

主題 4：spring-boot-starter-batch

介紹執行流程

[blocked URL](#)

參考來源：<https://docs.spring.io/spring-batch/docs/current/reference/html/job.html#javaConfig>

介紹 @EnableBatchProcessing [參考]

使用該 Annotation 預設註冊的物件清單

Class 名稱	Bean 名稱	說明
JobRepository	jobRepository	紀錄批次基本資料、Job與Step執行參數資料
JobLauncher	jobLauncher	呼叫 Job 執行的物件
JobRegistry	jobRegistry	註冊 Job 的實作
PlatformTransactionManager	transactionManager	負責 JobRepository 的資料庫交易
JobBuilderFactory	jobBuilders	Job Bean 產生器，主要用於註冊 Job 的設定
StepBuilderFactory	stepBuilders	Step Bean 產生器，主要用於註冊Step的設定

JobRepository 介紹

- 主要用於紀錄批次排程執行的狀況
- 需要建立對應的資料表，資料表說明可[參考]
- 需要建立的資料表 schema 可於 spring-batch-core-4.3.5.jar 的 org.springframework.batch.core 取得

介紹2種方式

- Chunk-oriented Processing
 - Item Readers[官方提供實作清單]
 - Item Processor
 - Item Writers[官方提供實作清單]
 - 時序圖(Sequence Diagram)
[blocked URL](#)
參考來源：<https://docs.spring.io/spring-batch/docs/current/reference/html/step.html#chunkOrientedProcessing>
- TaskletStep
 - 簡易執行批次流程可使用此方式開發

執行 Job 設定

JobLauncher 配置

```
@Scheduled(cron = "0 */3 * * * ?")
public void run() throws JobInstanceAlreadyCompleteException, JobExecutionAlreadyRunningException,
JobParametersInvalidException, JobRestartException {
    log.info(" chuck ");

    JobParameters args = new JobParametersBuilder()
        // key - -
        // parameter - -
        // identifying - - , true,
        //                                     false, parameter org.springframework.batch.core.
repository.JobInstanceAlreadyCompleteException
        // .addString(String key, String parameter, boolean identifying)
        .addString("scheduleName", "", false)
        .toJobParameters();
    jobLauncher.run(this.job(), args);
    log.info(" chuck ");
}
```

Builder Factory 說明

StepBuilderFactory 參數設定

```
@Bean
public Step step() {
    log.info(" chuck ");
    return this.stepBuilderFactory.get("export-data")
        // Job COMPLETE,
        .allowStartIfComplete(true)
        .transactionManager(transactionManager)
        .<Employee>chunk(10)
        .reader(this.itemReader())
        .writer(this.itemWriter())
        // ,
        .faultTolerant()
        // exception
        .skip(FlatFileParseException.class)
        .skipLimit(10)
        // exception
        .retry(DeadlockLoserDataAccessException.class)
        .retryLimit(3)
        // exception rollback
        .noRollback(ValidationException.class)
        .build();
}
```

範例程式參考

- [Chunk-oriented Processing](#) [[參考](#)]
- [TaskletStep](#) [[參考](#)]

介紹佈屬 Jboss runtime 失敗問題

運行環境：Jboss EAP 7.3

因 Jboss 內有提供預設的共用 jar，但由於 FBL 專案預設的 Spring batch 版本需要的 Jackson core 跟 Jboss 預設提供的版本不符合，因此需排除 Jboss 預設的 jar，改使用 AP 專案內的相依，需再專案路徑 mytest-batch/src/main/webapp/WEB-INF 建立 jboss-deployment-structure.xml，設定檔如下：

jboss-deployment-structure.xml

```
<jboss-deployment-structure>
<deployment>
  <!-- Exclusions allow you to prevent the server from automatically adding some dependencies -->
  <exclusions>
    <!-- jackson pom dependencies -->
    <module name="com.fasterxml.jackson.core.jackson-core" />
    <module name="com.fasterxml.jackson.core.jackson-annotations" />
    <module name="com.fasterxml.jackson.core.jackson-databind" />
    <module name="com.fasterxml.jackson.datatype.jackson-datatype-jdk8" />
    <module name="com.fasterxml.jackson.datatype.jackson-datatype-jsr310" />
    <module name="com.fasterxml.jackson.jaxrs.jackson-jaxrs-json-provider" />
    <module name="org.jboss.resteasy.resteasy-jackson2-provider" />
  </exclusions>
</deployment>
</jboss-deployment-structure>
```

相關資料參考來源：

- [jboss-deployment-structure.xml](#)
- [Jboss 預設 Jar 清單](#)