Spring batch

Batch processing is to process some logics or methods to modify or built data without any interaction or user or GUI or external, in short we didn’t have to trigger anything manually.

Its different from scheduling like in QUARTZ . it like a job , that reads data from data src , process some data in some way , write data in someway. Spring batch automates batch iteration of these three things.

Spring batch allow us to write batch processing logic or creating batch apps based on java easily.

Spring batch stores state of our batch jobs. It stores metadata about our job in repository at various state of executions of jobs.

Error handling . it provides robust error handling like , restart , retry, skip.

Built in readers and writers, for interacting with data stores , like

Flatfileitemreader , flatfileitemwriter for csv file

jdbcCursoritemreader , jdbcbatchitemwriter for relational databses

jsonitemreader , jsonfileitemwriter for json files

kafkaitemreader , kafkaitemwriter for kafka

there are 3 interfaces itemwriter, itemreader, itemprocessor

spring batch architecture-

now the thing is something is gonna trigger a joblauncher for jobs, like assume we have schedular , jobluancher gonna trigger job and job is going to trigger step and every step have 3 things itemreader, itemwriter, itemprocessor.

A job instance is gonna created means joblauncher have to trigger job via some jobparameters in short job plus jobparameters result into jobinstance that is a logical run.

Now a jobexecutation is gonna created that is physical run means for every job occurrence there going to be a jobexecutaion. Then there is stepexecutaion for step .

And for everything a meta data is going to be stored in jobrepository.

A job can have multiple job instances, a jobinstance can have multiple jobexecution , a job can have multiple step and a step can have multiple stepexecution.

Means these are in one to many relations.