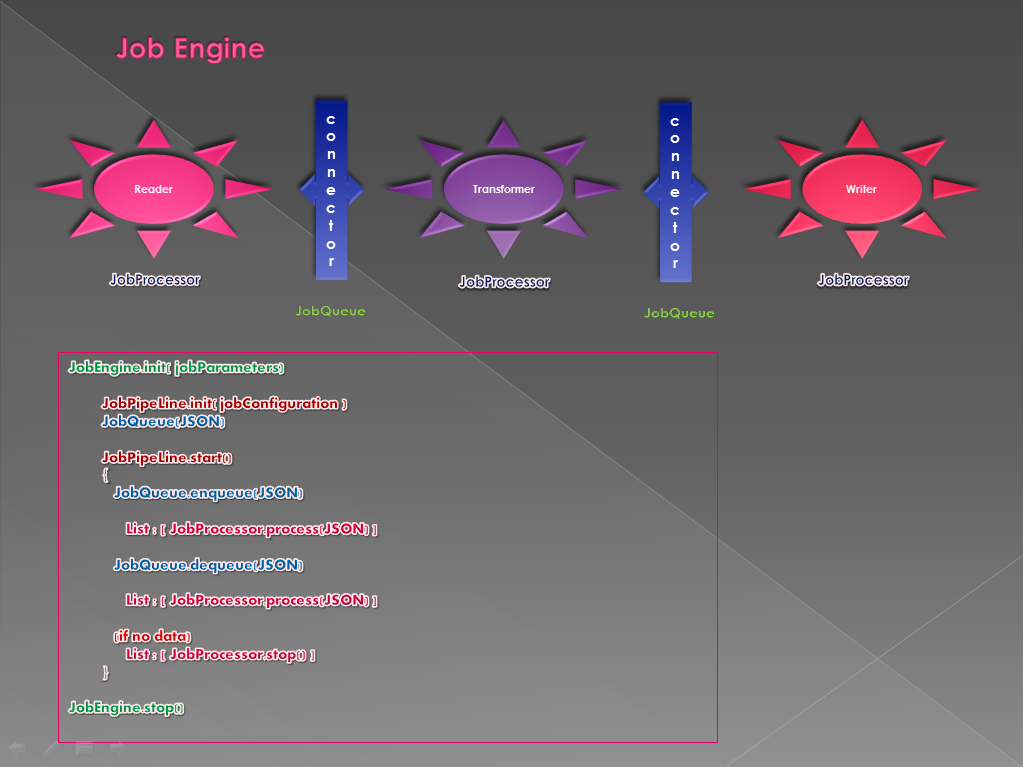
***Job Engine***

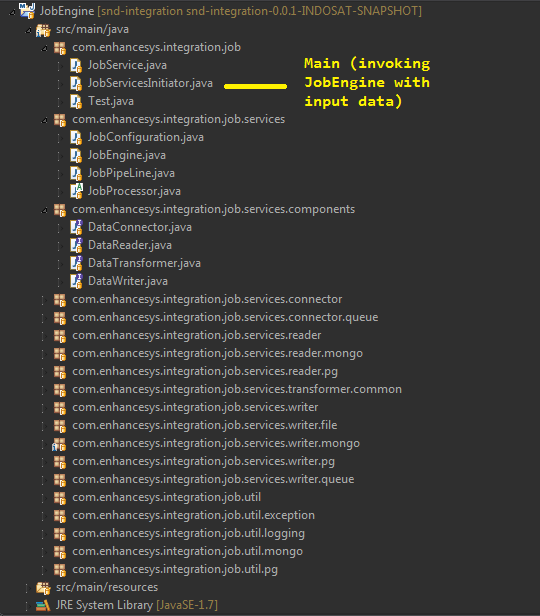
JobEngine is a project used for read the data from FILE/DB/JMS and process/write into FILE/DB/JMS with generic and efficient way. It has four major components.

* DataReader – read data and send to next level
* DataTransformer – transform data with lookup or any other operation and send to next level
* DataWriter – write a data
* DataConnecter – Connecting the any other different two components.

Below picture is overview of JobEngine implementation.



**JobEngine Project Explorer**

****

**JobEngine’s configuration data handled by JobConfiguration.java class**

**JobServiceInitiator.java accepts file path from the args of main class and the file data converted as JSON (JobParameter) and then invoking the JobEngine.java.**

**JSON (JobParameter) is an input for JobEngine**

*Example for JobParameter*

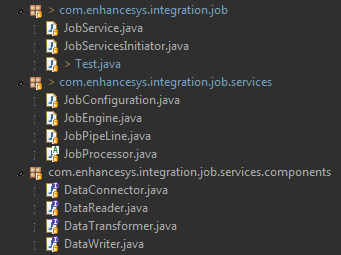
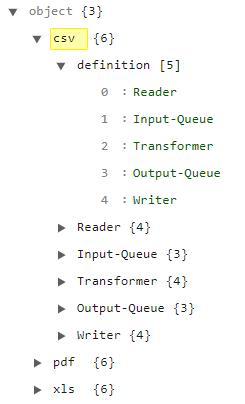


* Here **template-name, template-id** and **job-id** are preconfigured, so these fields are **mandatory** for JobEngine JobParameter’s feeds.
* If any other custom configuration maintained in **job-data‘s** feed.
* Here **request-id** and **RequestUpdateConf** are used to track the status of data processing.
* **StoreFileConf** is used for store data into **MongoDB**.
* The **job-id** is the name of **the file from /jobengineconf/ dump-templates/**
* The **template-name** is the name of **the file from /jobengineconf/ job-templates/**
* The **template-id** is the **JSON object key name** from **the template-name of the file**.

**Job-Templates (/jobengineconf/job-templates/\*)**

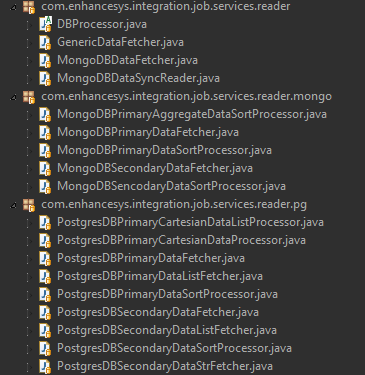
**Example: (JOB\_TEMPLATE\_1st\_level\_DATA\_FETCH.json)**

1. **definition** is an array for declaring Data flow in Execution level.



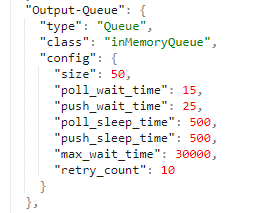
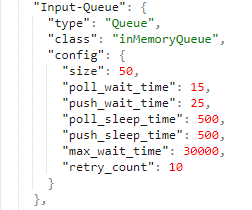
* Here **csv, pdf and xls** are template ids. Based on requirement we can use template id as input.
* **JobPipeLine** is responsible for initiating JobEngine’s components.

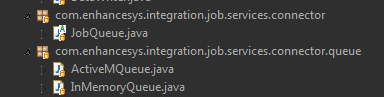
**2. Reader Configuration**

* **GenericDataFetcher** can have multiple data fetcher classes.
* This Reader’s configuration always present in **JOB\_ID. Primary (JSON).**

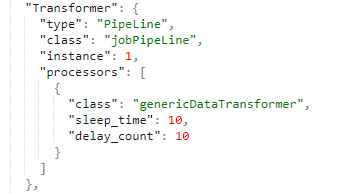
1. **Input-Queue / Output-Queue**





* **Config.size** is JobQueue can have data while processing time.
* If queue state crossed max\_wait\_time or retry\_count, data processing will stop.
* If data processing with large dataset means, then push\_\* and poll\_\* timing should be greater than 1 sec.
* All DataConnector should be implemented with JobQueue
* InMemoryQueue works with ArrayBlockingQueue<JSONObject>((int) queueSize

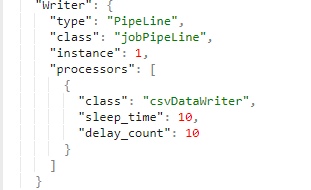
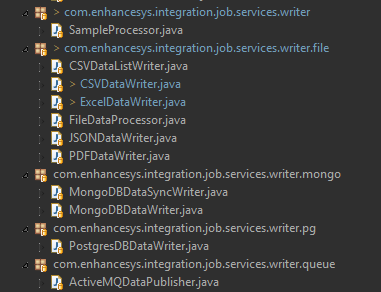
1. **Transformer**





* This class is responsible for lookups, field mapping or any other data transformation.
* This Transformer’s configuration always present in **JOB\_ID. Configuration (JSON) and JOB\_ID. Lookup (JSON).**

1. **Writer**

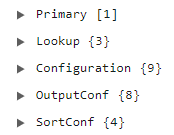
 

* This class is responsible for writing data into file and updates the status of the data processing.
* This Writer’s configuration always present in **JOB\_ID. OutputConf (JSON).**

**If JOB\_ID.StoreFileConf** and **JOB\_ID.RequestUpdateConf** are available then we can track the status of the data processing.

**Dump-Templates (/jobengineconf/dump-templates/\*)**

**Basic structure for job-id**



1. **Primary**



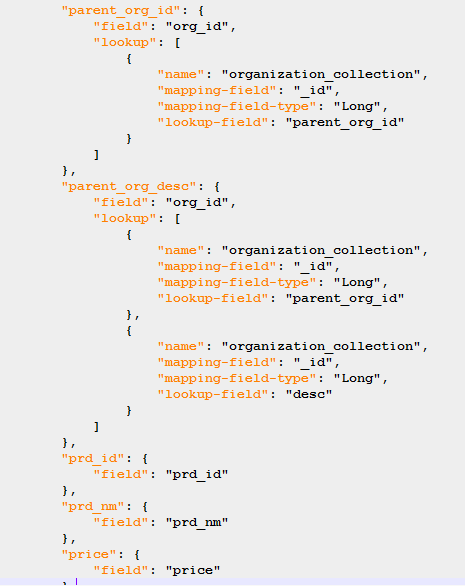
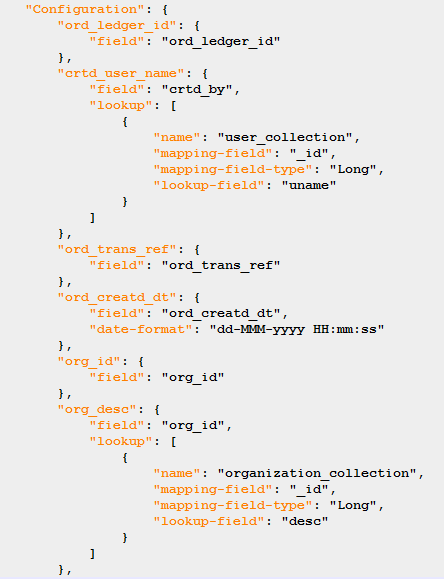
* Primary is an array to initiate and start place for the DataReader.
* DB Data will write into file > if sort needed (sorting will perform) > every time from file only DataReader will read the data.

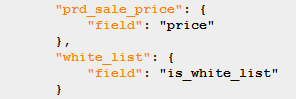
1. **LookUp**



* **LookUp** Configuration is handled by DataTransformer

1. **Configuration**





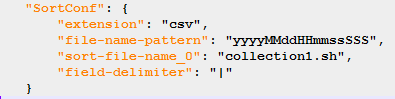
* Configuration is handled by DataTransformer.
* Here from primary object updated with Lookup and any other Logical or arithmetic logic
* The updated Data will send to DataWriter

1. **OutPutConf**



* String, Long, Double, Mathematical, Conditional and Boolean types are available for CSV, Excel, Pdf data writers.
* Here **$** means input data from **job-data** fields. If $ is configured, the particular data should be pass with **job-data**.

1. **SortConf**



* This configuration used by DataReader (if you are using sort processor)