PROCESSED FUTURE MOVEMENT Project

Overview:

System A has produced the file Input.txt, which is a Fixed Width text file that contains the Future Transactions done by client 1234 and 4321.

Requirements:

The Business user would like to see the Total Transaction Amount of each unique product they have done for the day The Business user would like a program that can read in the Input file and generate a daily summary report The Daily summary report should be in CSV format (called Output.csv) with the following specifications

The CSV has the following Headers - Client_Information - Product_Information - Total_Transaction_Amount

Client_Information should be a combination of the CLIENT TYPE, CLIENT NUMBER, ACCOUNT NUMBER, SUBACCOUNT NUMBER fields from Input file Product_Information should be a combination of the EXCHANGE CODE, PRODUCT GROUP CODE, SYMBOL, EXPIRATION DATE Total_Transaction_Amount should be a Net Total of the (QUANTITY LONG - QUANTITY SHORT) values for each client per product

Notes: Each Record in the input file represents ONE Transaction from the client for a particular product. Please focus on code re-usability.

Solution - Submission

Your submission should either be a Github repository on your personal Github page, or an archive that you submit to your HR contact.

A few requirements:

Part A - Java / Springboot. (Compulsory component for java roles).

1. Solution - Springboot and Java 8 or higher
2. Daily summary report - Rest API can give result either in JSON or download as CSV.
3. Output.csv - Provide a sample out of the result. (Without the need to run the solution.

Please complete part A before proceeding with part b.

Part B Containers (Additional component for the specialised role).

Please provide than the same solution above running in a container.

1. Docker or Docker Compose file.
2. Kubernetes of Openshift files is a bonus.

Demonstrate use of Java, Container

A few things to bear in mind if you take on this test:

1. We do not expect that you have familiarity with all of the tools and technologies used in this test (Java, Docker, Kubernetes, Openshift etc.). Part of the test is to see how you tackle an unfamiliar problem with an unfamiliar toolchain. Completing the test will require a combination of technical know-how, the ability to find and leverage useful resources and documentation (expect to use Google a lot!), and a great deal of creativity and the determination to follow through. We believe these are the primary attributes that will make for a successful team member!
2. There isn't a single right answer, and aren't many structured requirements. We are very much interested in how you approach and solve this problem. Our team needs engineers who can synthesize business problems into slick technical solutions that create business value without prescriptive oversight. This test will help us understand the kind of decisions you make and the kind of work you do, when given an open ended problem with intentionally few requirements.
3. Use Kubernetes or Openshift is a bonus

Grading Rubric

First and foremost, this is a design test. Given a set of constraints on input and output we are interested in the details of your implementation when you make all the decisions yourself. More specifically we will be looking at:

1. Unit tests: we would like to see unit tests
2. Coding style: clear and concise code is always preferred
3. Design decisions: this essentially a design test, in the sense that it is really asking you to design a system for capturing and presenting data via a specified interface. We are most interested not in which technologies you chose, but in why you chose them, and if the way that you used them demonstrated a clear understanding of that technology.