Get and Posts

ENVIAR NO BODY VAI EM (FORM-DATA) RETORNA EM (JSON)

O PATH DE CONTROLE VAI NO BODY

<?php

if($\_SERVER["REQUEST\_METHOD"]=="POST"){

echo $\_POST[‘path’];

}

?>

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

ENVIAR NA URL

http://localhost:8080/?path=(string função ou local)&nome=carlos&...(restante dos paramentros)

O PATH DE CONTROLE VAI NA URL

<?php

if($\_SERVER["REQUEST\_METHOD"]=="GET"){

echo $\_GET['path'];

}

USUÁRIO

Tipo: POST -path=login-

Uso: verifica a existência do usuário no banco de dados e faz login

Params: não tem

Body: string email\_nickName, string password

Return: user:{id,nickName:'string',email:'string'}, token:'string' OU error='string'

Tipo: POST -path=Register-

Uso: Executa o registro do cliente no banco de dados

Params: não tem

Body: all string (name,nickName, email, date,passoword, passwordConfir)

Return: string ‘success’ OU error='o que ocorreu

Tipo: POST -path=deleteUser-

Uso: Apllica um soft delete no usuário

Params: não tem

Body: user: id

Return: string ‘success’ OU error='o que ocorreu'

Tipo: POST -path=updateUser-

Uso: Executa uma atulização no registro do cliente no banco de dados

Params: não tem

Body: all string (name,nickName, email, date,password,newpassoword, passwordConfir)

Return: string ‘success’ OU error='o que ocorreu'

CONTATOS - CONVITES

Tipo: GET -path=contactInvitations-

Uso: Recupera uma lista de contatos que enviaram convite

Params: user:id

Body: não tem

Return: ListaContatos[{id,nickName,name}] OU error='o que ocorreu'

Tipo: POST -path=contactAccept-

Uso: Aceita o convite do contato

Params: não tem

Body: user: id, contact:id

Return: ‘success’ OU error='o que ocorreu

CONTATO DELETANDO

Tipo: POST -path=deleteContact-

Uso: deleta um contato

Params: não tem

Body: user: id, contact:id

Return: ‘success’ OU error='o que ocorreu'

CONTATOS - STATUS

Tipo: GET -path=contactStatus-

Uso: Recupera uma lista de contatos que pertençam ao usuário

Params: user:id

Body: não tem

Return: ListaContatos[{id,nickName,name,online,sendMessage}] OU error='o que ocorreu'

sendMessage(é usado caso o contato tenha enviado mensagem que o user ainda não viu)

CONTATO - ADICIONANDO

Tipo: POST -path=sendContactInvitation-

Uso: Envia um convite ao contato

Params: não tem

Body: user: id, contact:nickName

Return: ‘success’ OU error='o que ocorreu'

CONTATOS NA REUNIÃO

Tipo: GET -path=contactInThisMeeting-

Uso: Recupera um lista de contatos que foram convidados para

a reunião e aceitaram o convite e estados na reunião ex: ausente - na reunião

Params: user:id, meeting:id

Body: não tem

Return: ListaContatos[{id,nickName,name,status,admId}] OU error='o que ocorreu'

Status:String(se o está ausente ou presente na reunião)

AdmId: id( identificador do adm, caso o user seja o adm segunda tela)

CONTATOS - CONVERSAS

Tipo: GET -path=getContactChat-

Uso: Recupera o referencia da conversa

Params: user:id, contact:id

Body: não tem

Return: Chat:id OU error='o que ocorreu'

REUNIÃO DELETANDO

Tipo: POST -path=deleteMeeting-

Uso: caso seja adm deleta a reunião ou caso

seja participante se retira de uma reunião

Params: não tem

Body: user: id, reunião:id

Return: ‘success’ OU error='o que ocorreu'

REUNIÃO - STATUS

Tipo: GET -path=meetingStatus-

Uso: Recupera uma lista de reuniões que o usuário aceitou comparecer

Params: user:id

Body: não tem

Return: ListaMeeting[{id,theme,date,adm{id,name},occurring}] OU error='o que ocorreu'

occurring: boolean (caso a reunião esteja acontecendo naquele momento)

REUNIÃO ADICIONANDO E ATUALIZANDO

Tipo: GET -path=getMeeting-

Uso: Recupera uma reunião caso o user seja o adm da reunião

Params: user:id, meeting:id

Body: não tem

Return: meeting{id,theme,date,listContact[id,name,nickname,isThisMeeting] } OU error='o que ocorreu

isThisMeeting: boolean - (se o contato ja foi convidado ou aceitou aquela reunião - )

Tipo: POST -path=setMeeting-

Uso: Cria um nova reunião ou altera e envia os convites para os convidados ou retira eles

se meeting:id = -1 logo está sendo criada uma reunião

Params: não tem

Body: user: id, meeting:id, date: string data e hora, listContact[id]

Return: ‘success’ OU error='o que ocorreu'

REUNIÃO REMOVE O CONTATO

Tipo:POST -path=removeContactMeeting-

Uso: Retira um contato da reunião (somente se user:id == adm:id)

Params: não tem

Body: user:id, meeting:id, contact:id

Return: string ‘success’ OU error='o que ocorreu'

REUNIÃO ADICIONA O CONTATO

Tipo: POST -path=addContactMeeting-

Uso: manda um convite aos contatos

em um reunião ocorrendo (somente se user:id == adm:id)

Params: não tem

Body: user:id, meeting:id, contact:id

Return: string ‘success’ OU error='o que ocorreu'

REUNIÃO ENCERRA A REUNIÃO

Tipo: POST -path=finishingMeeting-

Uso: finaliza a reunião encerrando o envio de mensagens

Params: não tem

Body: user:id, meeting:id

Return: string ‘success’ OU error='o que ocorreu'

Reunião - CONVERSAS

Tipo: GET -path=getReuniãoChat-

Uso: Recupera o referencia da conversa

Params: user:id, meeting:id

Body: não tem

Return: Chat:id OU error='o que ocorreu'

CONVERSAS - ENVIA MENSAGEM E RECEBIMENTO DE MESSAGENS

a conversa será feita pelo ratchet framework (php)

URL: ‘ws://localdobackend/chat’

ENVIO será enviado um json para o ratchet

data:{user:id,user:name,chat:id,message:string}

a messagem do usuario deve passar primeiro pelo backend antes

de aparecer no frontend.

RETORNO será recebido um json similar ao enviado

data:{sentMessageId:id,sentMessageName:string, date: data e hora do envio, message:string}