OPE API Documentation

Introduction

This is the backend infrastructure build for Ope App, a stock portfolio web application with loan management features. It was built as part of Trove R challenge process.

STACKS AND TECHNOLOGY

To

TO UPDATE USER PROFILE

```
URL : https://be-ope.herokuapp.com/update-profile/{userId}
```

METHODE TYPE: Patch Authorization: Bearer token (Any valid user token that is not PAYMENT TOKEN)

```
PAYLOAD:
{
"firstname":"string",
"lastname":"strin",
"email": "string",
"phone": "string",
"account_no": "string",
"bank_name": "Zenith Bank"
}
```

TO UPDATE ONLY BANK DETAILS

```
URL: https://be-ope.herokuapp.com/user/bankupdate/{userId}
```

METHOD TYPE: Patch Authorization: Bearer token (Any valid user token that is not PAYMENT TOKEN)

```
PAYLOAD:
{
"bank_name":"Zenith Bank",
"account_no":"2085394463"
}
```

FUNDWALLET FLOW

Stage ONE

1 - Call FUND WALLET EndPoint

2 - Backend will respond with PAYMENT-BODY and PAYMENT-TOKEN

3 - Call FLutterwave gateway (Preferably Flutterwave inline)

4 - Conclude Flutterwave transaction

Stage TWO

1- Call COMPLETE-FUNDING EndPoint and send Back PAYMENT_TOKEN sent earlier

2- GET response from Backend Server.

DETAILS:

```
FUND-WALLET ENDPOINT : "https://be-ope.herokuapp.com/fund-wallet"
METHOD TYPE: POST
Authorization: login token or any active non-payment token
Payload:
     {
    "amount":"18000",
    "currency":"NGN",
    "payment_option":"card"
}
```

```
COMPLETE-FUNDING ENDPOINT : "https://be-ope.herokuapp.com/complete-funding"
METHOD TYPE: PATCH
Authorization: Payment-token (You can only use payment token for this)
Payload:
     {
    "tx_ref": "James-tx-724701",
    "amount": "18000",
    "wallet": "$2a$06$WRKRFEsAP/meZbjMP1lkOuzyu7jtZ66cu8uH0dQZPKP3pwzDwYRvi"
}
```

PLEASE NOTE:

1 - tx-ref is transaction reference. It will be the first attribute of the PAYMENT_BODY sent to you when you call FUND_WALLET endpoint

2 - Amount has to be the same as the one in the PAYMENT BODY

3 - wallet is the user's wallet address. It will also be sent as "customer_mac" under the "meta" attribute of the PAYMENT-BODY

SAMPLE RESPONSE

```
{
    "status": "success",
    "collection": "User",
    "data": {
        "user_id": "5",
        "firstname": "Aghahowa",
        "lastname": "Ogie",
        "email": "gulephil44@gmail.com",
        "created_at": "2021-11-15 00:52:42",
        "bank_account": {
            "account_no": "2085394463",
            "bank_name": "Zenith Bank"
        },
        "otp": "",
        "wallet": {
            "amount": 54000,
            "address":
"$2a$06$WRKRFEsAP/meZbjMP1lkOuzyu7jtZ66cu8uH0dQZPKP3pwzDwYRvi"
        },
        "token": "",
        "portfolio": [
            {
                "id": "5",
                "symbol": "AWS",
                "image_url": "https://buyshares.co.uk/wp-
content/uploads/2020/07/Screenshot-2020-07-04-at-16.49.55.png",
                "quantity": "120",
                "unit_price": "1200",
                "equity_value": "144000",
                "percentage_change": "-9"
            },
            {
                "id": "15",
                "symbol": "TSLA",
                "image_url":
"https://g.foolcdn.com/art/companylogos/square/tsla.png",
                "quantity": "25",
                "unit_price": "800",
                "equity_value": "20000",
                "percentage_change": "-1"
            }
        ]
    }
}
```

INVESTMENT

Once a user's wallet has been successfully funded, it can then be used to buy 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 below.

```
URL : https://be-ope.herokuapp.com/buy-stock/{userId}
METHODE TYPE: POST
Authorization: Bearer token (Any valid TOKEN of user that is not PAYMENT TOKEN)
PAYLOAD:
{
    "symbol":"FB",
    "image_url":"https://cdn3.vectorstock.com/i/1000x1000/02/37/logo-facebook-
vector-31060237.jpg",
    "quantity" : "2",
    "unit_price" : 341.13,
    "percentage_change": 1.57
}
```

LOAN MANAGEMENT API

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.

Depeending 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 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 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:
{
    "amount" : numerical e.g 32435.00
    "duration" : number of months (Numerical) e.g. 8
}
```

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:
{
    "loan_id": "string"
    "payment": nerical (1200.00)
}
```