Exercise

Problem:

We have a list of all the orders to be processed daily which needs to be split by country and assigned to a courier which drives a Van and has a weight limit of 500KG. This list arrives daily in a CSV file.

Create a **Service A** providing an endpoint to receive the CSV file, process its data and then send it to a **Service B** to be persisted.

Regarding **Service A**'s data, the country needs to be determined based on the client phone number. You should consider the following regex to determine the country:

Cameroon	| Country code: +237 |	Regex = \(237\)\ ?[2368]\d{7,8}$
Ethiopia	| Country code: +251 |	Regex = \(251\)\ ?[1-59]\d{8}$
Morocco	| Country code: +212 |	Regex = \(212\)\ ?[5-9]\d{8}$
Mozambique	| Country code: +258 |	Regex = \(258\)\ ?[28]\d{7,8}$
Uganda	| Country code: +256 |	Regex = \(256\)\ ?\d{9}$

The **Service B** will store all data on its own database and provide an endpoint to generate a daily cargo manifest file.

Basically, the service will need to group orders to be delivered together based on the weight limit, so that they can be prepared in the warehouse and assigned to a courier.

The service should output a json file with a format as follows:

```
{
        cargos: [
                {order_ids: [1, 2, 3], total_weight: 450 }
                {order_ids: [4, 5, 6], total_weight: 470 }
                {order_ids: [7, 8, 9], total_weight: 400 }
                …
        ]
}
```

Make sure you include a simple readme so that we know how to boot up the project. On this last point, please make sure the project includes a dockerfile.

Topics to take into account

- **Code standards/clean code**
- Unit Tests
- **Performance and memory usage**
- **Data Partitioning**
- Documentation

Hints

- Don’t be shy on expressing and commenting your decisions
    - It’s important that we understand your ideas
- Fill free to choose the best language/framework/architecture that best suits
    - Golang would be a plus

Contact us if you have any questions regarding the requirements or the problem itself!
Make sure that you understand everything that is requested!