Ope Project Documentation

INTRODUCTION

This is the backend infrastructure build for Ope App, a stock portfolio web application with loan management features. The Application features Golang, Gorilla Mux, Uba zap, MYSQL, and JWT-GO. Its application functionalities can be tested using the Ope Demostration Interface, or on any test client such as Postman.

To interract with the application core resources, a user is required to REGiSTER and verified his/her account. After registering, the user can login to the application using their registered credenials. This will provide the user an hour long authorization token that can be used to access the API core resources.

API ARCHITECTURE

Ope application is modeled after Hexagonal/Port-Adapter architecture. Apart from its global packages, like the custom Logger, securities, utils, and middleware, the application can be seen as consisting of three layers, viz: the core/business logic, the client-side-Ports, and the server-side Ports. All of which are wired using the principles of dependency injection. This makes the testing unit easier. Plus, it avails more flexibility with database swaps.

AUTH

To add an extra layer of security, a 2FA system is included in the auth workflow; users are sent OTP to complete each login and major financial operations on the application. Hence, the login process involves the client sending request to Login API, and Complete-Login API. Other user related APIs include, Profile-Update, Password-Change, and Bank-Details-Update.

WALLET SYSTEM

All transactions on the app are routed through user's wallet. For instance, to buy investment, a user has to fund his/her Ope wallet from which they can trade. Same is the case when requesting for loans but in reverse order. This reduces the number of times third party payment gateway will be called. Which in turn, reduce transaction charges. Plus, it makes transactions within the application reflect faster as no third party calls are needed. SEE DETAILS ON WALLET FUNDING API AND WORKFLOW

INVESTMENT

Once a user's wallet has been successfully funded, it can then start to trade investments in company stocks, Repay loans or withdrawn to user's registered bank account. The amount of stocks that can be bought by user is only limited by the user and the amount available in his/her wallet.

To buy investment, the client should call the endpoint as DETAILED HERE

LOAN MANAGEMENT

The loan management API is categorized into two, LOANS and loan REPAYMENT API.

For context, the user can take a loan as long as he/she doesn't have any open/active loan prior, and the proposed loan must not exceed 60% of the user's total investment in stocks.

Depending if loan has been fully repaid, a loan could be open or closed. It is open when the user is yet to conclude repayments on the loan and closed when repayment has been completed on the loan. Repayments are only possible for loans with open status.

Upon taking a loan, the user can then make REPAYMENTS in installments. These installments are called REPAYMENTS and are gonverned by the REPAYMENT API

On Successful loan request, the user wallet is credited with the requested amount which can then be withdrawn to the user registered bank account.

CHECKERS

In the course of repayment in installements, if a user attempt to pay an amount greater than the loan balance, the system checker will ensure that only the loan balanced is lessed from the user's wallet balance.

If the user request a loan greater than 60% of her investment in stocks, the checker program will flag the transaction attempt with a 406 error code along with a detailed error message.

- Loan API

To request a loan, the client should make request as detailed below

```
URL : https://be-ope.herokuapp.com/loan/request
METHODE TYPE: POST
Authorization: Bearer token (Any valid TOKEN of user that is not PAYMENT TOKEN)
PAYLOAD: Check addendum for Loan Request Payload

To Fetch user's loans, both open and closed, client should request as below
URL : https://be-ope.herokuapp.com/loans
METHODE TYPE: GET
Authorization: Bearer token (Any valid TOKEN of user that is not PAYMENT TOKEN)
```

- Repayment API

To repay a loan in installments, the client should send a request as detailed below

```
URL : https://be-ope.herokuapp.com/payment/loan/{loanId}
METHODE TYPE: POST
Authorization: Bearer token (Any valid user token that is not PAYMENT TOKEN)
PAYLOAD: Check addendum for Loan- Repayment Payload
```

API DOCUS

Register API | Login API | Complete-Login API, Login Sample Response | Funding API | Payment Response | Complete Funding API | Investment API | Request Loan API | Fetch Loans API | Repayment Api