Software Engineer - Coding Assignment

Assignment Objective:

As part of your evaluation, we will need you to complete a coding assignment that will help us understand your fit with the opening at Unscrambl.

Description of assignment:

Assignment Environment:

1. This application populates and provides retrieval features for transactions of a company.
2. Transaction information is coming as files (let's say every 5 minutes) in a folder.
3. Another folder contains a file, which contains a reference data for products, against which the transaction are happening.
4. This application is an in-memory application so no persistent storage is required. i.e. You can reload the already available data in the transaction folder upon start-up of the application.
5. A transaction record contains following attributes in a comma separated format
    a. transactionId
    b. productId
    c. transactionAmount
    d. transactionDatetime
6. The product reference data have following attributes in a CSV.
    a. productId
    b. productName
    c. productManufacturingCity
7. Reference data is static and transaction data is keep coming in real-time in their respective folders.

Implementation Language:

Python or Java

Application functionalities:

1. Processes the data in streaming fashion, meaning provides up to date view for below mentioned REST calls.
2. Following REST APIs should be implemented.
    a. GET request http://localhost:8080/assignment/transaction/{transaction_id}
        i. Type: GET
        ii. Output data JSON: { "transactionId": 1, "productName": "P1", "transactionAmount": 1000.0, "transactionDatetime": "2018-01-01 10:10:10"}

b. GET request
http://localhost:8080/assignment/transactionSummaryByProducts/{last_n_days}
   i. Type: GET
   ii. Output data JSON: { “summary”: [{“productName”: “P1”, {“totalAmount”: 3000.0}]}
c. GET request
http://localhost:8080/assignment/transactionSummaryByManufacturingCity/{last_n_days}
   i. Type: GET
   ii. Output data JSON: { “summary”: [{“cityName”: “C1”, {“totalAmount”: 3000.0}]}

Note:
Rest call (a) will receive transactionId from input and provides details of transaction.
Rest call (b), (c) will receive lastNDays in the input data, application will provide summary of transactions during last 10 days from current date by product name or manufacturing city based on type of call.

Sample data format:

Transaction_20180101101010.csv:

transactionId, productId, transactionAmount,transactionDatetime
1, 10, 1000.0, 2018-10-01 10:10:10
2, 10, 1000.0, 2018-10-01 10:15:10
3, 20, 2000.0, 2018-10-01 10:15:20
4, 10, 1000.0, 2018-10-01 10:10:10
5, 30, 3000.0, 2018-10-01 10:20:10
6, 20, 2000.0, 2018-10-01 10:15:30

ProductReference.csv:

productId, productName, productManufacturingCity
10, P1, C1
20, P2, C1
30, P3, C2

What you’ll be judged on:

1. Completeness
2. Code quality
3. Design