FRONT

Routes

Registration => registruj se

Login => uloguj se

Profile => pregled i izmena profila

Wallet => pregled stanja (**Account**), uplati sredstva (**Deposit**), posalji sredstva (**SendPayment**), razmeni sredstva (**Exchange**)

History => istorija transakcija (**Filter**, **Sort**?)

ExchangeRate => kursna lista

PUTANJE

FRONT: localhost:3000
BACK: localhost:5000

Iz perspektive back-a, kakve njegove metode treba da budu, sta da vracaju:

Login =>

method: POST
path: "/user/login"
payload: {email: string, password: string}
return: {user: object with all the user data}, OK / {}, 401

Verify =>

method: PUT
path: "/user/verify"
payload: {email: string, card_details: (card_number: int, card_name: string, card_expiration_date: string, card_security_code: int)}
return: {}, OK / {}, 401

Register =>

method: POST
path: "/user/register"
payload: {email}
return: {}, OK

ChangeProfile =>

method: PUT
path: "/user/profile"
payload: {email: string, user: new user data}
return: {user: object with updated data}, OK

ChangeWallet =>

method: GET
path: "/user/wallet"

payload: {email: string}
return: {wallet: six fields, same names as they are in model}, OK

Deposit =>
method: POST
path: "/transaction/deposit"
payload: {email: string, amount: double}
return: {}, OK

SendPayment =>
method: POST
path: "/transaction/send"
payload: {email_sender: string, email_receiver: string, amount: double, currency: string (names from wallet model)}
return: {}, OK

Exchange =>
method: POST
path: "/transaction/exchange"
payload: {email: string, amount: double, currency_from: string (names from wallet model), currency_to: string (names from wallet model)}
return: {}, OK

History =>
method: POST
path: "/transaction/history/{transaction_type}/{sort_value (if none send "default", else column name from model)}/{asc_desc (boolean: desc is false, asc is true)}/{search_value (if none send "default" else input text)}"
payload: empty
return: {transactions: [(e.g. json objects, each contains => fields from deposit: corresponding values)]}, OK

ExchangeRate =>
method: POST
path: "/crypto"
payload: empty
return: eg *{'bitcoin': {'usd': 3461.27}, 'ethereum': {'usd': 106.92}, 'ripple': {'usd': 106.92}, 'tether': {'usd': 106.92}, 'dogecoin': {'usd': 106.92}}*
, OK

NAPOMENE

* Koristiti socket.io da bi server obavestio klijenta kada se zavrsilo majnovanje.
Da bi back znao kada se zavrsilo majnovanje moze samo da pokrene jedan thread koji ce da ima timeout kojim ce se simulirati majnovanje i posle majnovanja da pozove od socket.io funkciju koja ce da obavesti servera.

* Back ne sme da ispunjava zahteve od nevalidiranog klijenta, treba da postoji zastita koja ce da onemoguci da nevalidirani klijent dobije uslugu jer na frontu ne moze da postoji adekvatna zastita za to.

* Da bi vratio data u flasku treba da uradis samo => return data, 200
Gde je 200 status code. Ima i komplikovanije ali ne znam da li je potrebno:

*Front rucno menja is_verified polje ako verifikacija vrati 200 OK

```
from flask import Flask, json

@app.route('/login', methods=['POST'])
def login():
    data = {"some_key":"some_value"} # Your data in JSON-serializable type
    response = app.response_class(response=json.dumps(data),
                                  status=200,
                                  mimetype='application/json')
    return response
```