**Spring Batch**

Spring Batch is an open-source framework for batch processing. It is a lightweight, comprehensive solution designed to enable the development of robust batch applications, which are often found in modern enterprise systems. Spring Batch builds upon the POJO-based development approach of the Spring Framework.

**How to add Spring batch in spring boot project?**

Add spring batch dependency in pom.xml.

**How to enable spring batch in spring boot application?**

Add @EnableBatchProcessing annotation in main class, this annotation register many beans for us like JobBuilderFactory, StepBuilderRepository, JobRepository and so on.

**What is a Job and Step in spring batch?**

A job represents the entire batch process we wanna execute. It defines one or more steps that execute in an order we commenly called as flow. A step is a phase in a batch job that defines how the actual processing will occur for that portion of a job. The framework retains information about JobExecution within the JobRepository, allowing a job to be restarted at the point of failure, preventing reprocessing of data.

Graphical user interface, diagram, application

Description automatically generated

JobInstances refer to the concept of a logical job run, which is defined by the job and its parameters. Once a job instance has been successfully completed it cannot be restarted. The job can only be re-run with different parameters. Only one JobInstance corresponds to a particular job and its parameters.

Graphical user interface, diagram, application

Description automatically generated

What are different types of Steps in batch?

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

on, from, to are called as transition. The on transition accepts a pattern argument that is matched against the ExitStatus. Transitions are used to define which step in a job executes next.

To understand conditional logic properly we can think “from” as ***else*** if, “on” as ***equals to***, “to” as ***then***

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**JobExecutionDecider** : Job execution decider is used to implement custom exit status codes. The JobExecution decider is registered within the Job configuration and is used to provide a custom ExitStatus following the execution of a step.

**Listeners** : At some point in a batch job you may need to take control of batch processing and introduce your own logic, that’s where listeners come into play. Listeners provide hooks into the flow of a job at various points that allow you additional batch processing logic for additional use cases.

Graphical user interface, application

Description automatically generated

Graphical user interface, diagram

Description automatically generated

External flows and job steps allow sequences of steps to be reused within jobs, preventing duplication of job configuration.

Graphical user interface, PowerPoint

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

**Nested Job** : In order to nest a job within another job we need to create a job step. Look below picture to unserstand.

Graphical user interface

Description automatically generated

We are now creating a job step (nestedBillingJobStep()) .

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

**Parallel Flow** : We can execute a parallel flow between the job. We can use split transition and we need to provide a task executor then we need to add flows to it.

Graphical user interface, text, application

Description automatically generated

**Chunk Oriented** :

Graphical user interface, website

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated