Websockets - Cliente

1. Conectar-se ao servidor

- **Nome do evento:** “connect”
- **Descrição:** abre uma conexão websocket com o servidor

2. Diffie-Hellman com servidor

- **Nome do evento:** “diffie-hellman”
- **Descrição:** diffie-hellman para definir chave compartilhada entre servidor e usuário
- **Parâmetros:** p, g, userPublicKey
- **Recebe:** Json
  - {sucess: boolean, serverPublicKey: string}

! Quando estiver logado

- **Nome do evento:** “online-loged”
- **Descrição:** evento para que o servidor adicione usuário e seu respectivo id de socket à lista de usuários online e logados.
- **Parâmetros:** user_name
- **Recebe:** JSON criptografado com chave compartilhada entre servidor-usuário
  - { **success**: true, message: string,

    offlineMessages: json com mensagens recebidas enquanto estava offline,

    friendRequests: json com solicitações recebidas enquanto estava offline
    }

! Fazer login

- **Nome do evento:** “login”
- **Descrição:** evento para validar login do usuário; dados de login são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados = e-mail, senha hasheada
- **Recebe:** Json
  - {sucess: boolean, message: string}
  - Se (sucess: true){ deve chamar o evento “online-loged”}

! Registrar-se

- **Nome do evento:** “register”
- **Descrição:** evento para registrar usuário; dados de registro são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados = nome, e-mail, senha hasheada, nome de usuário, foto de perfil
- **Recebe:** Json
  - {sucess: boolean, message: string}

! Listar usuários que não são amigos

- **Nome do evento:** “list-users”
- **Descrição:** evento listar usuários que não são amigos, não enviaram solicitação ou não têm solicitação pendente do usuário conectado.
- **Parâmetros:** user_name
- **Recebe:** Json criptografado usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.

- {{user_name1: string, name1: string}, {user_name2: string, name2: string},{user_name3: string, name3: string}}

! Solicitar amizade

- **Nome do evento:** “friend-request”
- **Descrição:** usuário solicita amizade com outro usuário, enviando também parâmetros diffie-hellman para que seja estabelecida uma chave compartilhada fixa entre ele e o outro usuário. Dados de solicitação são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Problema:** se um usuário apaga o app e perde sua chave privada de amizade, não pode mais se comunicar com os amigos.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados: nome de usuário, nome do usuário a receber a solicitação, p, g, publicKeyDestinatário
- **Recebe:** Json
  - {sucess: boolean, message: string}

! Aceitar solicitação de amizade

- **Nome do evento:** “accept-friend”
- **Descrição:** evento para que o usuário aceite o pedido de amizade, atualizando o campo “friendship” para `true` e enviando sua chave pública diffie-hellman para seu amigo. Esses dados são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados: nome de usuário, nome do amigo, sua chave pública
- **Recebe:** Json
  - {sucess: boolean, message: string}

¡ Rejeitar solicitação de amizade

- **Nome do evento:** “reject-friend”
- **Descrição:** evento para que o usuário rejeite o pedido de amizade, fazendo com que a solicitação do outro usuário seja excluída do banco de dados. Esses dados são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados: nome de usuário, nome do amigo
- **Recebe:** Json
  - {sucess: boolean, message: string}

! Mandar mensagem para amigo

- **Nome do evento:** “send-message”
- **Descrição:** evento para o usuário mandar mensagem para um amigo. Dados são criptografados usando o algoritmo Blowfish e a chave compartilhada gerada entre servidor e usuário.
- **Parâmetros:** dadosCriptografados
  - dadosCriptografados: nome de usuário remetente, nome do amigo destinatário, timestamp, mensagemCriptografada
  - mensagemCriptograda = conteúdo da mensagem é criptografado com o algoritmo Blowfish usando a chave compartilhada entre amigos
- **Recebe:** Json
  - {sucess: boolean, message: string}

¡ Recebe solicitações de amizade

- **Nome do evento:** “receive-friend-request”
- **Descrição:** evento disparado pelo servidor para que o usuário receba solicitação de amizade. Necessário descriptografar os dados recebidos com a chave compartilhado entre servidor e usuário.
- **Recebe:** dadosCriptografados

    - dadosCriptografados: nome de usuário remetente, nome do usuário destinatário, p, g, publicKeyDestinatário
    - formato dos json descriptografado: { friend1: string, p: string, g: string, publicRemetentKey: string}
- **Envia:** Json
    - {sucess: boolean, message: string}

¡ Recebe mensagens

- **Nome do evento:** “receive-message”
- **Descrição:** evento disparado pelo servidor para que o usuário receba mensagens. Necessário descriptografar os dados recebidos com a chave compartilhado entre servidor e usuário.
- **Recebe:** dadosCriptografados
    - dadosCriptografados: nome de usuário remetente, nome do amigo destinatário, timestamp, mensagemCriptografada com chave de amizade.
    - formato dos json descriptografado: { friend1: string, datetime: timestamp, content: string }
- **Envia:** Json
    - {sucess: boolean, message: string}

! Desconectar

- **Nome do evento:** “disconnect”
- **Descrição:** fecha uma conexão websocket com o servidor