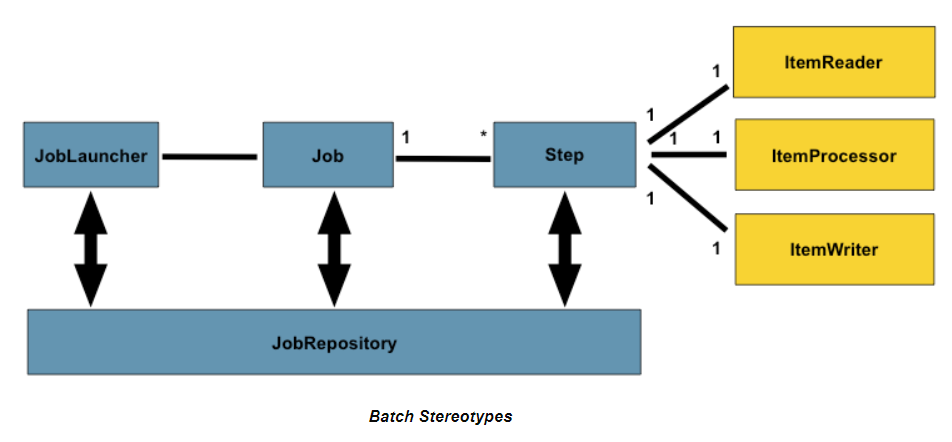
1. Spring Batch

* Batch Processing là chương trình thực hiện một loạt các bước mà không chịu sự can thiệp từ người dùng.
* Spring Batch là một framework sinh ra để xử lý Batch dựa trên các đặc điểm của Spring Framework.
* Dùng trong các trường hợp đọc một bản ghi lớn từ cơ sở dữ liệu, sửa đổi các bản ghi theo yêu cầu,...

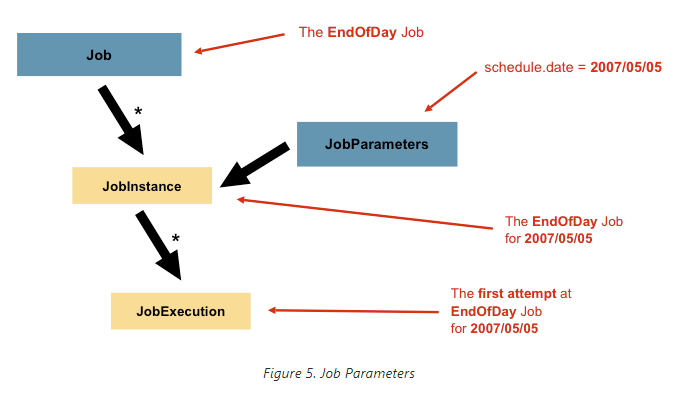
1. Các thành phần của Spring Batch



Spring Batch bao gồm 4 thành phần:

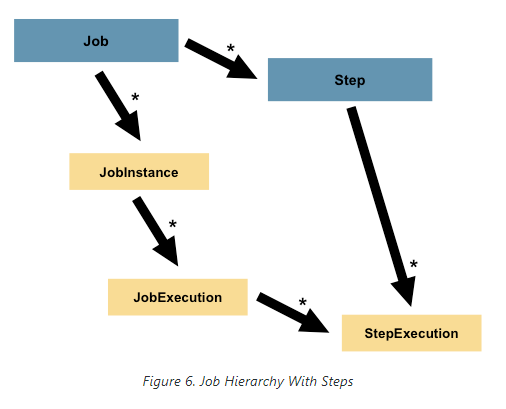
* JobLauncher là 1 interface cho 1 job thực hiện truyền tham số đầu vào, Job muốn launch thì phải thông qua JobLauncher.
* Job sẽ thực hiện nhận các tham số đầu vào bởi JobLauncher và thực hiện các bước được xây dựng trong Job.
* Step được xây dựng để điều khiển các hoạt động của Batch Processing. 1 step bao gồm các thành phần: ItemReader, ItemProcessor, ItemWriter.
* JobRepository cung cấp CRUD cho JobLauncher, Job, Step trong quá trình chạy.

1. Job



* 1 job sẽ có các job instance và các job instance phân biệt với nhau thông qua jobparameter.
* 1 job instance chỉ hoàn thành khi jobexecution được thực hiện thành công, nếu fail thì lần thực hiện thành công lần sau mới được tính là 1 JobInstance.

1. Step



Step là 1 đối tượng thực hiện các hành động độc lập và mỗi Job sẽ bao gồm một hoặc nhiều bước.

StepExecution được sinh ra khi một Step bắt đầu. Nếu 1 step bị lỗi do step trước nó không thực thi được thì cả quá trình sẽ không được thực hiện.

1. ExecutionContext

* Là các cặp key/value được framework ghi lại các trạng thái hệ thống và thực thi được lưu vào trong database.

1. Job Repository

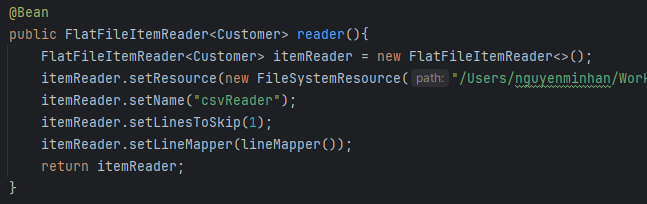
* Cung cấp CRUD cho JobLauncher, Job, Step trong quá trình thực thi.

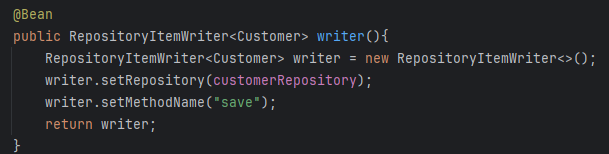
1. JobLauncher
   * JobLauncher là 1 interface cho 1 job thực hiện truyền tham số đầu vào, Job muốn launch thì phải thông qua JobLauncher. Khi chạy cần truyền tham số là Job và JobParameter vào JobLauncher.
2. ItemReader, ItemWriter, ItemProcessor

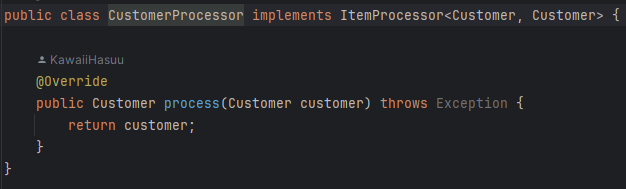
* ItemReader sẽ đóng vai trò là đầu vào, đọc các dữ liệu cho 1 Step. Đầu vào có thể là FlatFile, XML hoặc dữ liệu từ Database.
* ItemWriter sẽ đóng vai trò là đầu ra và lưu dữ liệu cho 1 Step. Các tài nguyên vẫn cần giống như ItemReader nhưng ở đây sẽ đóng vai trò là đầu ra, các dữ liệu được đọc sẽ được lưu vào các tài nguyên đầu ra như đã cấu hình.
* ItemProcessor sẽ giúp ta tùy chỉnh đầu ra theo mong muốn

1. Config Spring Batch

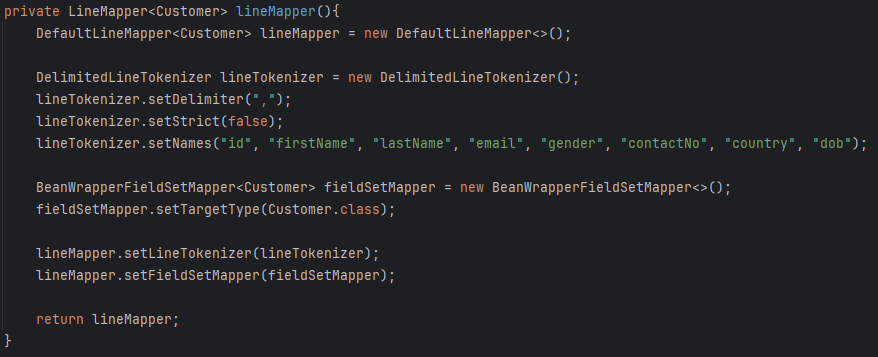
* JobExecution được dùng để xem các thông tin về quá trình thực thi bao gồm thời gian thực thi, thời gian bắt đầu, kết thúc, trạng thái của công việc,... Để dùng thì có thể sử dụng
* Để config ta dùng 2 annotation @Configuration và @EnableBatchProcessing
* @EnableBatchConfiguration sẽ cung cấp cho ta các base configuration, các beans sẽ được tạo ra để ta sử dụng như:
* JobRepository: a bean named jobRepository
* JobLauncher: a bean named jobLauncher
* JobRegistry: a bean named jobRegistry
* JobExplorer: a bean named jobExplorer
* JobOperator: a bean named jobOperator
* Tiếp theo ta sẽ cần tạo Reader, Writer và Processor cho Job



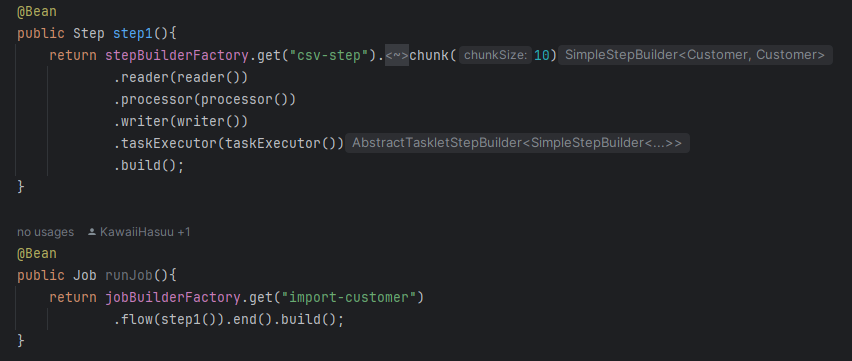




Ngoài ra để xác định cách ánh xạ dữ liệu từ file csv vào các đối tượng cho phân Reader ta sẽ thiết lập phần LineMapper:



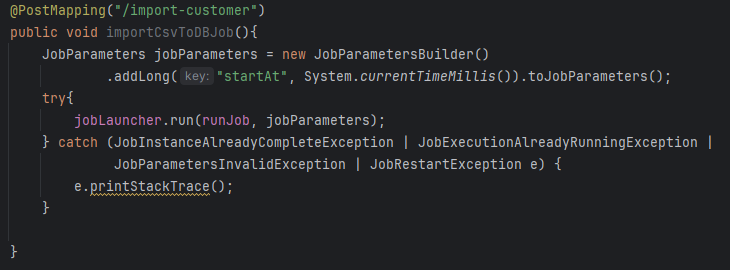
* Khi đã cấu hình xong Reader, Writer và Processor ta sẽ tiến hành Config Step và Job:



* Step ở đây có tên là step1, chunk(10) ở đây có nghĩa là cứ read + process 10 record thì chương trình sẽ tiến hành write kết quả vào database.
* Job ở đây có tên là runJob và nó sẽ tiến hành thực thi step1 trước khi kết thúc.

1. Run batch:

Khi đã config xong ta sẽ tiến cấu hình Controller:



Ta sẽ tiến hành chạy thử và các kết quả sẽ được lưu từ file Csv vào Database.