Tax Calculator

Preparation

Before starting, you will need:

- Git
- Your own dev setup
- Docker for deployment
- docker-compose
- 6 hours of your time

The Exercise

For this exercise, you will be creating a set of APIs to be used by front-end engineers to develop an application that store and display tax amounts.

Please use the following tech stack:

- Any backend language and framework you prefer -- Python, Go, Ruby, Java, NodeJS...
- Docker for deployment
- MySQL or PostgreSQL

Since docker-compose is used to manage the application and database, only `docker-compose up` should be required to start the server.

User Stories

As user I want to create my tax object

The form UI may look as such:

Fields	Example Input
Name	Big Mac
Tax Code	1
Amount	1000

Tax Codes can be statically assigned:

- 1 = food
- 2 = tobacco
- 3 = entertainment

As user I want to see my bill

Name	Tax Code	Type	Amount	Tax Amount	Total Amount
Lucky Stretch	2	Tobacco	1000	100	1100
Big Mac	1	Food	1000	20	1020
Movie	3	Entertainment	150	0.5	150.5

- Total Amount: 2150
- Total Tax Amount: 120.5
- Grand Total: 2270.5

Calculating Tax Amount

food

- 10% of `value`

tobacco

- 10 + (2% of `value`)

entertainment

- 0 < value < 100: tax-free
- value => 100: 1% of (`value` - 100)

Evaluation Checklist

As this exercise is a very simple one, the functional correctness of this exercise is secondary. It should be a given that you will be able to get the correct outputs from above. Therefore, to make your work really stand out we look at the following things:

- Code quality & readability: Will any engineer be able to understand the execution just by briefly scanning through the tests and source code?
- Software design: Does the implementation make full use of classes, objects, functions, abstractions, interfaces, etc.
- Engineering best practices: Does it follow proper architectural patterns (MVC), and SOLID principles?
- Any automated tests (e2e, integration, unit, etc.)

and *NOT*:

- Fancy UI.

Submission

Once you have completed the exercise, please push the git repository to a host of your choice, preferrably GitHub. Your Dockerfile and code should be sufficient for us to recreate and test your API.

Please submit the following items:

- Git repository for your code (including Dockerfile)
- API documentation (that FE dev is gonna make use)
- Database design documents (DB structure and explanation)