**Spring Batch**

Many applications within the enterprise domain require bulk processing to perform business operations in mission critical environments. These business operations include automated, complex processing of large volumes of information that is most efficiently processed without user interaction. These operations typically include time based events (e.g. month-end calculations, notices or correspondence), periodic application of complex business rules processed repetitively across very large data sets (e.g. Insurance benefit determination or rate adjustments), or the integration of information that is received from internal and external systems that typically requires formatting, validation and processing in a transactional manner into the system of record. Batch processing is used to process billions of transactions every day for enterprises.

**Introduction**

* The core concept of spring batch as the name suggests is processing of data in batches.
* Spring Batch is a lightweight, comprehensive batch framework designed to enable the development of robust batch applications vital for the daily operations of enterprise systems.
* Spring Batch builds upon the productivity, POJO-based development approach, and general ease of use capabilities people have come to know from the Spring Framework, while making it easy for developers to access.
* Spring Batch provides reusable functions that are essential in processing large volumes of records, including :
  + - Logging/tracing
    - Transaction management
    - Job processing statistics
    - Job restart, skip, and resource management.

**Architecture**

DB

DB

Files

Q

Job Repository

Item Writer

Item Reader

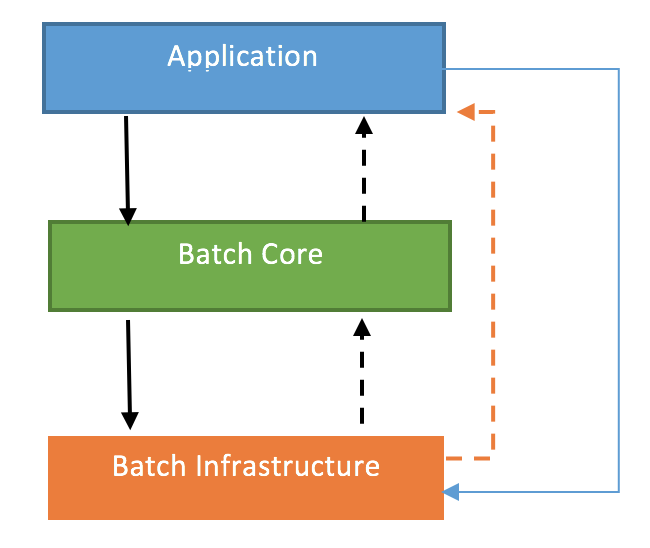
ItemProcessor

Step

Job

Job Launcher

Scheduler



* Application: This contains all the batch jobs and code written by the developer based on business needs.
* Batch Core: It contains the runtime classes necessary to run a batch job. The classes such are JobLauncher, Job and Step implementation are part of the Batch Core.
* Batch Infrastructure: This contains the reader and writer services which are used by developer and the framework itself. The classes are ItemReader and ItemWriter. It also contains services to retry read and write.

#### Spring Batch Business Use Case

1. At the end of a month when a company has to send salary to its employee’s respective accounts.
2. Processing of salary slips at month end is when spring batch can be used.
3. Sending out mass communication emails.
4. For generating automated reports on daily, weekly or monthly basis.
5. Executing business workflow automatically without human intervention.

#### Spring Batch Technical Use Cases

1. For automatic test execution at a defined frequency.
2. This can be used for doing automatic database updates on a predefined frequency.
3. It can be used with a queue system to handle huge amount of transactions without any failure.
4. Spring batch can be used along with APIs to perform tasks such as health check for server or application, dummy data generation for load test, etc.

## Batch Processing Strategies

To help design and implement batch systems, basic batch application building blocks and patterns should be provided to the designers and programmers in form of sample structure charts and code shells. When starting to design a batch job, the business logic should be decomposed into a series of steps which can be implemented using the following standard building blocks:

* Conversion Applications: For each type of file supplied by or generated to an external system, a conversion application will need to be created to convert the transaction records supplied into a standard format required for processing. This type of batch application can partly or entirely consist of translation utility modules (see Basic Batch Services).
* Validation Applications: Validation applications ensure that all input/output records are correct and consistent. Validation is typically based on file headers and trailers, checksums and validation algorithms as well as record level cross-checks.
* Extract Applications: An application that reads a set of records from a database or input file, selects records based on predefined rules, and writes the records to an output file.
* Extract/Update Applications: An application that reads records from a database or an input file, and makes changes to a database or an output file driven by the data found in each input record.
* Processing and Updating Applications: An application that performs processing on input transactions from an extract or a validation application. The processing will usually involve reading a database to obtain data required for processing, potentially updating the database and creating records for output processing.
* Output/Format Applications: Applications reading an input file, restructures data from this record according to a standard format, and produces an output file for printing or transmission to another program or system.

In addition to the main building blocks, each application may use one or more of standard utility steps, such as:

* Sort - A Program that reads an input file and produces an output file where records have been re-sequenced according to a sort key field in the records. Sorts are usually performed by standard system utilities.
* Split - A program that reads a single input file, and writes each record to one of several output files based on a field value. Splits can be tailored or performed by parameter-driven standard system utilities.
* Merge - A program that reads records from multiple input files and produces one output file with combined data from the input files. Merges can be tailored or performed by parameter-driven standard system utilities.

(Updated upto basics. More information visit :

https://docs.spring.io/spring-batch/trunk/reference/html/spring-batch-intro.html)