🧪

Skill test

Introduction

Thank you for taking the time to do this skill test. We really appreciate your dedication.

It should take **3 hours** to go through this test. Feel free to scope it in a way you are comfortable with, every deliverable, even incomplete, is helpful to us to understand how you approach a problem.

When finished, please submit your code in a **compressed archive zip via email**.

Goal

At TreasuryX, one of the main challenges we have is to expose a clean, easy to use API to our customers that abstracts heavy protocols and transport layers complexity.

The goal of exercise is to expose a secure web service that takes one payment request in JSON format and perform the following:

1. Allow only authenticated requests, we will be using **basic HTTP authentication**, we are expecting the following credentials
    - username: Your last name uppercase
    - password: xxxx
2. Validate the request according to the provided JSON schema and example
3. If the request is not authorized or not valid, return the corresponding HTTP error
4. Store the request data in Sqlite database

5. Generate the payment XML file and depose it in the bank folder, an HTTP 200 response is expected if this succeeds
6. After the file deposit by few seconds we expect the bank to respond by depositing a response file containing the ID and the status of payment (PROCESSED or REJECTED), we expect the payment status in the database to be updated accordingly.

Instructions

It's recommended to use Go programming language

You can access the Internet for information.

You can use open source libraries when relevant

Please provide a README.MD explaining your approach, choices and the overall solution

Time

We expect the test to take less than 3 hours, if you would like to take more, explain the motivation behind in the README.md file

Test resources

Please download the following archive: https://drive.google.com/drive/folders/1Tj-BmfBcxLOXrbXw4LgHBGLkGcoeZyke?usp=sharing

You can find the following files to help you on your way:

- `request_sample.json` : the expected payment request,
- `request_schema.json` : the corresponding JSON schema.
- `payment.xsd` : schema of the xml expected by the bank.
- `payment_sample.xml` : sample xml file payment as expected by the bank
- `bank_response_sample.csv`: sample bank response file

Environment variables:

- `BANK_FOLDER`: bank folder where to deposit payment file and load response
- `SQLITE_DB_FILE_LOCATION`: Location of the Sqlite database

Feedback

We would like to have your feedback regarding the test by completing the fallowing:

1. How much time did you spend performing this exercise?

   □ Less than 2h

   □ 2h to 3h

   □ 3h to 4h

   ☒ 5h to 6h

   □ More than 6h (your time: ________)

2. How proud are you of your work?

   □ Very proud

   ☒ Fairly proud

   □ Good enough

   □ Somewhat disappointed