Problem Statement :

- We would like you to create an MVP for reporting API. This API will be used by our intranet Front-End to fetch some analytical data. To start with, we want you to work with a subset of [**booking data**].
- We are going to share with you data samples in a CSV file. [**file format**]
- We expect you to expose POST API for uploading more data in **JSON** format. [**API_1**].But your solution should be easily extendable for other input formats (CSV file, XML). If any of the records already exist we want them to be updated.
- Reporting API MVP should have 2 GET APIs with JSON response. **API_2**, **API_3**

**booking data: information about itineraries that have been purchased by our clients.*

File format:

hotel_id,booking_id,customer_id,selling_price_local_currency,currency,to_usd_exchange_rate

Data types:

hotel_id (integer), booking_id(integer), customer_id(string), selling_price_local_currency(double), currency(string), to_usd_exchange_rate(double)

Requirements for Solution :

- Readme (Explanation how to setup, Endpoints to hit)
- You can use any web-framework to set up the REST API endpoints.
- Please use CSV files or databases (dockerized ones) (SQL/ NO-SQL) to maintain the state.
- We recommend using batch insert/update, or multithreading to speed up **API_1.**
- Please use multithreading instead of DB aggregation to speed up **API_3.**
- We expect you to cover code with the necessary unit and functional tests.
- You will be given additional points for: Correct Validations, Logging, Configurable solution, exception handling, multithreaded solution(concurrency handled well).
- You can also choose any of the following constructs for design: OOPS/ Functional programming/CQRS
- Write your production-quality code in Java or Scala.
- Submit your solution by using the link that was included in the email.

Requirements for API :

API_1: Any URL POST

Request:

```
{
   "bookings": [
       {
           "hotel_id": 1,
           "booking_id": 123,
           "customer_id": "00000000-0000-0000-0000-000000000001",
           "selling_price_local_currency": 10.0,
           "currency": "EUR",
           "to_usd_exchange_rate": 1.21
       },
       {
           "hotel_id": 2,
           "booking_id": 124,
           "customer_id": "00000000-0000-0000-0000-000000000001",
           "selling_price_local_currency": 20.0,
           "currency": "MYR",
           "to_usd_exchange_rate": 0.25
       }
   ]

}
```

Response:

```
{
   "processed": 2
}
```

API_2: Any URL GET

Request Params:

hotel_id (int), current_to_usd_exchange_rate(optional, if no provided use from DB) (double)

Response:

```
{
   "number_of_bookings": 10,
   "total_price_usd": 1020.60
}
```

API_3: Any URL POST

Request:

```
{
  "customer_ids": [
       "00000000-0000-0000-0000-000000000001",
       "00000000-0000-0000-0000-000000000002"
   ]
}
```

Response:

```
{
   "results": [
       {
           "customer_id": "00000000-0000-0000-0000-000000000001",
           "number_of_bookings": 10,
           "total_price_usd": 1020.60
       },
       {
           "customer_id": "00000000-0000-0000-0000-000000000002",
           "number_of_bookings": 5,
           "total_price_usd": 520.20
       }
   ]

}
```