

## Spring Batch

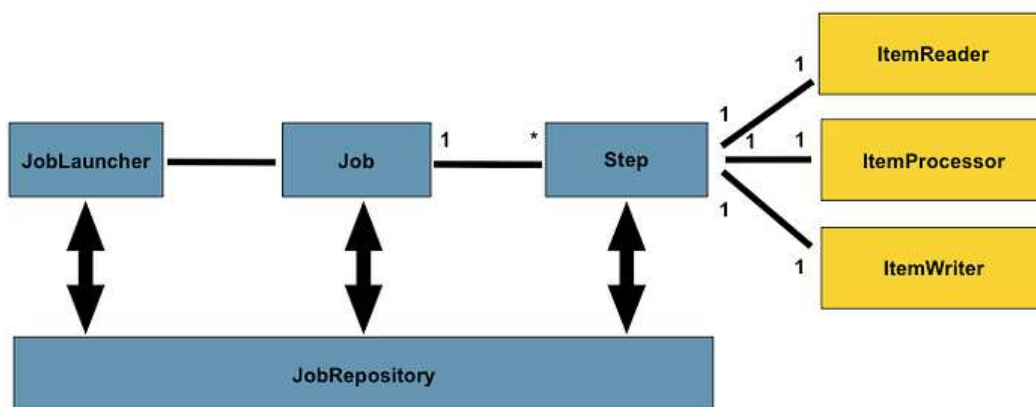
Spring Batch es un framework de aplicaciones batch que facilita el desarrollo de una aplicación usando Java. Spring Batch proporciona funciones reutilizables para procesar grandes registros, incluyendo logging/tracing, gestión de transacciones, estadísticas de procesamiento de trabajos, reinicio de trabajos, omisión y gestión de recursos. Es una propiedad esencial del procesamiento batch que puede ser reutilizada. El punto importante aquí es que es reutilizable, porque si un programador genera periódicamente el proceso batch necesario, sería un trabajo ineficiente.

### Quartz vs Spring Batch

A menudo, algunas personas confunden los planificadores, en particular Quartz, con Spring Batch. Spring Batch no es una alternativa a un planificador, pero puede ser el objetivo del planificador.

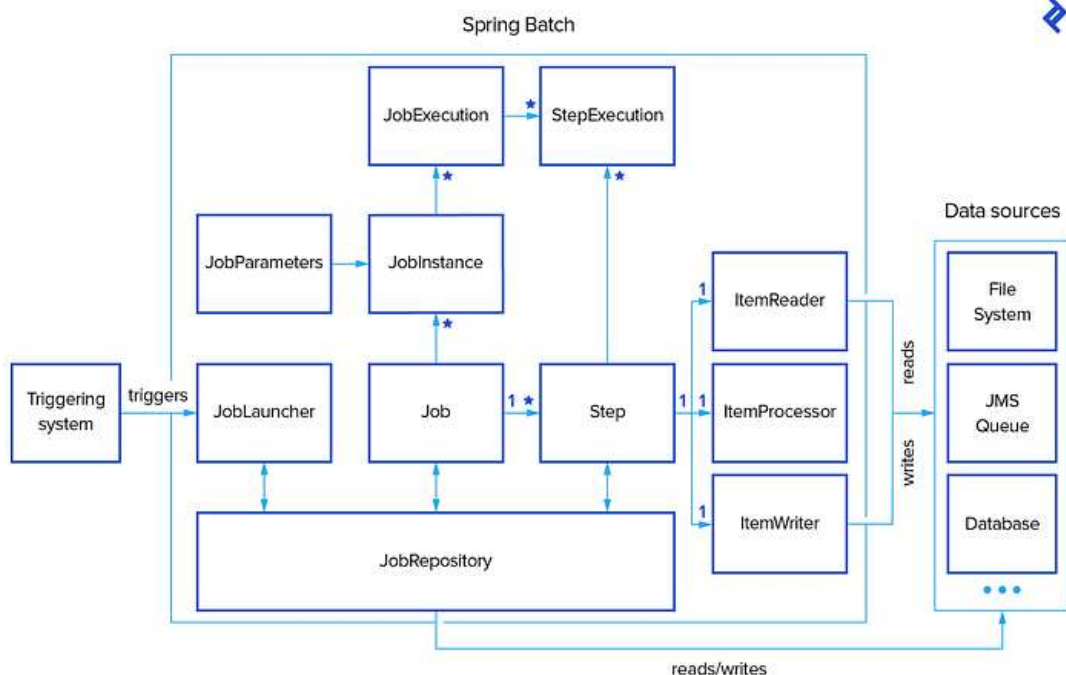
Quartz es un framework de planificación, como “ejecutar algo cada hora o cada último viernes del mes”, similar a otros planificadores. Spring Batch es un framework que define “algo” que es ejecutable. Puedes implementar ese algo definiendo un Job que consiste en Steps. De hecho, Spring Batch puede ser alternado con otros como Hadoop, Talend, Apache Spark y Kafka.

### Key Concepts



Como puedes ver en la imagen de arriba, hay palabras clave: Job, Step e Item. El Job puede tener uno o más Steps y un Step puede tener ItemReader, ItemProcessor e ItemWriter, uno para cada uno. ItemReader e ItemWriter son obligatorios, pero ItemProcessor es opcional.

Job y Step pueden acceder a un repositorio persistente llamado JobRepository que puede almacenar metadatos usados para la ejecución del Job.



## Job

Un Job es una unidad de tarea batch que consta de uno o más Steps. Llamar a los Steps registrados para cada uno es el rol del Job.

## JobExecution

JobExecution es un intento único de ejecutar el Job. La ejecución puede finalizar con un estado de éxito o fracaso. Hay dos tipos que representan el estado en Spring Batch: BatchStatus y ExitStatus. BatchStatus

## BatchStatus

BatchStatus es un tipo de enumeración que representa el estado de la ejecución.

```
public enum BatchStatus {  
    STARTING, STARTED, STOPPING, STOPPED,  
    FAILED, COMPLETED, ABANDONED  
    ...  
}
```

## ExitStatus

ExitStatus es la clase que tiene campos que representan el estado del resultado de la ejecución. Tiene los siguientes estados listados:

- UNKNOWN: el trabajo termina en estado desconocido. Significa un estado no continuable.
- EXECUTING: estado continuable sin acción adicional. Se utiliza para ejecuciones asíncronas que no requieren esperar resultados.
- COMPLETED: estado de procesamiento finalizado.
- NOOP: representa que no ocurrió ningún procesamiento, como cuando las operaciones ya están terminadas.
- FAILED: procesamiento finalizado con error.
- STOPPED: procesamiento finalizado con un estado de interrupción.

## JobInstance

JobInstance se refiere a un proceso lógico de un Job. Un Job debe ejecutarse una vez. Por ejemplo, el Job 'EndOfDay' se ejecuta cada medianoche y las ejecuciones del Job se acumulan cada día. En el caso de este Job, hay un JobInstance lógico por día. Es decir, es una colección de JobExecutions y se puede identificar mediante JobParameters.

## Step

Un Step es una unidad independiente y secuencial del Job. Cada Step debe tener solo un ItemReader y un ItemWriter. Si es necesario, puede tener un ItemProcessor entre ItemReader y ItemWriter para funciones como convertir tipos de datos o calcular estadísticas.

## StepExecution

StepExecution es un intento único de ejecutar el Step derivado de un JobExecution. StepExecution puede finalizar con un estado de éxito o fracaso, por lo que tiene un BatchStatus y un ExitStatus como un JobExecution. Por ejemplo, StepExecution puede tener el BatchStatus: STARTED, FAILED, COMPLETED, etc.

## ExecutionContext

ExecutionContext representa una colección de pares clave/valor persistidos y controlados por el framework para almacenar el estado persistente que está delimitado a un objeto JobExecution o StepExecution.

### **JobRepository**

JobRepository representa un repositorio persistente utilizado para almacenar los modelos de dominio batch y su estado accediendo a la base de datos.

JobRepository puede usar el procesamiento general desde el iniciador de trabajos para finalizar los últimos pasos.