**NKHH BANK**

==========================================================

**LINK APP HEROKU:**

**acc mặc định trên app: admin/khongcopass**

**acc email trên heroku:** [**kien54176@gmail.com**](mailto:kien54176@gmail.com)**/kien54176@@N**

**heroku connectionString:**

**BE và FE chỉ nhận các loại format sau đây**

**toàn bộ ngày tháng xài format:YYYY-MM-DD**

**toàn bộ thời gian xài format: HH:mm:ss**

**ngày tháng thời gian xài: YYYY-MM-DD HH:mm:ss**

**(HH là dạng 24h)**

**Gmail:**

[**kien54176@gmail.com**](mailto:kien54176@gmail.com)

**kien54176@@N**

**người chưa đăng ký/đăng nhập:**

**đăng ký: /**[**api/**](https://side-web2020.herokuapp.com/api/)**register**

method: POST

input:

email, username, fullName, cardId, cardType( tạo combobox CMND hoặc CCCD gửi sang BE), date(ngày cấp CMND YYYY-MM-DD),address, birth (YYYY-MM-DD), password, password\_retype

các output:

thành công: ‘ok’

trống data truyền vào: ‘empty’

username và cardId truyền không đúng dạng: ‘wrong type’ (username và cardId phải viết liền, ko có dấu cách)

password không giống nhau: ‘not equal’

email đã có người dùng: ‘email used’

cmnd đã có ng dùng: ‘cardId used’

username đã có người dùng ‘username used’

mô tả thêm:

khi đăng ký thành công, mã xác nhận sẽ được gửi qua email, nếu người dùng kích hoạt thì tình trạng emailVeirfy = 1

tình trạng duyệt CMND vẫn = 0, phải chờ nhân viên duyệt

---------------------------------------------------------------------------------------------------------------

**đăng nhập: /api/login**

method: POST

input:

username, password

req.session.currentUser phải trống mới gọi dc

output:

thành công :’ok’

thất bại: ‘fail’

yêu cầu kích hoạt email: ‘email’

khi đăng nhập thành công sẽ lưu user vào session (req.session.currentUser)

---------------------------------------------------------------------------------------------------------------

**xác nhận email với mã: /api/verify-email**

method:GET

input: code

ví dụ:

localhost:3000/api/verify-email?code=12345

output:

thành công :’ok’

thất bại: ‘fail’

---------------------------------------------------------------------------------------------------------------

**gửi mã khi chưa login : /api/send-code**

method: POST

input: email

gửi mã cho user, mã dùng chung cho kích hoạt email, quên mật khẩu, 2 bước...

output:

thành công :’ok’

thất bại(không tìm thấy, ng dùng nhập email ngu ): ‘fail’

req.session.currentUser phải trống mới gọi dc

---------------------------------------------------------------------------------------------------------------

**nhập mật khẩu mới dựa vào mã: /api/new-password**

method: POST

input: code, password, password\_retype

req.session.currentUser phải trống mới gọi dc

output:

thành công :’ok’

sai mã: ‘wrong code’

pass ko giống nhau : ‘not equal’

trống dữ liệu vào ‘empty’

nếu người dùng chưa xác nhận email, dùng thành công tính năng này thì trạng thái email sẽ được bật

==============================================================

**người dùng đã đăng nhập:**

***cơ bản:***

**đăng xuất: /api/logout**

method: POST/GET

input: không

req.session.currentUser phải có mới gọi dc

ouput:

thành công :’ok’

--------------------------------------------------------------------------------------------------------------

**gửi mã khi đã login : /api/resend-code**

method: POST

input: không có

gửi lại mã cho user, mã dùng chung cho kích hoạt email, quên mật khẩu, 2 bước...

output:

thành công :’ok’

thất bại(không tìm thấy, currentUser ngu): ‘fail’

--------------------------------------------------------------------------------------------------------------

**người dùng lấy thông tin cá nhân: /api/info**

method:GET và POST

input: không (sử dụng req.session.currentUser.id để search)

req.session.currentUser phải có mới gọi dc

output:

trả ra thông tin người dùng trong bảng user nối với bảng card, che vài cái

---------------------------------------------------------------------------------------------------------------

**người dùng tự update vài thông tin bản thân: /api/update-self**

method: POST

input: fullName, birth, address, cardId, date, cardType, email

* nếu để trống thì ko update thông tin đó (truyền ‘’, null, hoặc không khai báo params)
* để dữ liệu như cũ thì ko update thông tin đó
* nếu update email mới thì check có ai xài chưa, nếu update dc thì gửi tới email cũ là đổi thành email mới nào, giữ veriyEmail =1
* nếu update cardId thì đổi verifyCardId = 0, nhớ là update bảng user và cả bảng card

req.session.currentUser phải có mới gọi dc

nếu update thành công, cập nhật lại req.session.currentUser

const userWithNewInfo = await USER.findOne()....

req.session.currentUser = userWithNewInfo

output:

thành công :’ok’

email đã có người dùng: ‘email used’

cmnd đã có ng dùng: ‘cardId used’

--------------------------------------------------------------------------------------------------------------

**người dùng update mật khẩu: /api/change-password**

method: POST

input: password\_old, password, password\_retype

req.session.currentUser phải có mới gọi dc

output:

thành công :’ok’

pass ko giống nhau : ‘not equal’

---------------------------------------------------------------------------------------------------------------

**người dùng lấy lịch sử giao dịch: /api/history**

METHOD: GET và POST

input: start, limit và accountId (accountId có thể để trống để trả ra tất cả)

search where accountId\_1 == accountId or accountId\_2 == accountId

req.session.currentUser phải có mới gọi dc

giải thích:

* thằng user A, có STK 123, 456, 789
* thằng user B có STK 333,444,555

=> khi thằng A đăng nhập nếu ko nhập accountId tìm gì, thì trả ra toàn bộ lịch sử của STK 123, 456 ,789. nếu thằng a nhập accountId 123 thì chỉ trả ra lịch sử STK của STK 123.

Lưu ý ko hiện lịch sử STK của người dùng khác dù có nhập accountId hay ko

output:

trả ra lịch sử toàn bộ STK của user này + count

hoặc lịch sử của 1 STK nhất định của user này + count

---------------------------------------------------------------------------------------------------------------

***tài khoản:***

**người dùng lấy ra danh sách STK dạng array: /api/account-array**

METHOD: GET VÀ POST

input: accountType (truyền 0 để lấy array các STK tiết kiệm, truyền 1 để lấy array STK thanh toán), bỏ trống thì mặc định là 1

output:trả ra danh sách STk dạng list [123,456,789,111111...]

**--------------------------------------------------------------------------------------------------------------**

**người dùng lấy danh sách STK của mình: /api/account-list**

METHOD: GET VÀ POST

input: start, limit để phân trang (sử dụng req.session.currentUser.id để search STK theo userId)

req.session.currentUser phải có mới gọi dc

output:

trả ra danh sách tài khoản của người dùng này + count

---------------------------------------------------------------------------------------------------------------

**người dùng/nhân viên lấy thông tin 1 STK: /api/account-info**

METHOD: GET và POST

input: accountId

req.session.currentUser phải có mới gọi dc

output:

trả thông tin của tài khoản truyền vào

---------------------------------------------------------------------------------------------------------------

**người dùng chuyển khoản: /api/transfer**

METHOD: POST

input: accountId\_1 (bên gửi), accountId\_2 (bên nhận), amount, message(được để trống), code

trước khi chuyển khoản, bấm nút “lấy mã xác minh” ( gọi api resend-code)

sẽ gửi mail cho 2 bên nếu thành công

phí chuyển: bé hơn 10tr là free, từ 10tr là 1%, từ 30tr là 2%

output:

thành công: ‘ok’

sai mã so với tài khoản đang đăng nhập: ‘wrong code’

accountId\_2 không tồn tại hoặc là tài khoản tiết kiệm: ‘accountId\_2 not exists’

accountId\_2 Đang bị khóa: ‘accountId\_2 locked’

accountId\_1 đang bị khóa: ‘accountId\_1 locked’

tiền nhập vào bị giới hạn mỗi cuộc giao dịch: ‘limit’ (giới hạn là 50 triệu)

tiền nhập vào lớn hơn số dư tài khoản: ‘over balance’

STK rút không thuộc sở hữu user đang đăng nhập hoặc STK rút là STK loại tiết kiệm, ko chuyển được: ‘not owner’

---------------------------------------------------------------------------------------------------------------

**người dùng rút tiền: /api/withdraw**

method: POST

input: accountId, code

để tính lãi hiện lên FE (lãi dự tính, gọi api/profit bên dưới)

trước khi rút tiền, bấm nút “lấy mã xác minh” (gọi api resend-code)

* nếu tài khoản thanh toán:
* không được rút
* nếu tài khoản tiết kiệm:
* tính lãi: 12,24,36 (3-5%, 4-6%, 5-7% / 360 ngày) - (x% trước kỳ hạn, y% đúng kỳ hạn)
* tiền lãi tối đa là y%, nếu ngày đã gửi vượt quá kỳ hạn đăg ký trước cũng y%
* tính lãi từ ngày gửi + lời => gửi email
* xóa tài khoản sau khi rút

tính ngày khoảng cách bằng moment. diff

output:

thành công: ‘ok’

sai mã so với người đang đăng nhập hoặc mã không tồn tại: ‘wrong code’

accountId đang bị khóa: ‘locked’

STK rút không thuộc sở hữu user đang đăng nhập hoặc STK rút là STK loại thanh toán, ko rút được: ‘not owner’

---------------------------------------------------------------------------------------------------------------

**Tính tiền lãi dự tính: /api/profit**

method: POST / GET

input: accountId

output: trả ra lãi dự tính của tài khoản này nếu nó là tài khoản tiết kiệm

==============================================================

**Tính phí chuyển dự tính: /api/fee**

method: GET / POST

input: accountId, amount

output: trả ra phí dự tính chuyển khoản

**nhân viên:**

***cơ bản:***

**NV lấy danh sách người dùng (có search): /api/list-user**

METHOD: POST và GET

input: search(email người dùng), cardIdVerify(để lấy danh sách duyệt/ chưa duyệt), start, limit

nếu cardIdVerify ko truyền thì trả ra toàn bộ người dùng được duyệt và chưa duyệt

nếu ko truyền vào email thì trả ra toàn bộ ng dùng

trả ra danh sách người dùng + count

---------------------------------------------------------------------------------------------------------------

**NV lấy thông tin cá nhân 1 user: /api/info-user**

METHOD: POST và GET

input: id (id người dùng)

trả ra thông tin 1 người dùng duy nhất

nếu không tìm thấy thì trả ra ‘fail’

---------------------------------------------------------------------------------------------------------------

**NV thay đổi thông tin cơ bản người dùng: /api/update-user**

METHOD: POST

input: id (id người dùng), status, cardIdVerify, fullName, birth, address, permission, email

trường hợp đổi email user thì gửi mail thông báo cho user (mail cũ và mail mới THÔNG BÁO, KO GỬI MÃ), set emailVerify =1 luôn

ouput:

thành công: ‘ok’

email đã có người dùng: ‘email used’

---------------------------------------------------------------------------------------------------------------

***tài khoản:***

**NV tạo STK cho user: /api/create-account**

METHOD: POST

input: id(id người dùng), accountType(1 là thành toán, 0 là tiết kiệm), balance, currencyType(VND hoặc USD), term (nếu tạo tài khoản tiết kiệm mới thêm term, chỉ 12 24 và 36)

output:

thành công: trả ra STK vừa tạo được (1 chuỗi random 10 số)

thất bại: ‘fail’

---------------------------------------------------------------------------------------------------------------

**NV nạp tiền cho user: /api/load-up**

METHOD: POST

input: accountId(STK cần nạp), balance, currencyType (VND hoặc USD)

hệ thống tự chuyển tiền tệ nạp thành đơn vị của STK cần nạp nếu cần (ví dụ STK xài VND, nạp USD thì tự quy ra rồi nạp vào)

Khi nạp tiền:

* nếu tài khoản được nạp là tài khoản thanh toán: tăng tiền bình thường, đang khóa thì mở lại + update ngày đóng + ngày mở
* tài khoản tiết kiệm không cho nạp, chỉ được nạp 1 lần lúc tạo

output:

thành công: ‘ok’

thất bại: ‘fail’

---------------------------------------------------------------------------------------------------------------

**NV lấy danh sách tài khoản của 1 user: /api/account-list-user**

METHOD: GET / POST

input: id(id của người dùng), start, limit

output:

trả ra danh sách tài khoản của người dùng này + count

---------------------------------------------------------------------------------------------------------------

**NV lấy lịch sử giao dịch của 1 user: /api/history-user**

METHOD: GET / POST

input: id(id người dùng hoặc accountId), params: start, limit

tham khảo api/history để hiểu thêm

id hiểu như sau:

* nếu id là id ở bảng user thì trả ra toàn bộ lịch sử toàn bộ STK của ng dùng này
* nếu id là accountId ở bảng account thì trả ra toàn bộ lịch sử của STK này
* nếu không truyền thì findall

trả ra lịch sử có phân trang + coung (có thể tái dụng hàm của api/history)

**Hệ thống**

Tự động thêm vào bảng history khi 1 STK::

* Nạp tiền
* chuyển tiền
* rút tiền

==========================================================

**THÔNG TIN THÊM**

tiền tệ: USD và VND, set cứng 1 USD = 23 000 VND

giải thích các quy trình:

* đăng ký -> gửi mã kích hoạt -> đăng nhập -> nhập mã (nếu mất mã có thể ấn nút gửi lại ở đây)
* quên MK: nhập email ở quên MK -> chờ mã => nhập mã + mật khẩu mới + xác nhận mật khẩu mới(nếu người dùng chưa xác nhận email, xài thành công tính năng này thì email sẽ được kích hoạt, khỏi phải nhập mã kích hoạt email)
* sau khi đăng nhập: tự update thông tin cá nhân, nếu update mã CMND thì phải chờ duyệt, update email thì phải vào email mới kích hoạt, upload ảnh...
* mở tài khoản ngân hàng: người dùng phải chờ nhân viên duyệt, nhân viên sẽ mở tài khoản cho người dùng
* tài khoản thanh toán: chuyển cho người khác, nhập STK người khác, mã 2 lớp và giá trị muốn chuyển. tài khoản thanh toán có thể rút tiền, nhập giá trị muốn rút và mã 2 lớp rồi rút
* tài khoản tiết kiệm: bấm rút tiền và mã 2 lớp, sẽ rút vốn + lãi, tài khoản tự đóng chờ nạp tiền để mở lại

giải thích các bảng trong database:

bảng user:

* staus: 0 là khóa,1 là bình thường
* emailVerify: 0 là chưa xác nhận email, 1 là rồi
* cardIdVerify: 0 là chưa được duyệt cmnd, 1 là rồi
* userCode: mã dùng chung cho kích hoạt email, quên MK, 2 lớp
* permission: 0 là người dùng, 1 là nhân viên
* username sẽ không được thay đổi bởi nhân viên hay người dùng. cố định

bảng card:

* cardType: CMND hoặc CCCD
* date: ngày cấp CMND (YYYY-MM-DD)
* URL: dường dẫn ảnh CMND lưu trong server

account:

* accountId: hệ thống sẽ tự tạo 1 số ngẫu nhiên trong lúc nhân viên yêu cầu tạo STK cho người dùng nào đó
* userId: chỉ số này giải thích accountId thuộc về người dùng nào
* status: 0 hoặc 1, mặc định là 1. 1 mở, 0 là khóa
* currencyType: đơn vị tiền tệ, VND hoặc USD
* accountType: 1 là tài khoản thanh toán, 0 là tài khoản tiết kiệm, mặc định là 1
* term: chỉ áp dụng cho tài khoản thanh toán, term chỉ nhận các chỉ số 12, 24 và 36 (tháng)
* startDate: ngày mở TK
* closeDate: ngày đóng TK

**DATABASE**

**heroku connectionString:**

