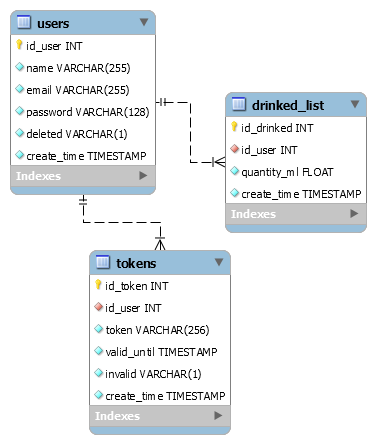
# Banco de dados



# Informações importantes

Abaixo de cada título haverá a definição do endpoint, sendo especificado como

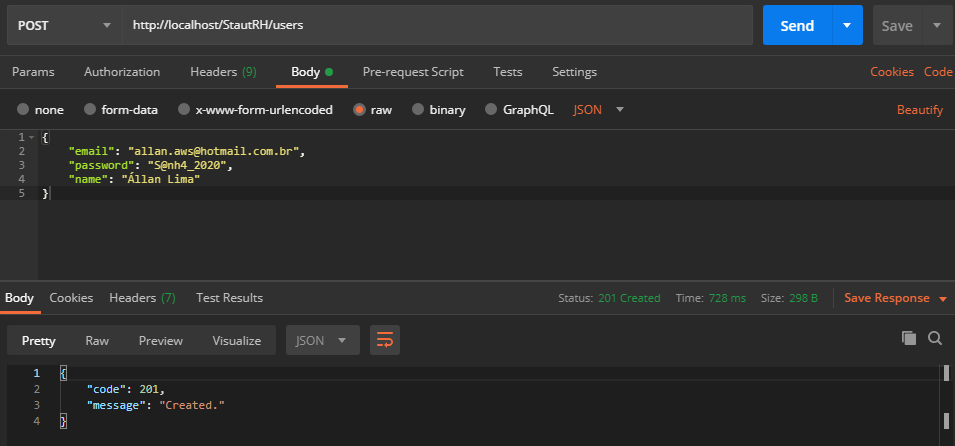
/SERVICO -> MÉTODO HTTP

Por exemplo:

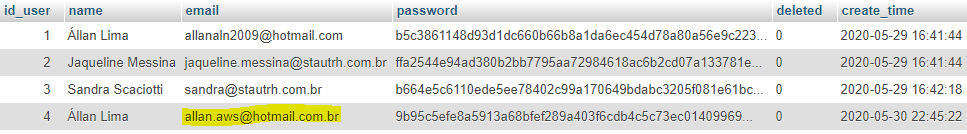
/users -> post

# Criação de usuário

/users -> post

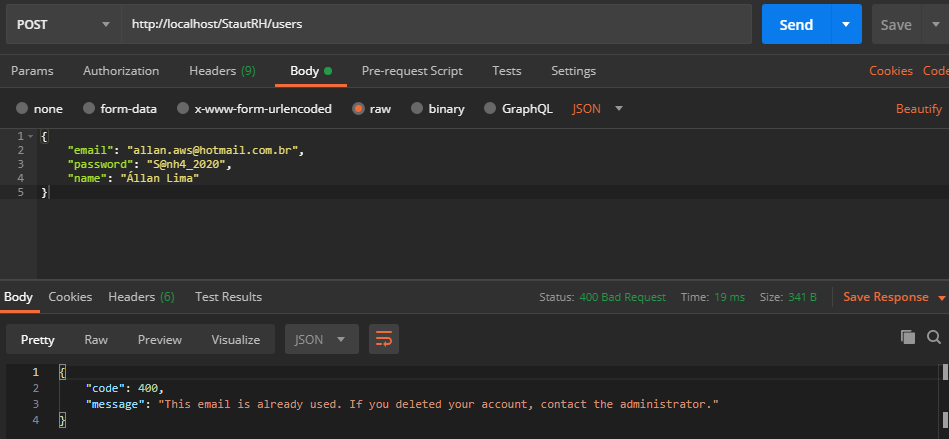


Dado inserido em tabela *users* do banco de dados.

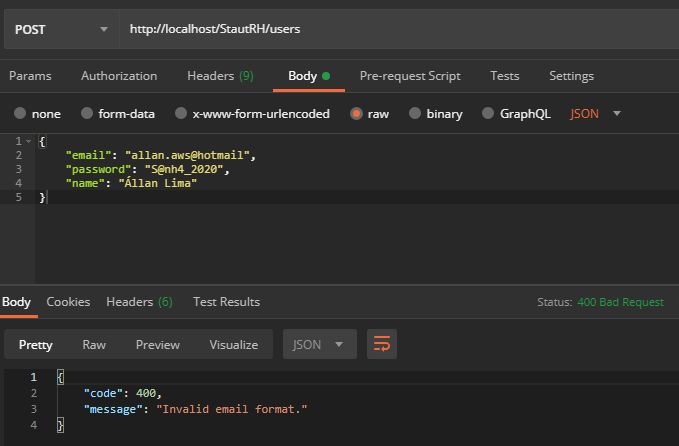


## Tratativas

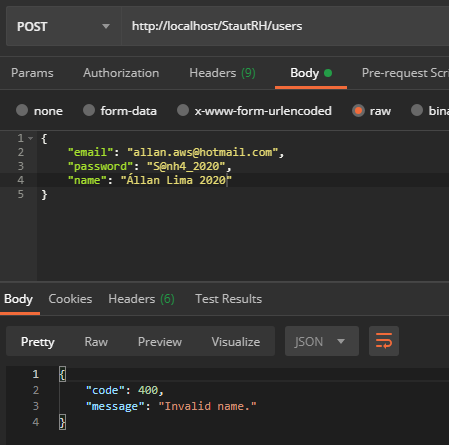
### Reenvio



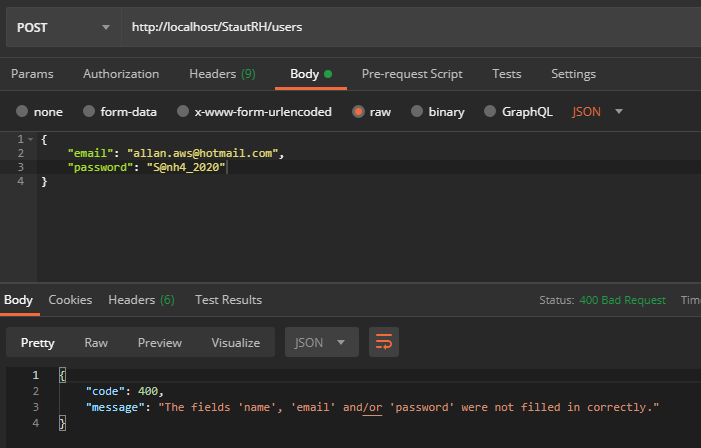
### Email inválido



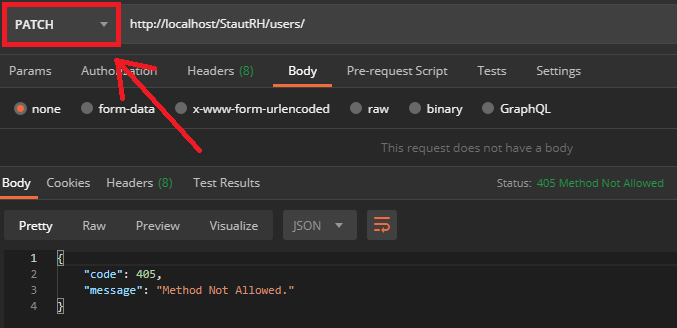
### Nome inválido



### JSON inválido

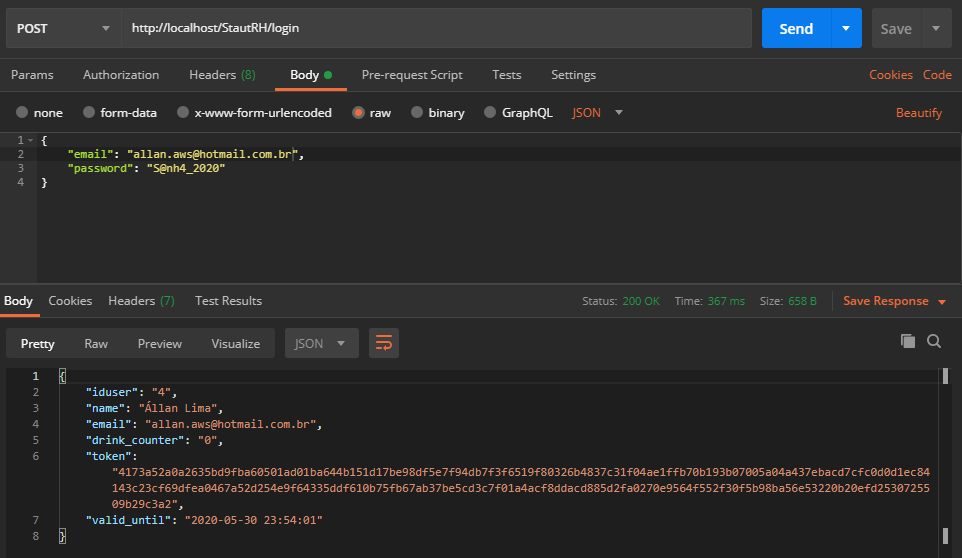


### Método não permitido



# Login

/login -> post



Registro de token criado na tabela *tokens* do banco de dados. Token válido por 1 hora.

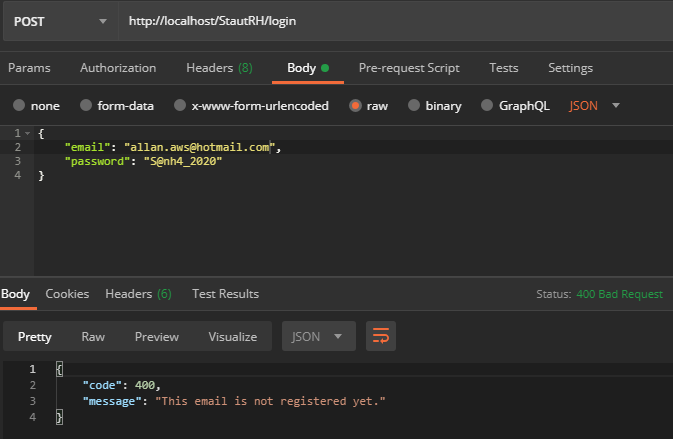


A API não permite o uso de tokens antigos ou de outros usuários.

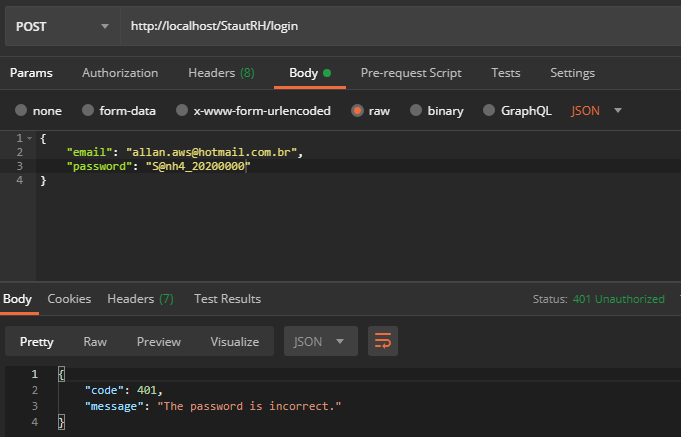
## Tratativas

### Email inexistente

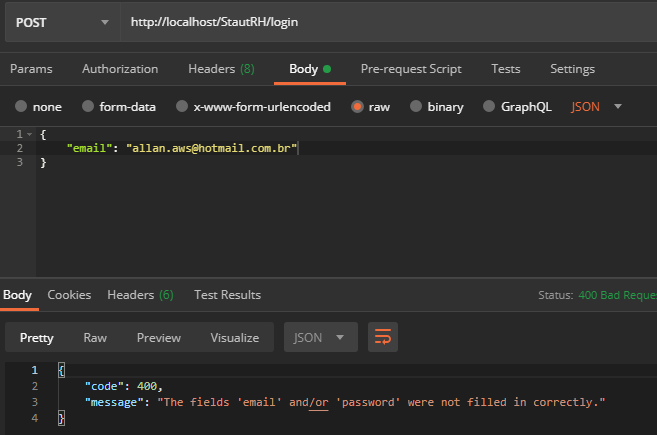
([allan.aws@hotmail.com](mailto:allan.aws@hotmail.com) não existe no BD, apenas [allan.aws@hotmail.com**.br**](mailto:allan.aws@hotmail.com.br))



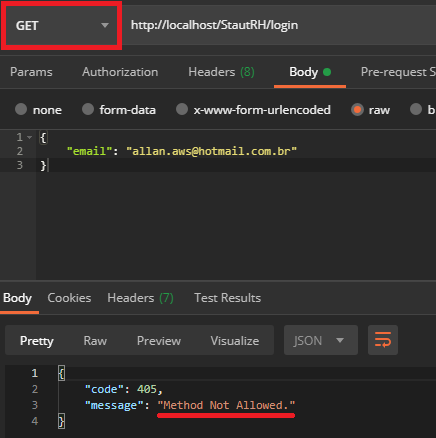
### Senha informada não confere com a de cadastro



### Campo não informado corretamente



### Método não permitido



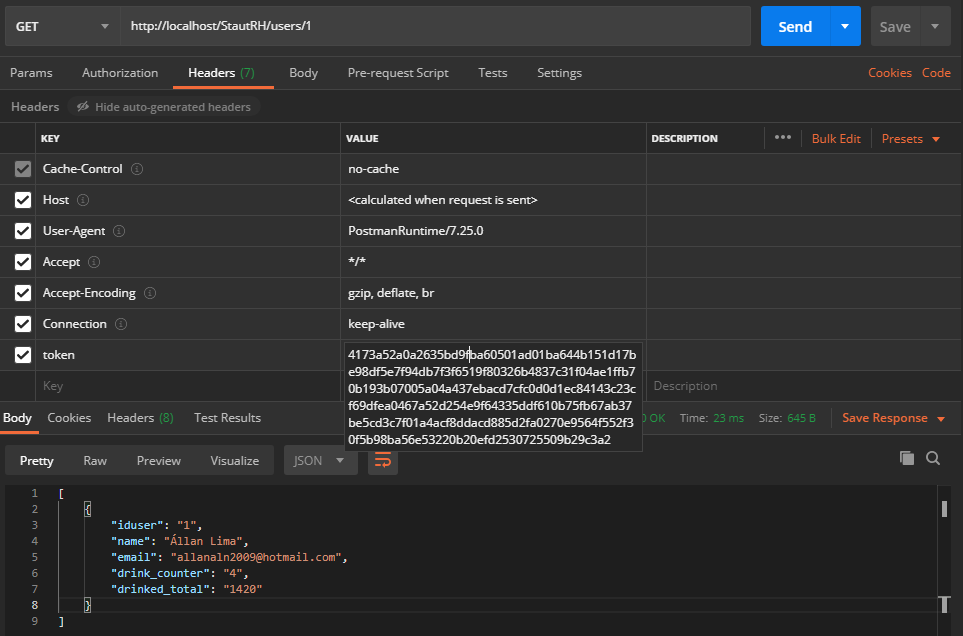
### Ao realizar um novo login, tokens antigos ativos são invalidados.

### 

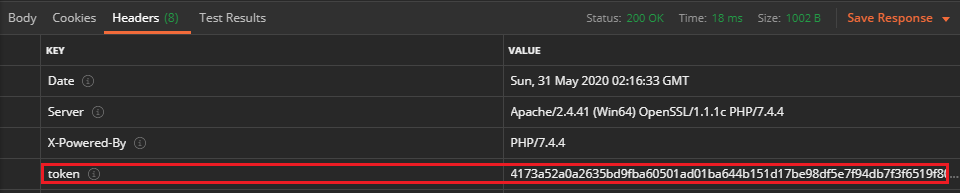
### 

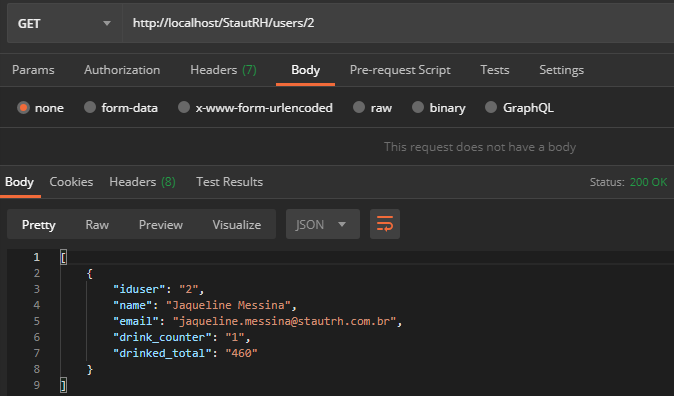
# Obtenção de usuário

/users/:iduser -> get

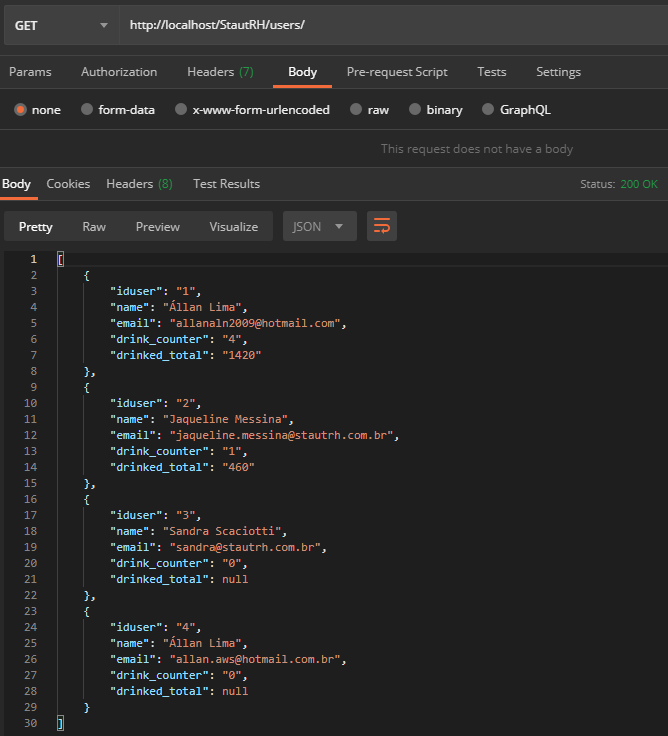


ID de usuário informado na URI e token passado através do header da requisição como ‘token’.

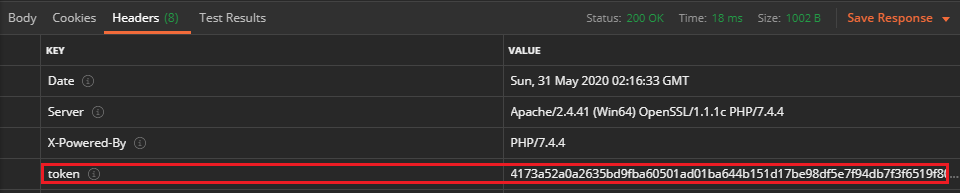




/users/ -> get



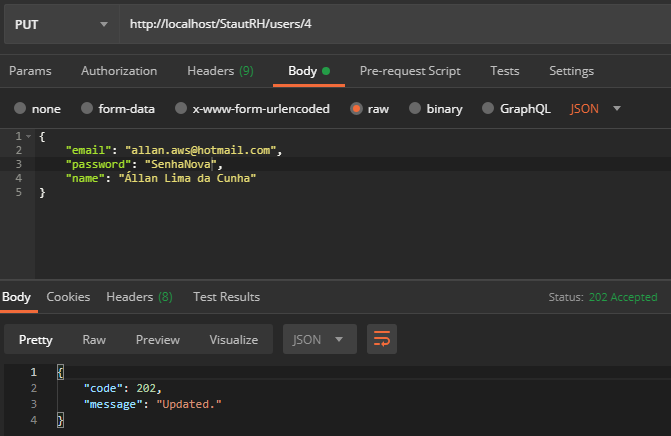
ID de usuário não especificado retorna todos os registros.



Token sempre devolvido em header da resposta.

# Alteração de dados do usuário

/users/:iduser -> put

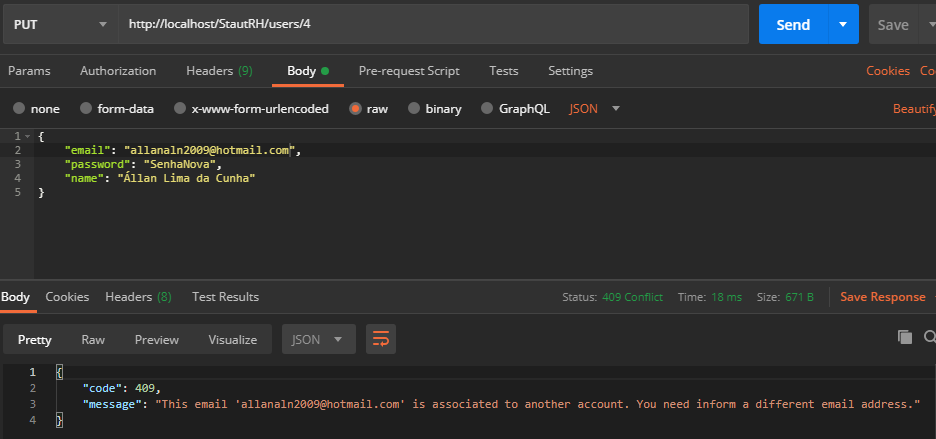




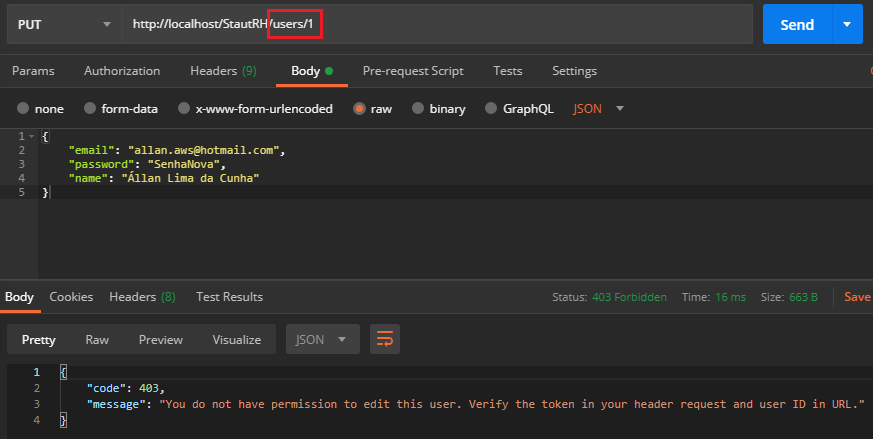
Dados atualizados no banco.

## Tratativas

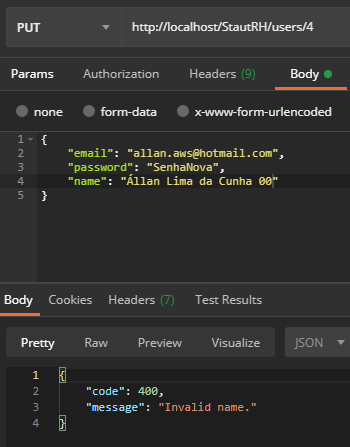
### Uso de email já utilizado por outra pessoa



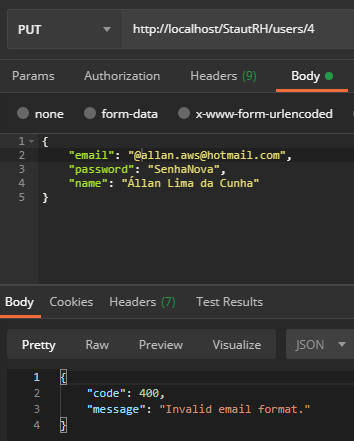
### Tentativa de alteração de outro ID de usuário



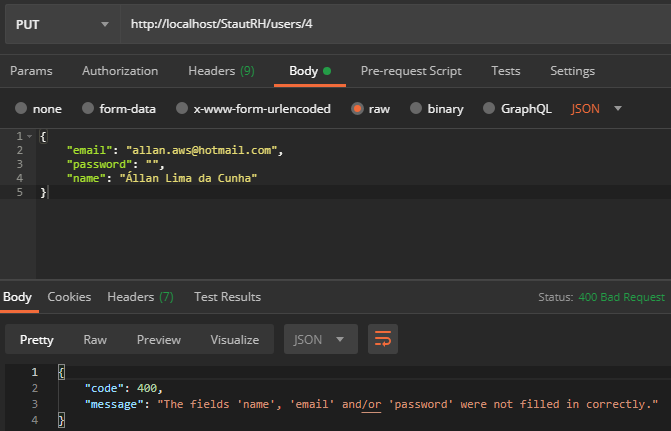
### Nome inválido



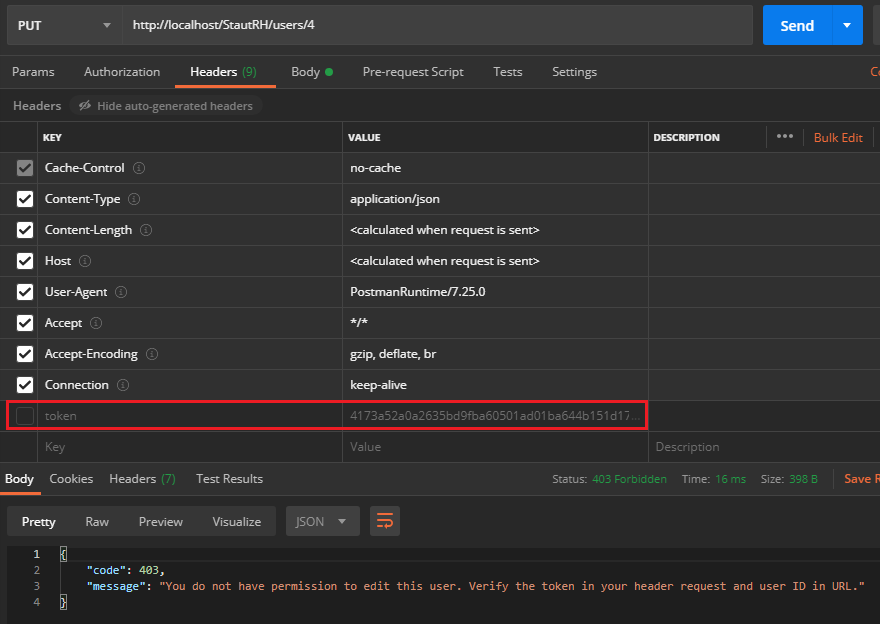
### Email inválido



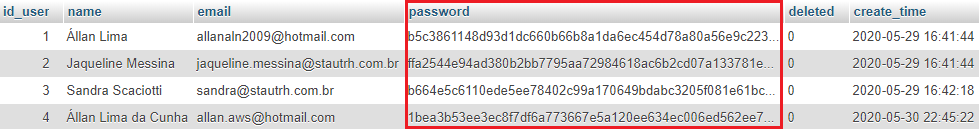
### Campos não preenchidos



### Token não enviado no cabeçalho



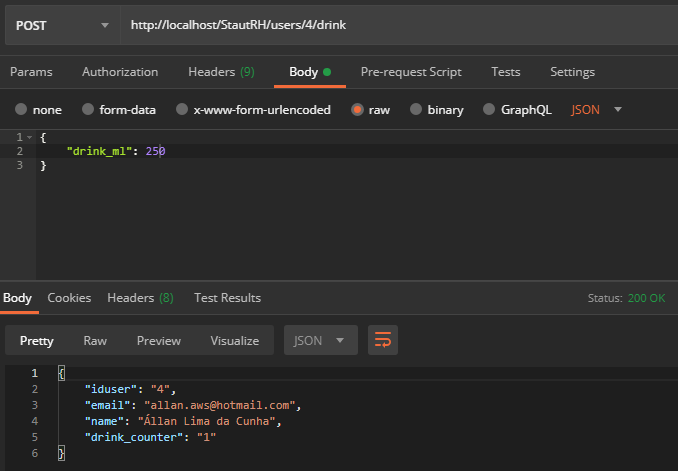
### Senha salva com criptografia gerando hashes exclusivos para cada registro



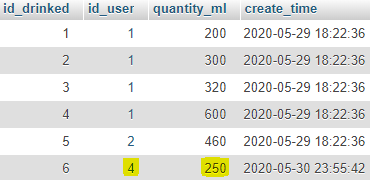
Um usuário nunca terá seu hash idêntico ao de outro usuário, mesmo que suas senhas sejam iguais. Isso aumenta a segurança e impede o uso de dicionário de senhas caso a base de dados seja comprometida.

# Inclusão de itens bebidos

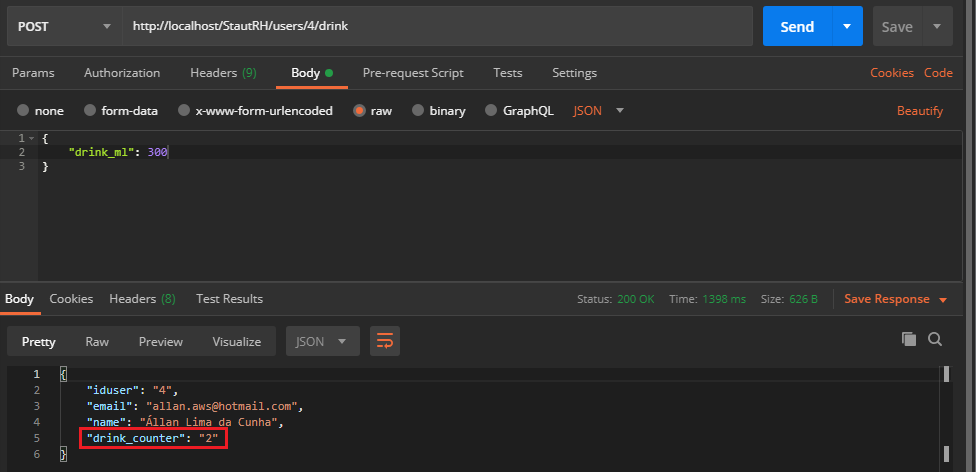
/users/:iduser/drink -> post

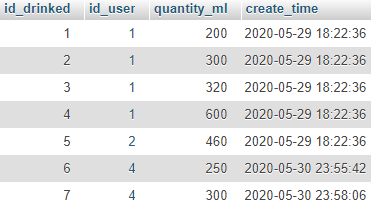


Uma linha foi adicionada à tabela *drinked\_list*.



Novo teste. Contador é atualizado a cada requisição.

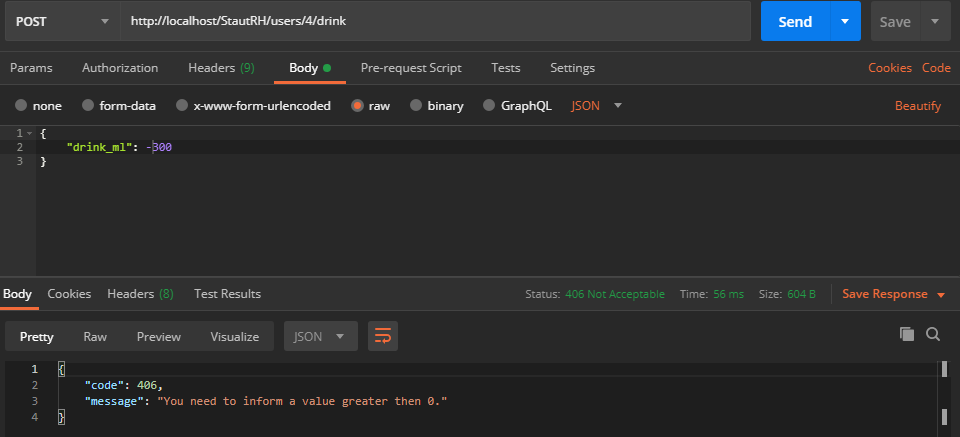




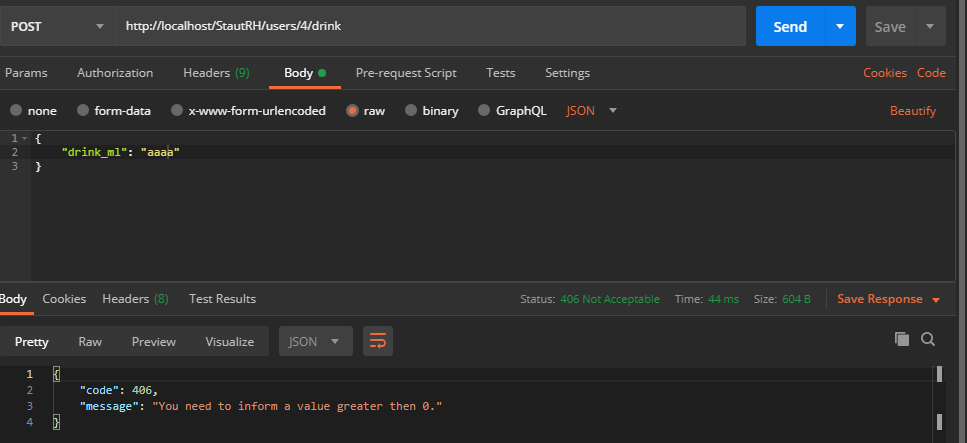
## Tratativas

Assim como nos casos anteriores, não é possível inserir um item em ‘drink’ sem estar devidamente informando o token de seu usuário. Não é possível inserir um registro para outro usuário diferente do proprietário do token.

### Valor negativo

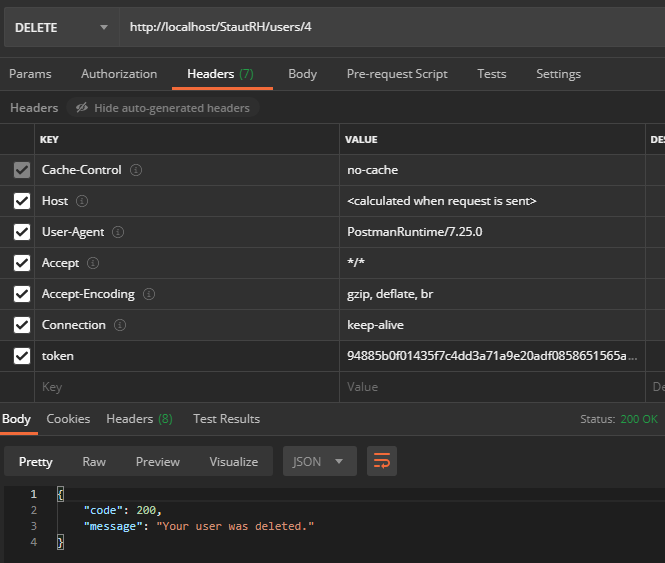


### Valor não numérico



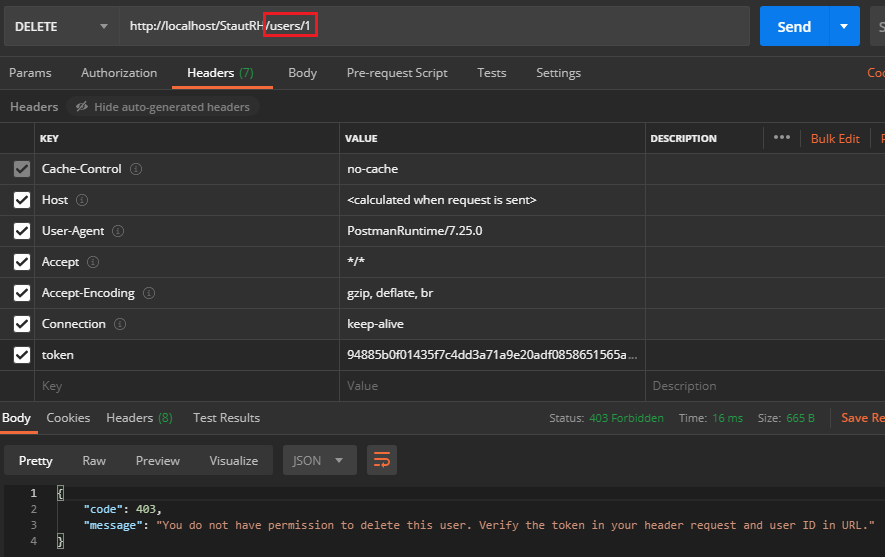
# Exclusão de usuário

/users/:iduser -> delete

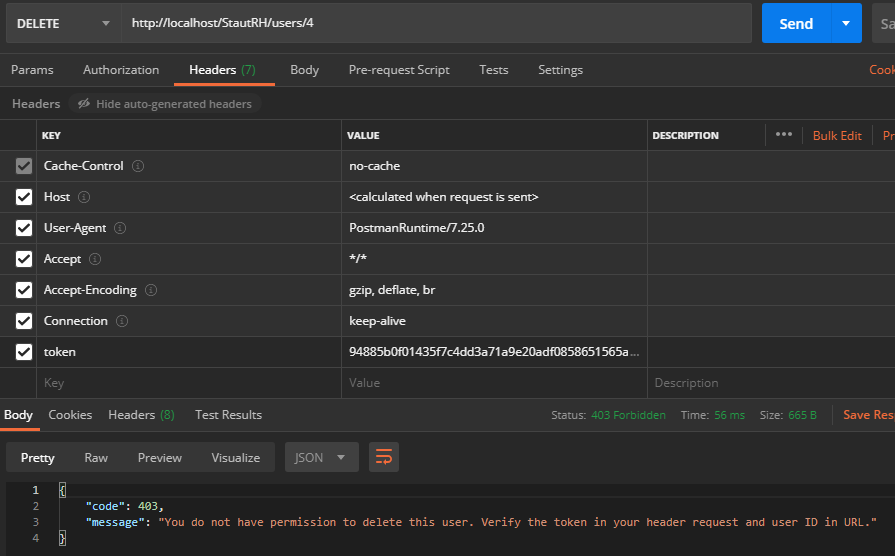


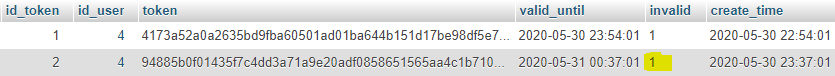
## Tratativas

### Não é possível excluir outro usuário diferente do proprietário do token



### Token do usuário passa a ser inativo





# Opcionais

Paginação na lista de usuários

/users&**per\_page**=:valor&**page**=:valor -> get

Paginação simples aplicada por deslocamento de IDs. Apenas funciona com offset SQL, não aplicando nenhum filtro na busca e considerando todos os registros.

## Parametros

*per\_page*: quantidade de itens recuperados por página

* obrigatório para paginação.

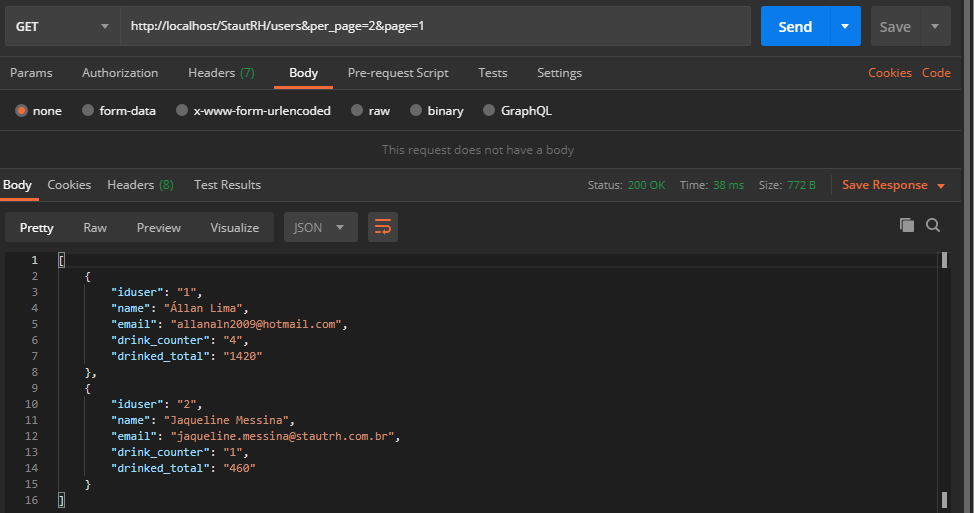
*page*: página a ser recuperada

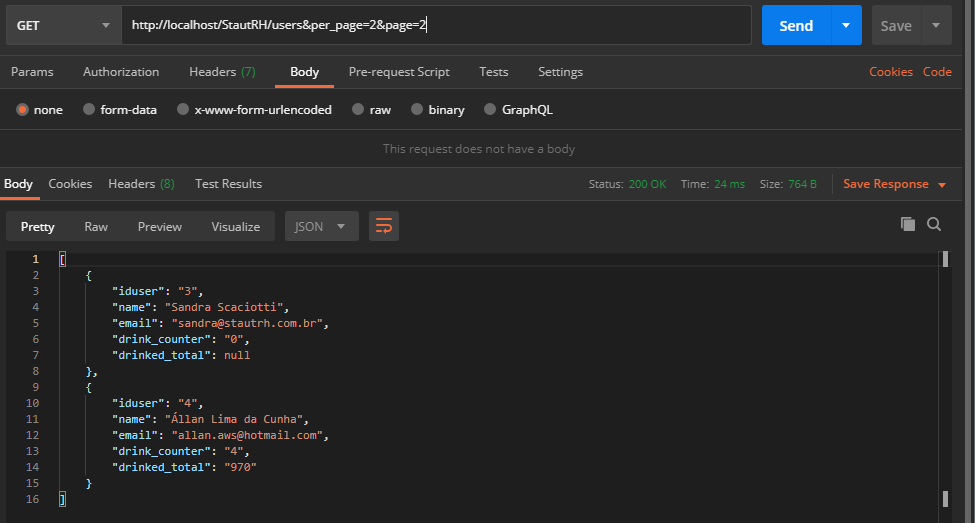
* é ignorado caso per\_*page* não seja informado.
* é opcional; não informando ou preenchendo incorretamente, valor atribuído será 1.

Ambos os parâmetros são validados.

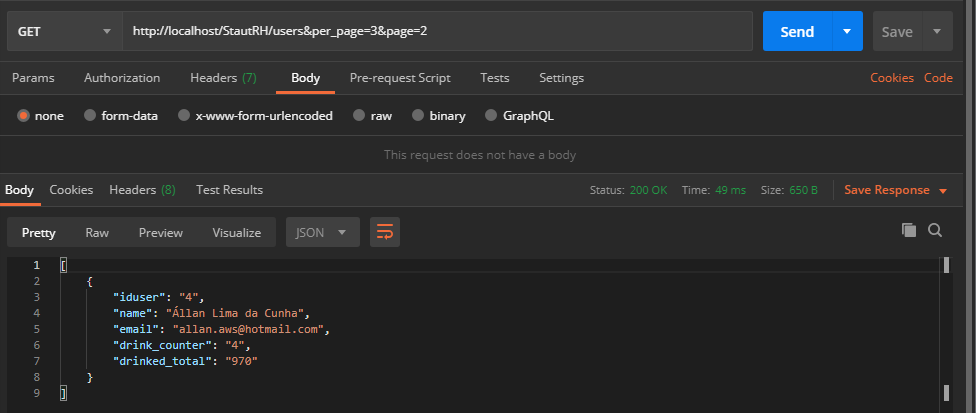
## Exemplos

2 itens por página, página 1 e 2:

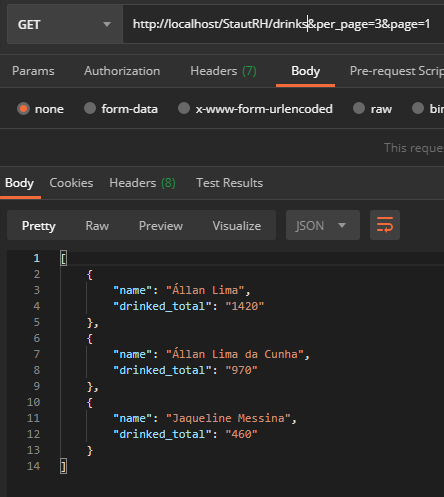




3 itens por página, página 2:

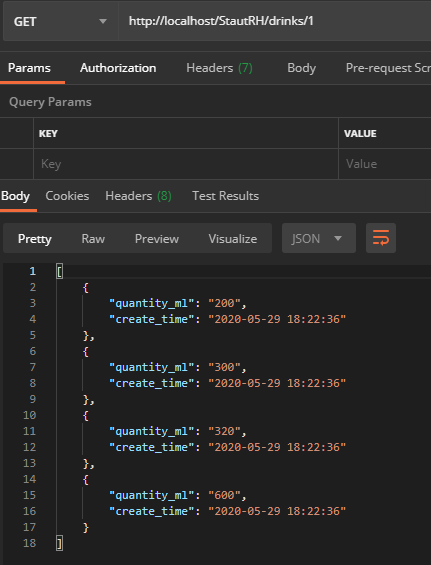


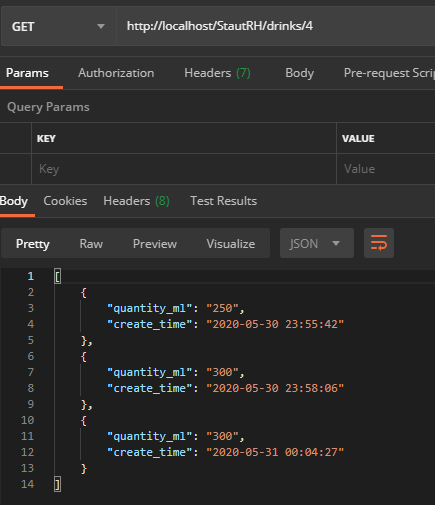
Também funciona no serviço de consulta de ranking.

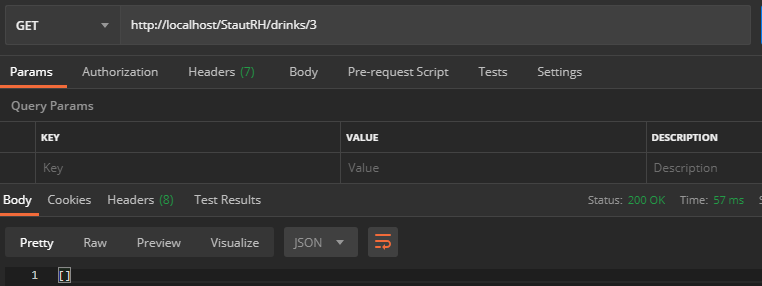


Criar um serviço que liste o histórico de registros de um usuário (retornando a data e a quantidade em mL de cada registro)

/drinks/:iduser -> get



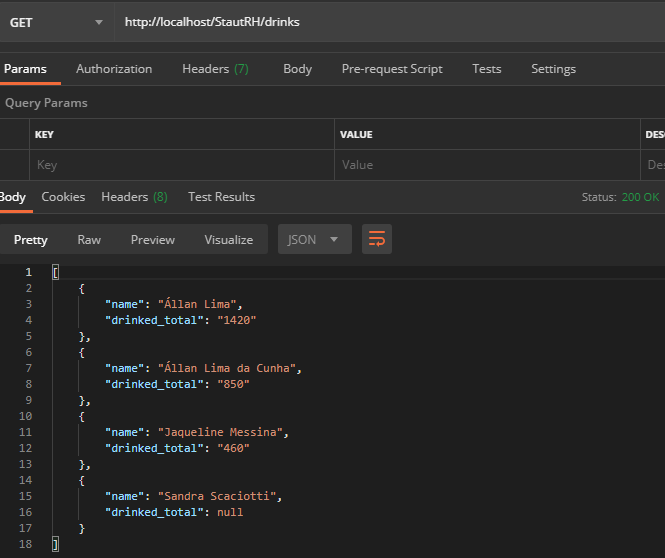




Para esse serviço não é necessário informar nenhum token no header. O serviço está aberto e não necessita de autenticação. Qualquer ID de usuário pode ser pesquisado, até mesmo os excluídos.

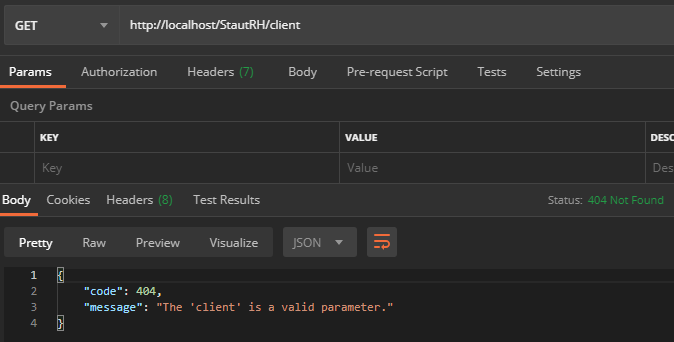
Criar um serviço que liste o ranking do usuário que mais bebeu água hoje (considerando os ml e não a quantidade de vezes), retornando o nome e a quantidade em mililitros (mL)

/drinks -> get



Assim como no caso anterior, o serviço pode ser acessado sem um token. Para visualizar o ranking, basta não informar nenhum ID de usuário na URI.

Ao tentar acessar um serviço inexistente, a API aponta o erro com o HTTP code 404.



Todo o ecossistema da API possui tratativa para métodos não permitidos.

