

Spring batch

Natalia Reyes Altamirano



¿Qué es un batch?

- Un proceso batch o proceso por lote es un proceso pensado para trabajar con volúmenes muy grandes de datos y generalmente de una forma programada. Es decir, sin intervención humana.
- Algunos ejemplos de batch son la carga de un fichero enorme con millones de registros; o bien un proceso nocturno que, a partir de una serie de consultas, envía una gran cantidad de e-mails, sms, etc.

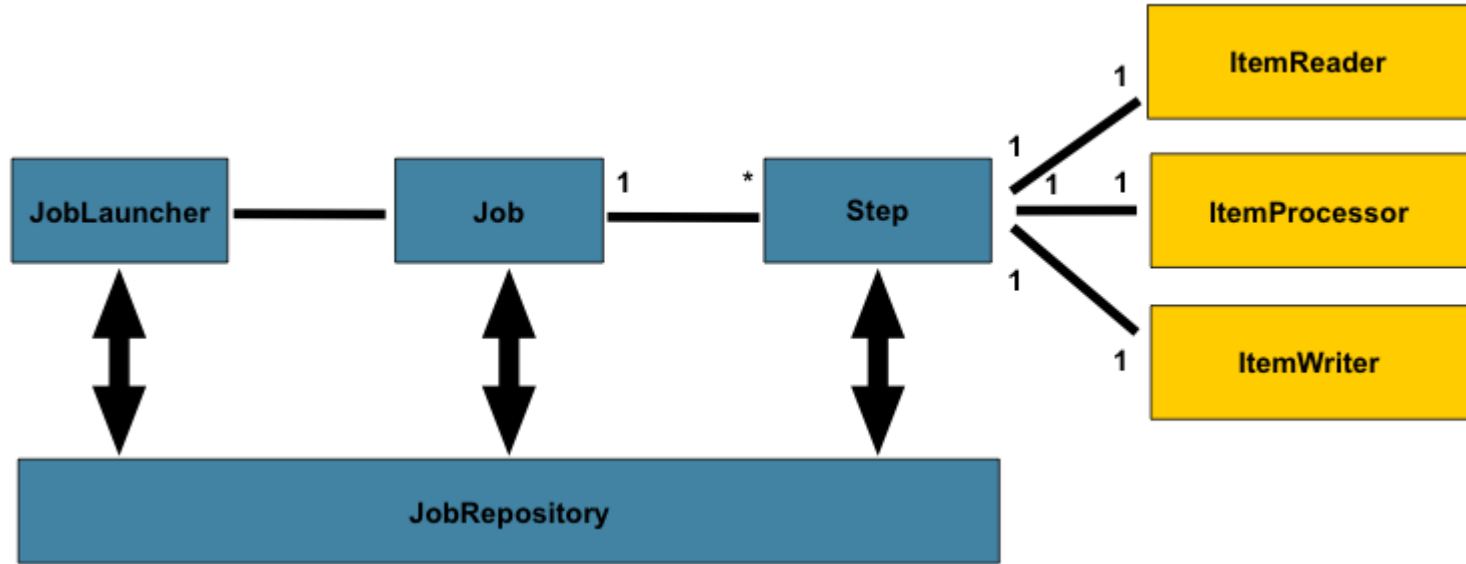


¿Qué es Spring batch?

- Es un framework ligero open source para procesamientos batch o procesamientos por lotes. Este framework es un módulo de Spring y fue desarrollado como fruto de una colaboración entre SpringSource (ahora Pivotal) y Accenture. Spring Batch es un framework ligero enfocado específicamente en la creación de procesos batch. Spring Batch también proporciona con una gran cantidad de componentes que intentan dar soporte a las diferentes necesidades que suelen surgir a la hora de crear estos programas: trazas, transaccionalidad, contingencia, estadísticas, paralelismo, particionamiento, lectura y escritura de datos, etc...



Componentes de Spring Batch

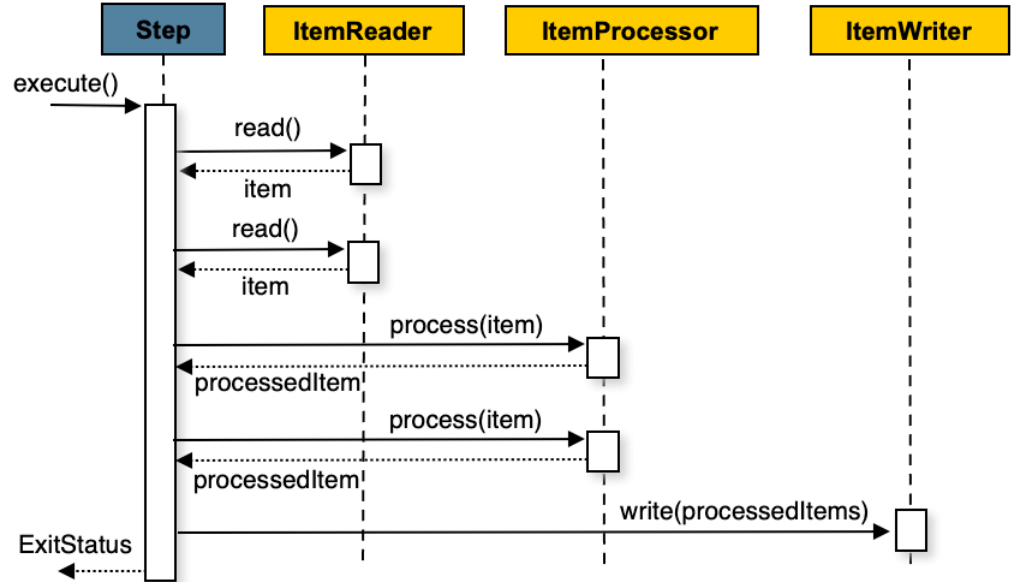


Batch Processing

Los parámetros esenciales del batch processing incluyen:

- Quien está enviando el trabajo
- Qué programa se ejecutará
- La ubicación de las entradas y salidas
- Cuándo se debe ejecutar el trabajo

Se responde a las preguntas quién, qué, donde y por qué



Job Repository

- Spring Batch está pensando para que la información de los procesamientos quede almacenada en un repositorio persistente (Job Repository) o bien en memoria. Este repositorio se utiliza:
 - Principalmente para escritura
 - Es consultado para comprobar si ya se ha procesado un fichero previamente.
 - Si se produce un job fallido, en lugar de re-procesar todo el fichero de nuevo, únicamente se re-procesa el trozo que ha fallado.
- El JobRepository escribe y consulta una serie de tablas existentes en base de datos.
- Es responsabilidad de este repositorio almacenar información sobre cada job, step que se produzca, los parámetros del Job, los errores que tengan lugar, etc.



- **JobLauncher:** es el componente encargado de lanzar los procesos suministrando los parámetros de entrada deseados.
- **Job:** El Job es la representación del proceso. Un proceso, a su vez, es un contenedor de pasos (steps).
- **Step:** Un step (paso) es un elemento independiente dentro de un Job (un proceso) que representa una de las fases de las que está compuesto dicho proceso. Un proceso (Job) debe tener, al menos, un step.
- Aunque no es obligatorio, un step puede estar compuesto de tres elementos: reader, writer y processor.

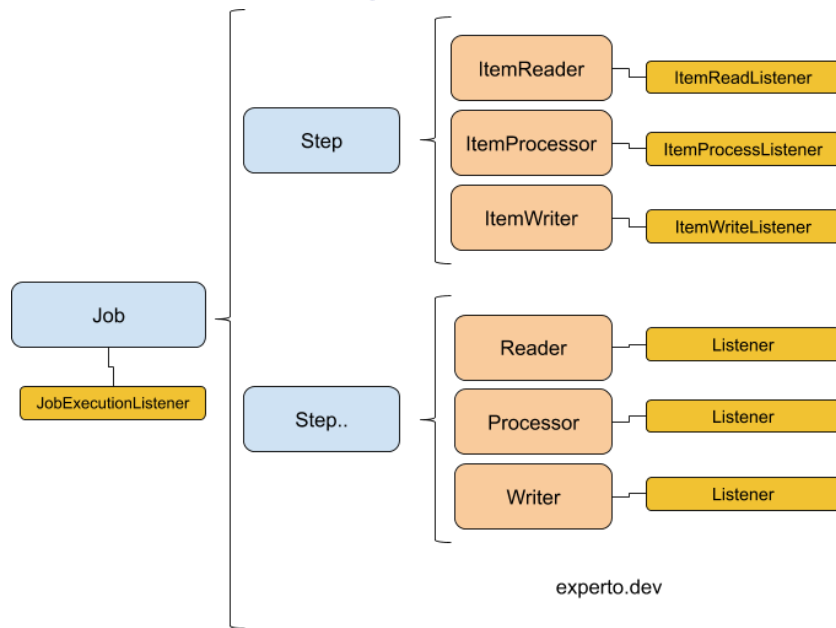


- **ItemReader:** Elemento responsable de leer datos de una fuente de datos (BBDD, fichero, cola de mensajes, etc...)
- **ItemProcessor:** Elemento responsable tratar la información obtenida por el reader. No es obligatorio su uso.
- **ItemWriter:** Elemento responsable guardar la información leída por el reader o tratada por el processor. Si hay un reader debe haber un writer.
- **Tasklet:** Un step no tiene que estar compuesto por un reader, processor y writer. También puede tener únicamente una lógica de negocio. Es el caso del tasklet con el código que se desea ejecutar en el step.



Cómo configurar el Job en Spring Batch

- Un Job contiene Steps y cada Step requiere generalmente (no siempre) un Reader, un Processor, y un Writer. Opcionalmente se pueden agregar Listener para escuchar y saber que está pasando en cada parte del proceso del batch.
- **El Reader** lee datos.
- El Procesor recibe los datos del Reader y los procesa para luego entregarlos al Writer.
- **El Writer** recibe los datos que fueron procesados y se encarga de guardarlos.
- Los **Listener** “escuchan” lo que sucede durante el proceso.



Chunk

- Se usa al configurar los Job. Un chunk es una unidad de procesamiento. Con este valor se le dice a Spring Batch que procese una determinada cantidad de registros y al completar la cantidad envíe todo al reader para hacer el commit.

