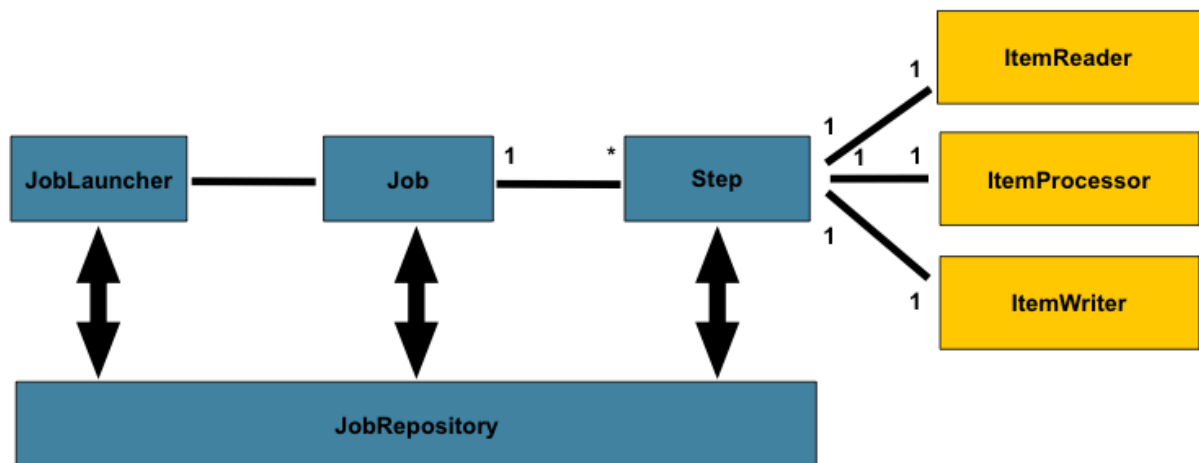


Spring Batch es una librería o framework que nos sirve para manejar lotes de información en grandes volúmenes, esto lo vamos a lograr agregando la dependencia (Spring Batch) una vez iniciemos el proyecto, junto con otras dependencias como REST JPA dependiendo de lo que hagamos en el proyecto.

Este tratamiento por lotes lo haremos mediante algunos procesos repetitivos, los cuales los va a estar ejecutando durante un tiempo largo, estos procesos son escalables así que podemos ocuparlos para distintos volúmenes de datos.



Este es un diagrama que nos esquematiza cómo funciona este framework, vamos a desglosar cada uno de estos.

El JobRepository es el repositorio de los procesos que va ir ejecutando el trabajo o job, es una tabla o conjunto de tablas donde vamos a poder checar la información sobre los procesos que se van haciendo. Por ejemplo en el ejercicio de la clase importamos unos datos que deben cumplir con 2 parametros ser gender "Female" y vivir en "China" esta importación se guardó en Batch\_Job\_Inatance la cual podemos consultar en la base de datos en el SQLWorkbrench donde me dará el tipo de trabajo y estos tienen un id único.

JobLauncher en el ejercicio de hoy tenemos el job controller que es donde vemos el método para importar los datos si lo vemos bien este tiene una interface llamada JobLauncher por lo cual, el JobLauncher es una interfaz que vamos a poder implementar para poder controlar nuestros procesos o trabajos dependiendo de los requerimientos que necesitemos y de nuestras tablas.

Parecido a los otros ejercicios de CRUD que hemos hecho en este controller tenemos un método post que es el que nos ayudará a importar estos datos la manera de ejecutarlo es mediante apis que tendrán por ejemplo terminación /jobs/importCustomers esto lo tendremos que poner en el postman con un post para que nos importe los Customers con los parámetros que quisiéramos

Job es un bloque de trabajo y está compuesto por uno o varios pasos o steps. Una vez se han llevado a cabo todos estos pasos, se considera el job como completado. En el ejemplo que vimos como ya lo mencionamos en JobLauncher es donde vamos a poner las condiciones que queremos que nuestros datos cumplan para ser importados en nuestro ejemplo estaría localizado en CustomersProcessor

Steps es algo que la vamos a indicar al job cuántos “pasos” tiene que hacer para llevar a cabo el trabajo esté como mínimo debe ser un paso cada paso debe contar con ItemReader, ItemProcessor e ItemWriter estos pasos en el ejemplo que vimos lo tenemos en SpringBatchConfig donde tenemos Step1 y también definimos esos 3 elementos que requerimos para nuestro Step

ItemReader: se encarga de la lectura del procesamiento por lotes. Esta lectura puede ser, por ejemplo, de una base de datos; o también podría ser de un broker de mensajes o bien un fichero csv, xml, json, etc.

ItemProcessor: se encarga de transformar items previamente leídos. Esta transformación además de incluir cambios en el formato puede incluir filtrado de datos o lógica de negocio.

ItemWriter: este elemento es lo opuesto al itemReader. Se encarga de la escritura de los ítems. Esta puede ser inserciones en una base de datos, en un fichero csv, en un broker de mensajes, etc.

Siento que estas fases son las más complicadas de implementar ya que tienes que basarte en métodos, interfaces para poder realizar estas tareas que integran el Step por ejemplo en el ejercicio que vimos tenemos un método reader para leer los datos que queremos importar, también tenemos linemapper que es una interfaz para mapear cadenas de strings leídas, además tenemos el processor que lo da el spring batch y el RepositoryItemWriter estos métodos son utilizados en el Step para realizar este trabajo