



Spring Batch

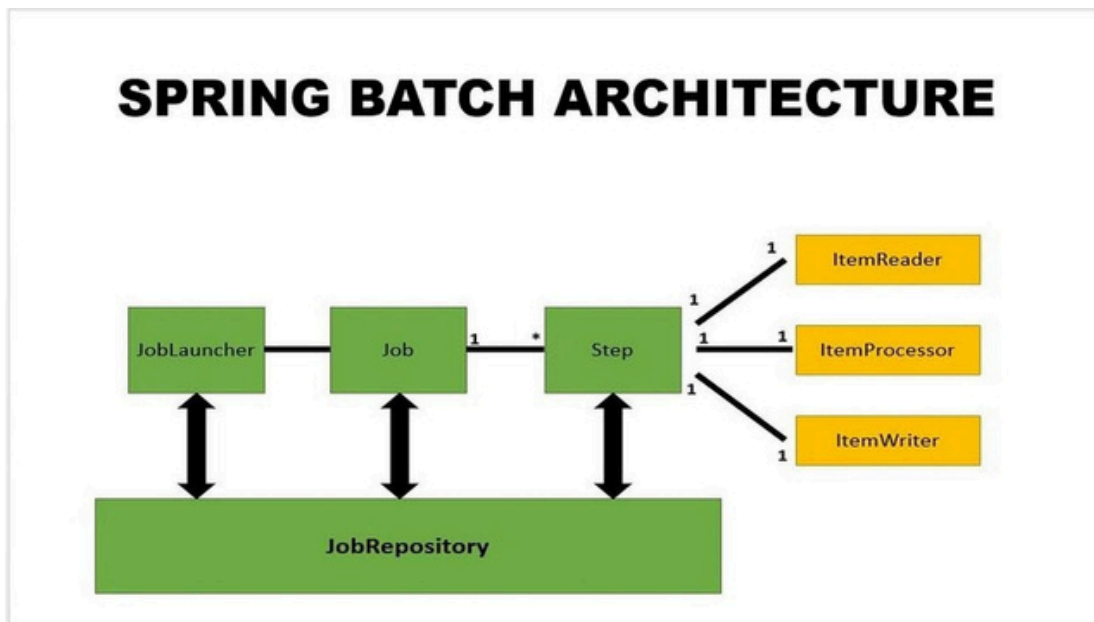
Es un framework ligero enfocado específicamente en la creación de procesos batch.

Además de marcar unas directrices para el diseño de procesos, Spring Batch 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...



operaciones comerciales incluyen:

- Procesamiento automatizado y complejo de grandes volúmenes
- Aplicación periódica de reglas complejas de negocio procesadas repetitivamente en conjuntos
- Integración de la información recibida de sistemas internos y externos que normalmente requiere formateo, validación y procesamiento en una forma transaccional en el sistema de registro.
- El procesamiento por lotes se utiliza para procesar miles de millones de transacciones todos los días para las empresas.



- **JobRepository:** componente encargado de la persistencia de metadatos relativos a los procesos tales como procesos en curso o estados de las ejecuciones.
- **JobLauncher:** 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.
- **ItemReader, ItemWriter, ItemProcessor:** componentes opcionales para el tratamiento de datos (lectura, escritura y procesado).





JobInstance: Es una representación lógica de un determinado job con ciertos parámetros de ejecución.

JobParameters: Es un conjunto de parámetros utilizado para comenzar la ejecución de un Job. Puede usarse para identificar una ejecución o para proporcionar datos a la propia ejecución.

JobExecution: Representa la ejecución de una determinada instancia de un job en un determinado instante de tiempo. Identifica una ejecución del job.

Un **StepExecution** representa cada intento de ejecución de un determinado Step. Cada vez que se ejecuta un Step se creará un nuevo StepExecution.

Item Readers, Item Writers, Item Processors

- **Item Readers:** Representa la fase de lectura de información para un Step. El ItemReader realizará la lectura elemento a elemento. Una vez concluya la lectura de todos los elementos de la fuente de información configurada retornará null.
- **Item Writer:** Representa la fase de salida o escritura de información de un Step, batch o chunk. Generalmente un ItemWriter no tiene conocimiento de la información que recibirá a continuación.
- **Item Processor:** Representa la lógica de negocio implementada para realizar el procesamiento de la información. Mientras que un ItemReader realiza la lectura de elementos y el ItemWriter se encarga de la persistencia de la información.

FlatFileItemReaders

Componente genérico de Spring Batch que permite realizar la obtención de información en un fichero o stream. Este componente genérico permite configurar los siguientes aspectos de su implementación:

```
<bean id="csvFileItemReader" class="org.springframework....file.FlatFileItemReader">
  <property name="resource" value="file:csv/inputs/report.csv" />
  <property name="lineMapper">
    <bean class="org.springframework.batch.item.file.mapping.DefaultLineMapper">
      <property name="lineTokenizer">
        <bean class="org.springframework.batch....transform.DelimitedLineTokenizer">
          <property name="names" value="id,name" />
        </bean>
      </property>
      <property name="fieldSetMapper">
        <bean class="com.maldiny.spring.batch....MyObjectFieldSetMapper" />
      </property>
    </bean>
  </property>
</bean>
```

Resource: Recurso de entrada (fichero).

LineMapper: Permite realizar la lectura de información. Se podrán realizar numerosas configuraciones sobre este elemento para determinar el número de campos a obtener, líneas que ignorar.

FieldSetMapper: Permite realizar el mapeo de la información obtenida a objetos generados con una determinada clase.

