FileAndStore jar

This solution is very flexible, non blocking, extensible.

The client can use ReadAndStore class or CSVRead as suits to get the job done. In case user want to store in database the CSV file read then can use ReadAndStore else if need only model objects then can use CSVRead.

The solution uses OpenCSV third party jar which is very popular in reading CSV files and handling csv data.

The solution is very generic , and can be extended for any database layer which can replace MySQLAccess .

The solution is designed to keep in mind that input file if have any quotes , spaces, special characters etc can be handled at first layer i.e application layer .

Cases handled for CSV data

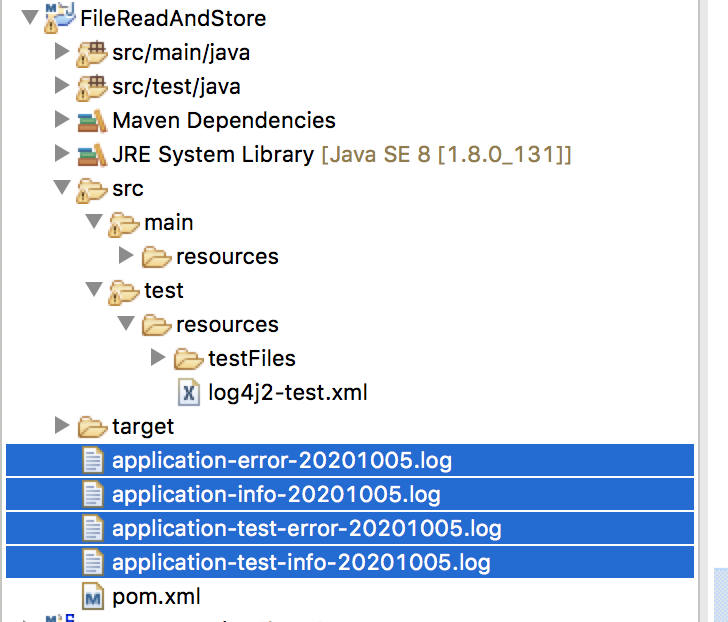
1. Change the position of column i.e segement comes after profit ,
2. Configurable seperator for CSV read hence solution is very flexibile.
3. We have adopted strategy of not Throwing the SQL exception encountered while inserting the data into database as we are logging the SQLException into log file . This is choosen considering this CSVRead is used for a backend process which we dont want to halt even if DB insertion of one of the rows throws error.

For **Logging** Error and Info Log4J2 is used.

Error and Info files are generated to keep user aware which records from CSV are rejected and which ones can be processed successfully , If any invalid line is encountered it is entered in the log file and process moves to the next line.

Logger can be configured in xml so even when the application is deployed you can change the location of logs.

2 Logging files are created which stores the error and info- **application-error-date.log and application-info-date.log.**



Why need 2 files for logging - Error and info file . Info stores all the information you need in case there is error . In production we quickly look at the error file aftet processing each CSV , this is efficent then going throw all the log files. How to make out if there is any error we quickly look at the error file size.

Logging test is also done hence test files are also created.

The solution choose not the handle IOException and FileNotFound exception as these should be handled by the main client.

Some Assumptions made:

Mostly CSV readers are used by Backend process i.e intra day or inter day feeds where we want the kind of strategy to read the whole file and enter cleaned the data into database.

If the file is in GB the solution may not perform very well as we are using ArrayList (and not linkedlist) to store objects.

Buffered reader would be a better choice for big files .

During testing because of some strategy decisions we might not be able to fully unit test the solution for various conditions.

**Some Issues :**

QuoteQuote is inserted as single quote in DB

Because db is only till 2 digits after decimal the incoming number specially profit is not gettting properly stored

CreateTable used to create table in DB:

|  |
| --- |
| CREATE TABLE STORE\_ORDER ( |
| ID INT AUTO\_INCREMENT PRIMARY KEY |
| ORDER\_ID VARCHAR(20) UNIQUE NOT NULL |
| ORDER\_DATE DATE NOT NULL |
| SHIP\_DATE DATE NOT NULL |
| SHIP\_MODE VARCHAR(20) |
| CUSTOMER\_ID VARCHAR(20) UNIQUE NOT NULL |
| CUSTOMER\_NAME VARCHAR(255) NOT NULL |
| SEGMENT VARCHAR(255) |
| COUNTRY VARCHAR(255) NOT NULL |
| CITY VARCHAR(255) |
| STATE VARCHAR(255) |
| POSTAL\_CODE VARCHAR(255) |
| REGION VARCHAR(255) |
| PRODUCT\_ID VARCHAR(20) UNIQUE NOT NULL |
| CATEGORY VARCHAR(255) NOT NULL |
| SUB\_CATEGORY VARCHAR(255) |
| PRODUCT\_NAME VARCHAR(255) |
| SALES DECIMAL(6 |
| QUANTITY INT NOT NULL |
| DISCOUNT DECIMAL(3 |
| PROFIT DECIMAL(6 |
| ); |