Sample Project : Data Processing

The purpose of the sample project is to test the candidate developer and to assess if he/she will be able to handle the current codebase in terms of the libraries and frameworks that our applications are based on. For instance, we deal with a lot of CSV file-based batch processing, in-memory computations, and we make use of Google libraries such as Google Guava (for Collections) and Google Guice (for Dependency Injection). Some of our applications generate the code during run-time using FreeMarker – the sample project will test if the candidate developer will be able to put together an application that uses all these components.

Project Description:

Write a Java application that will achieve the following:

Read a CSV file (input data) and perform the processing logic below:

Firstly, parse the data into a Java object/class named "InputData" which has a List of InputDataRow objects. The InputDataRow represents a row in the CSV file and contains the fields that match the column/headers in the CSV file.

Secondly, calculate the sum of the column called "Amount" based on the combination of "SystemName" and "ProductCode"; this means you need to group the records by SystemName and ProductCode and then calculate the sum of the "Amount" column. Note that some of the Amounts have negative values.

Write the output of each computation to a CSV file; the file should be named using the following convention: {SystemName}_{ProductCode}_{DateTimeStamp}.csv

Finally, and this is crucial, generate an HTLM report file based on the supplied FreeMarker template: **output_data_report.ftl**

Please note the following preferences for your project:

Your source code must be compiled for Java 8 (1.8).

You must create a Maven project and produce an executable Jar which you can run in command line – you should be able to specify JVM options for memory allocation (i.e. Xms and Xmx).

For processing the CSV files as part of your application, please create a "Processor" interface and bind a SysProdProcessor implementation to it using Google Guice dependency injection. Create a simple function called "processData" in the Processor interface and implement the logic in the SysProdProcessor as outlined in the Project Description above.

For reading CSV file, please use an Apache library which is compatible with Java 8.

For FreeMarker, Google Guice, Google Guava, please make use of the latest available version.

Input:

You are provided with 3 “transaction” batch files which you will use to run through the batch processing scenarios. Process the files in the following order: Transactions_001.csv, Transactions_002.csv and Transaction_003.csv.

Output:

You need to submit the following as output from your project:

Source code (in zipped format).

Screenshots of your application’s logs - take a screenshot after processing each file.

Output files from processing input files – this includes the CSV files as well as the generated HTML.

Please submit by 4th August 2023.

Good luck!