FORTER®

Conversion System @ Forter

Goal: Implement a basic conversion system

Note: This challenge should take up to **2-3 hours** of your time. We know you are busy and respect your time.

Background:

Our company needs a file conversion system - a system that converts given files from one format to another.

1. The system should receive a CSV formatted file and convert it to a JSON file.
   Each line in the CSV file represents a single rejected order.
   The json file should contain an array of JSON objects - one object per order.
2. When converting an order from the CSV file, the system should add new calculated fields for each order, as will be described below.
   A calculated field is a new field that is based on the existing order data.
3. The system will filter out duplicate orders based on orderId (order will be considered duplicated if it's orderId already exists in the same file). Duplicated orders will be saved in a different file (and should be converted as well). For example, if the CSV file contains 2 orders with orderId 123, one of them will be stored in the Orders JSON file and the second in the Duplicate Orders JSON file. Both orders should be stored after conversion. The same applies if there are more than 2 orders with the same orderId - all duplicated orders will be stored in the duplicated file.

Instructions

Implement a basic conversion system with the following capabilities:

- Converts order files from one format to another (CSV to JSON)
- Enrichs each order with additional, calculated fields
- Save converted orders in json file
- Filter out duplicate orders and store them to a different json file

Your task is to implement a basic conversion system that can perform the following:

Read a CSV file that contains orders, in which each order has the following properties (columns in the CSV):

- OrderId - Order identifier
- ReasonCode - A code representing why the order was rejected
- Amount - Order amount (see more information under currencies formatting section below)
- Currency - Order currency (see more information under currencies formatting section below)
- ProcessorName - The processor that charged this order (a processor is a financial company that manages the credit card transaction)
- DeliveryDate - The date for the delivery of the ordered item (see more information under dates formatting section below)
- OrderDate - The date in which the order was placed (see more information under dates formatting section below)
- MerchantName - The name of the merchant that sold the item
- Address - The customer address, the address to which the item will be delivered

Calculate the following fields for each order:

- ReasonCategory: depends on the reason code and processor of the order. You can find the mapping in the appendix directory (reason codes.csv). For example, reason code 10.4 with processor VISA will translate to ReasonCategory = FRAUD.
- AmountUSD: convert amount property of the order to USD. The exchange rates are: 1 EUR = ½ USD, 1 AUD = ⅓ USD. For example, order amount of 20 EUR will translate to amountUSD = 10
  You can assume that those are the only currencies (see more information under currencies formatting section below)
- ProcessingDate: three days after the order date. For example, OrderDate 10/3/2023 will translate to ProcessingDate = 13/3/2023

NOTE: The scope of this assignment is three merchants (MyShop, MyBook, MyFlight) and two processors (VISA, AMEX). You can assume that there are no other merchants or processors.

The json output should contain the following fields:

- OrderId - OrderId from the input CSV
- ReasonCode - ReasonCode from the input CSV
- Amount - Amount from the input CSV (formatted as described under inputs formatting)
- Currency - Currency from the input CSV
- ProcessorName - ProcessorName from the input CSV
- DeliveryDate - DeliveryDate from the input CSV (formatted as described under inputs formatting)

- OrderDate - OrderDate from the input CSV (formatted as described under inputs formatting)
- MerchantName - MerchantName from the input CSV
- Address - Address from the input CSV
- ReasonCategory - calculated field
- AmountUSD - calculated field (output currency format)
- ProcessingDate - calculated field (output date format)

Inputs formatting -
- DeliveryDate, OrderDate - The date input format is unique for each merchant and processor combination. You can find the mapping in the formatting appendix file. For example: the inputs dates for MyShop via AMEX is DD/MM/YYYY (15/12/2023)
- OrderAmount - The Currency input format is unique for each merchant and processor. You can find the mapping in the formatting appendix file.
  For example: the amount for MyShop via VISA is cent/10 currency unit so 5500 USD means 5.5$.

Outputs formatting -
- DeliveryDate, OrderDate, ProcessingDate - The desired output format is YYYY-MM-DD (2023-12-15)
- OrderAmount, AmountUSD - The output format is single currency unit (example: OrderAmount 60, Currency EUR means 60 euro, amountUSD 30 mean 30$)

Input & Output files -
Inputs file should be at “Inputs/{merchant_name}” directory under the root project directory.
Outputs file should be at “Outputs/{merchant_name}” directory under the root project directory.
For example:
- Input path: **/conversion_system/Inputs/MyShop/file1.csv**

Output paths:
- Converted Orders JSON file path: **/Users/conversion_system/Outputs/MyShop/file1.json**
- Converted Duplicate Orders JSON file path**:** **/Users/conversion_system/Outputs/MyShop/file1_duplicates.json**

Desired outcome:
You should develop a console (terminal) application.
The application should receive an input file as a command-line argument and print the output paths.

Important points for the exercise:
- You may choose any language to implement it
- You may use google or any other web resource, just not an already solved solution to the same problem

- You may use external libraries, just not an immediate solution for the converting csv to json
- <u>The solution must compile and run</u>
- You are encouraged to work iteratively. Make something work, then add more functionality as you go

Example (bold is what the user types in the console):

Please enter a file path for conversation:
/conversion_system/Inputs/MyShop/file1.csv

The Converted Orders JSON file located at:
*/Users/conversion_system/Outputs/*MyShop*/file1.json*

The Converted Duplicate Orders JSON file located at:
*/Users/conversion_system/Outputs/*MyShop*/file1_duplicates.json*

Good Luck