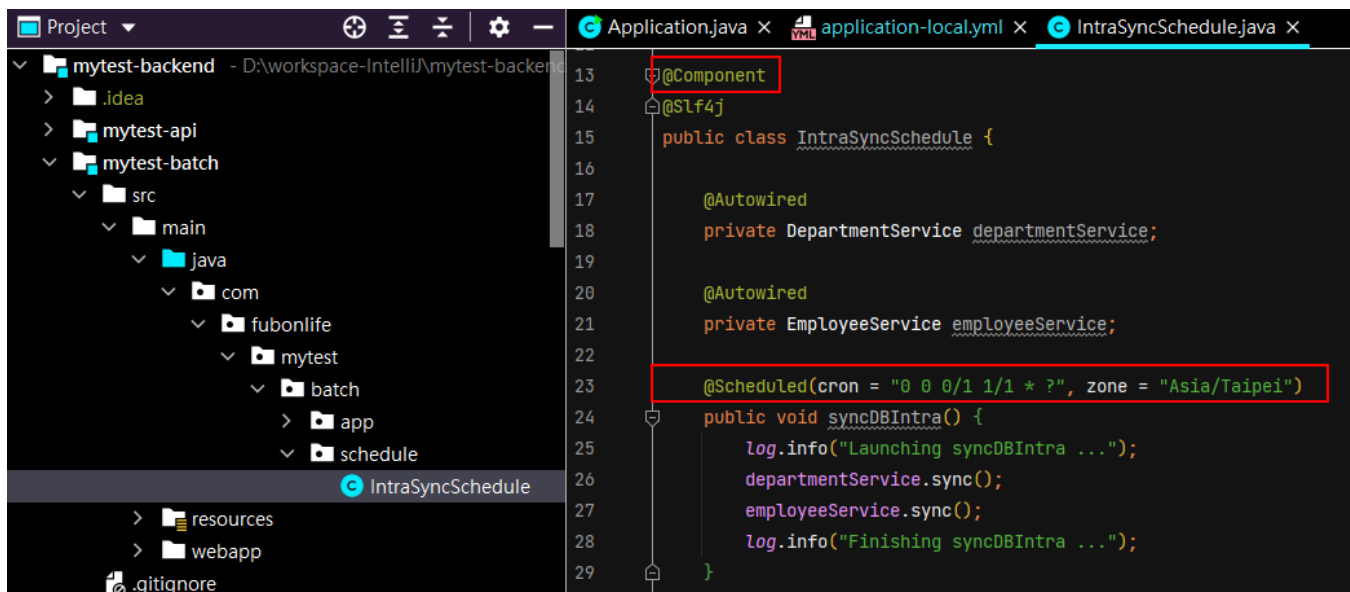
The screenshot shows an IDE with a project structure on the left and the SchedulingConfiguration.java file open on the right. The project structure includes a 'mytest-common' module with a 'src/main/java/com/fubonlife/mytest/common/config' directory. The SchedulingConfiguration.java file is located in this directory. The code in the file is as follows:

```
1 package com.fubonlife.mytest.common.config;
2
3 import ...
4
5 @Configuration
6 public class SchedulingConfiguration implements SchedulingConfigurer {
7
8     @Override
9     public void configureTasks(ScheduledTaskRegistrar taskRegistrar) {
10
11         ThreadPoolTaskScheduler taskScheduler = new ThreadPoolTaskScheduler();
12         taskScheduler.setPoolSize(20); // 執行緒池大小
13         taskScheduler.setThreadNamePrefix("schedule-thread-pool-"); // 執行緒名稱的前綴文字
14         taskScheduler.initialize();
15         taskRegistrar.setTaskScheduler(taskScheduler);
16     }
17 }
18
19
20
```

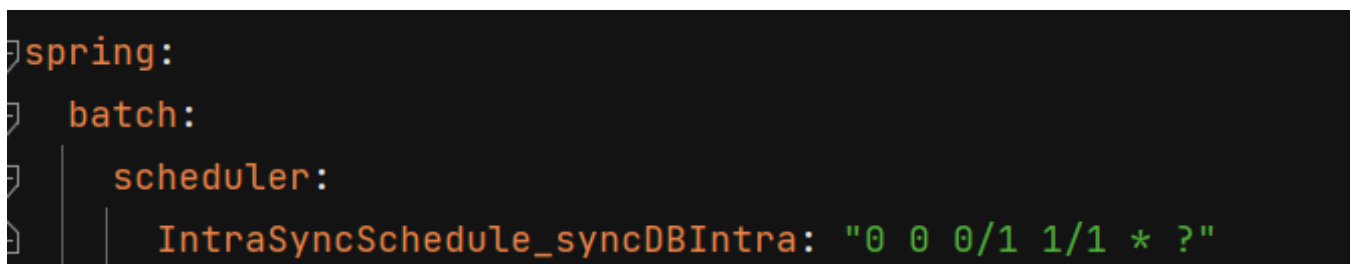
步驟二：在子專案 mytest-batch 的啟動類上增添一標記 @EnableScheduling



步驟三：用 `@Component` 將後端排程式註冊為 `Spring` 容器的 `Bean`，然後在增添 `@Scheduled` 用來標記一個定時任務方法



步驟四：（非必要步驟）將所有定時任務的預設執行時間OR區間，統一集中紀錄在「`application.yml`」檔案內



```

@Scheduled(cron = "${spring.batch.scheduler.IntraSyncSchedule_syncDBIntra}", zone = "Asia/Taipei")
public void syncDBIntra() {
    log.info("Launching syncDBIntra ...");
    departmentService.sync();
    employeeService.sync();
    log.info("Finishing syncDBIntra ...");
}

```

定時任務排程器的種類

cron表達式：是一個字串，指定任務在特定時間執行

```

//
@Scheduled(cron = "0 0 0/1 1/1 * ?")
public void testCron() {
    ...
}

```

fixedDelay：表示上一次任務執行完成後多久再次執行，參數型別為 long，單位 ms

```

// fixedDelay = 60000 60000ms(1) Spring scheduling
@Scheduled(fixedDelay = 60000)
public void testFixDelay() {
    ...
}

```

fixedRate：表示按一定的頻率執行任務，參數型別為 long，單位 ms

```

// fixedRate = 60000 60000ms(1) Spring scheduling
@Scheduled(fixedRate = 60000)
public void testFixedRate() {
    ...
}

```

initialDelay：表示延遲多久後第一次執行任務，參數型別為 long，單位 ms

```

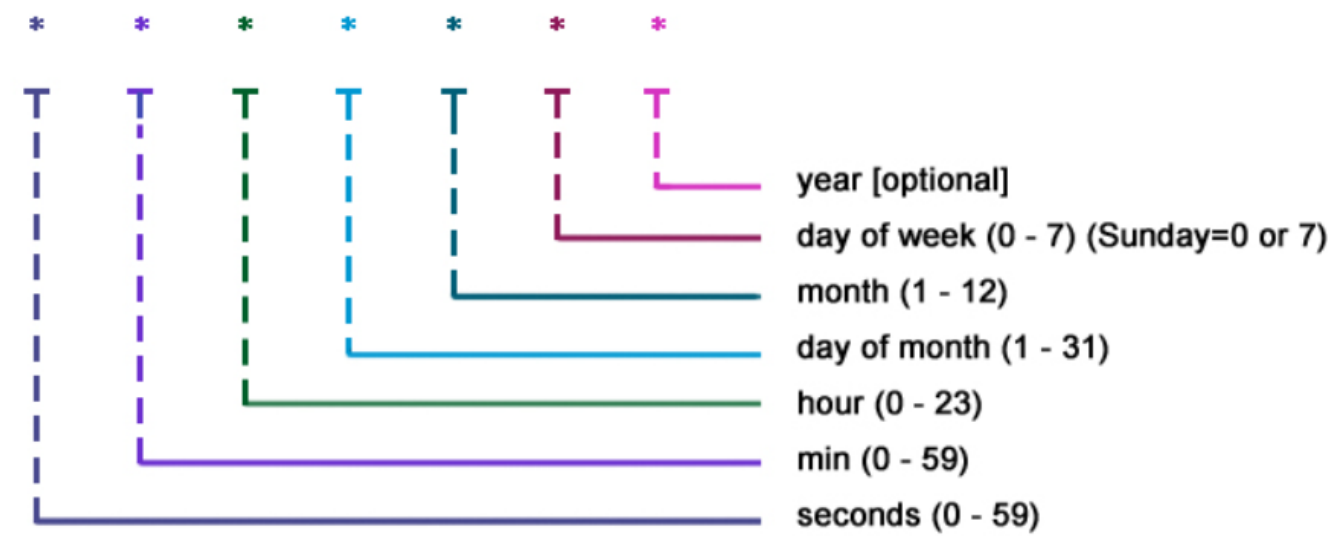
// initialDelay = 180000 180000 (3) , 5000ms(5 )
@Scheduled(initialDelay = 180000, fixedRate = 5000)
public void testInitialDelay() {
    ...
}

```

細談 Cron 表達式

表達式結構

Cron表示式是一個字串，是由空格隔開的 6 或 7 個域組成，每一個域對應一個含義（秒 分 時 每月第幾天 月 星期 年），其中年是可省略的域。



符號功能說明

域	值	支援的符號	功能說明
seconds	0 到 59	, - * /	<div>* : *</div> <div>? : day week</div> <div>?</div> <div>- : 5-20520</div> <div>/ : min5/2052545</div> <div>, : 5,20520</div> <div>L : 1L,</div> <div>W : (),</div> <div>LW :</div> <div># : 1#3</div>
min	0 到 59	, - * /	
hour	0 到 23	, - * /	
day	1 到 31	, - * ? / L W C	
month	1 到 12 JAN - DEC	, - * /	
week	0 到 7 MON - SAT	, - * ? / L C #	

舉例說明

[Cron解析器（網頁版）](#)

0 0/5 * * * ?	每 5 分鐘一次。
0 0 12 * * ?	於每日中午 12:00。
0 15 10 ? * *	於每日上午 10:15。
0 15 10 * * ?	於每日上午 10:15。
0 15 10 * * ? *	於每日上午 10:15。
0 0 12 1/5 * ?	於每月從第一天開始每五天的中午 12:00。
0 11 11 11 11 ?	每個 11 月 11 日上午 11:11。