繼前篇說明

java ee7 batch application 分享.docx

很多人可能會好奇說

如何監控執行結果

其實可以透過幾種方式如下

1. 去glassfish console看(前篇介紹)
2. 透過程式去跟glassfish要job執行結果清單

(各ap只能跟自己的glassfish要結果)

1. 執行job時就把結果放到某個DB去

( 這邊可以做成集中化監控 一次看多個ap

,不過看來可能要在第二點之後 拉到某個集中處理的DB

)

上面所述第一個有個問題是 不是所有人都有glassfish正式環境帳號/密碼

因此必須要想辦法用上面第2,第3點達成

以下內容講的是如何取得 glassfish執行結果

並想辦法顯示出來

1. 取得job operator

JobOperator jobOperator = BatchRuntime.getJobOperator();

1. jobOperator可以找出某個特定的job所有紀錄
   * 下面case是去找名稱叫simpleDataStrJob的job,然後能拿的全要



List<JobInstance> jobInstanceList = //

jobOperator.getJobInstances("simpleDataStrJob"//

,//

0//

, Integer.MAX\_VALUE - 1);

* 當然我也可以要某個特定的instance by id

如果你知道id的話



1. 取得jobExecution

這邊就是指每次job執行的表層結果

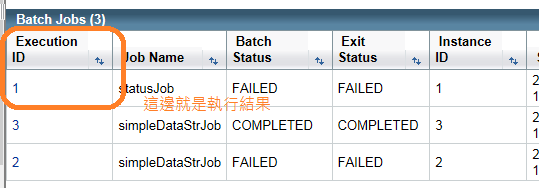
List<JobExecution> jobExeList = jobOperator.getJobExecutions(jobInstance);

正常如果成功執行 一個instance 應該一個execution

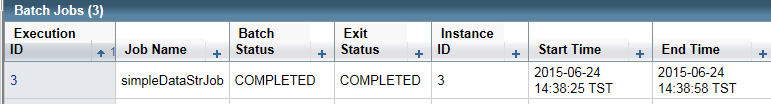
(List 裡面只有一個 情況居多…..)

但如果錯誤 但指定instance id重跑

可能就會是 one instance vs multi execution



1. Job表層結果的各項屬性取得如下



下面就是根據job instance拿到的jobExecution取得對應屬性

//instance Id

Long instanceId = jobInstance.getInstanceId();

//執行ID

Long exeId = jobExecution.getExecutionId();

//job 名稱

String jobName = jobExecution.getJobName();

//job status enum 執行狀態

//STARTING, STARTED, STOPPING, STOPPED, FAILED, COMPLETED, ABANDONED

BatchStatus jobStatus = jobExecution.getBatchStatus();

//執行完畢狀態

String exitStatus = jobExecution.getExitStatus();

//job執行開始時間

Date startTime = jobExecution.getStartTime();

//job執行完畢時間

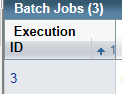
Date endTime = jobExecution.getEndTime();

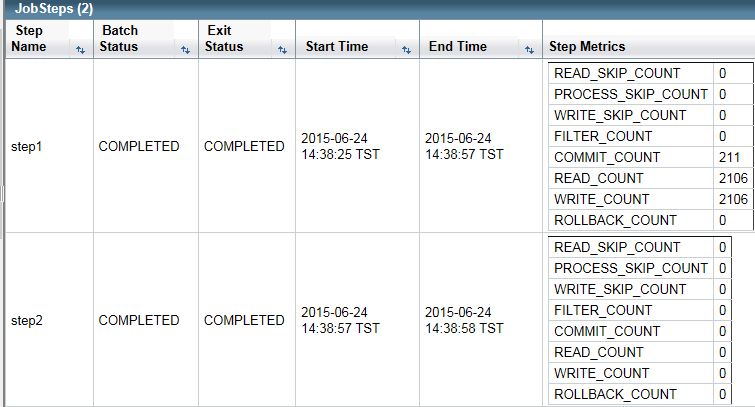
1. 表層的資料 有些人可能會覺得不夠 所以會想往下面看明細

下面就是明細要如何取

List<StepExecution> stepExeList = jobOperator.getStepExecutions(exeId);

上面黃色的是指execution Id





上圖是說我的job有兩個step

一個job會根據使用者設計 而有多個step 或 最少一個

//步驟名稱

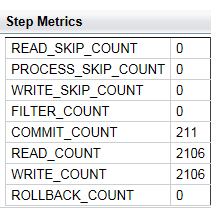
String stepName = stepExecution.getStepName();

logger.debug("stepName=" + stepName);

//

Metric[] metricArray = stepExecution.getMetrics();

下圖是用Metric去接值



//metric 是 enum幾種屬性紀錄如下

//READ\_COUNT, WRITE\_COUNT, COMMIT\_COUNT, ROLLBACK\_COUNT

//, READ\_SKIP\_COUNT, PROCESS\_SKIP\_COUNT, FILTER\_COUNT

// ,WRITE\_SKIP\_COUNT

for (Metric metric : metricArray) {

String metricType = metric.getType().toString();

long metricValue = metric.getValue();

logger.debug("type=" + metricType + " value=" + metricValue);

}

1. 以上如果資訊都可以取得的話

就可以自己做顯示畫面了

甚至重跑 都可以有UI

