Exam - Customers & Purchases

Python Software Engineer

Context

We want to integrate in-flight payment, which will take place during the flight. The goal is to be able to purchase Duty-Free products or food (snacks, beverages, etc.). An airplane may not have continuous internet connectivity, so all customer data and purchases will be recorded in flat CSV files and then sent to Display's servers once the plane has landed, using 4G.

What to do

Your mission is to create a system that can read these files, interpret them, and send the data to the API of the ground processing system.

You will need to perform your development in Python, and you can use the framework of your choice.

You should create a command-line interface (CLI) that will read the two files present on the hard drive and send them to an existing REST API.

What is expected

Provide a repository (in the form of Git or a compressed archive) with all the essential resources (code, documentation, etc.). You can provide additional details in an email if necessary.

Resources

Two CSV files are created. They are simplified for this exam.

- "customers.csv"
    - The customer ID is unique
    - Mapping for the civility (File to API - “title” in the CSV) :
        - 1 -> Female
        - 2 -> Male
- "purchases.csv"
    - A customer can perform 0 to n purchases

Cloud API

The e-CRM has a REST API that only accepts JSON as the input format.
PUT Customer : https://myhostname.com/v1/customers

Important notice: This API URL is provided as an example and does not actually exist. Please perform your developments considering that the URL may be subject to easy changes between development, staging, and production environments.

JSON must be formatted as it:

```
[
      {
            "salutation": "xxx",
            "last_name": "xxxx",
            "first_name": "xxxx",
            "email": "xxxxx",
            "purchases": [
                  {
                        "product_id": "xxxx",
                        “price”: 10,
                        “currency”: “dollars”,
                        “quantity”: 1,
                        "purchased_at": "YYYY-MM-DD"
                  },
                  {
                        "product_id": "xxxx",
                        “price”: 10,
                        “currency”: “dollars”
                        “quantity”: 1,
                        "purchased_at": "YYYY-MM-DD"
                  }
            ]
      },
      {
            (customer 2…)
      },
      (...)
]
```