

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
ENGENHARIA DE COMPUTAÇÃO
REDES DE COMPUTADORES I

LUCAS JOSE DE FREITAS

TRABALHO PRÁTICO I

Implementação de um API RESTful para um sistema de *chat*

Belo Horizonte/MG

2024

Documentação da API

A API foi implementada em node.js permitindo que os usuários registrem-se, entrem em salas de chat, enviem mensagens para salas de chat e enviem mensagens diretas para outros usuários. Dessa maneira foram implementados os seguintes recursos dentro dos blocos:

- Usuários: Gerencia o registro, autenticação e informações dos usuários.
- Salas de Chat: Permite a criação, entrada, saída e remoção de salas de chat.
- Mensagens em Salas de Chat: Envia e recebe mensagens em salas de chat.
- Mensagens Diretas: Envia e recebe mensagens diretas entre usuários.

Logo, faz-se necessário que saiba os seguintes dados:

Base URL: `http://localhost:3000`

Endpoints - Usuários

Registro um Novo Usuário

POST /users - Cria um novo usuário.

Parâmetros:

- `username` (string, obrigatório): Nome de usuário único.

Respostas:

- 201 - Criado: O usuário foi criado com sucesso.
- 400 - Requisição Inválida: Se o nome de usuário não for fornecido.
- 409 - Conflito: Se o nome de usuário já estiver em uso.

Autenticar um Usuário

POST /users/login - Autentica um usuário.

Parâmetros:

- `username` (string, obrigatório): Nome de usuário a ser autenticado.

Respostas:

- 200 - OK: Autenticação bem-sucedida, retorna o ID do usuário.
- 404 - Não Encontrado: Se o usuário não for encontrado.

Obter Informações de um Usuário

GET /users/:userId - Obtém informações de um usuário específico.

Parâmetros:

- `userId` (string, obrigatório): ID do usuário.

Respostas:

- 200 - OK: Retorna informações do usuário.
- 404 - Não Encontrado: Se o usuário não for encontrado.

EndPoints - Salas de Chat

Criar uma Nova Sala de Chat

POST /rooms - Cria uma nova sala de chat.

Parâmetros:

- `name` (string, obrigatório): Nome da sala de chat.

Respostas:

- 201 - Criado: A sala de chat foi criada com sucesso.
- 400 - Requisição Inválida: Se o nome da sala não for fornecido.

Remover uma Sala de Chat

DELETE /rooms/:roomId - Remove uma sala de chat existente.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat a ser removida.

Respostas:

- 200 - OK: A sala de chat foi removida com sucesso.
- 404 - Não Encontrado: Se a sala de chat não for encontrada.

Entrar em uma Sala de Chat

POST /rooms/:roomId/enter - Entra em uma sala de chat existente.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat.
- `userId` (string, obrigatório): ID do usuário que está entrando na sala.

Respostas:

- 200 - OK: O usuário entrou na sala de chat com sucesso.
- 404 - Não Encontrado: Se a sala de chat ou o usuário não forem encontrados.

Sair de uma Sala de Chat

POST /rooms/:roomId/leave - Sai de uma sala de chat existente.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat.
- `userId` (string, obrigatório): ID do usuário que está saindo da sala.

Respostas:

- 200 - OK: O usuário saiu da sala de chat com sucesso.
- 404 - Não Encontrado: Se a sala de chat não for encontrada.

Remover um Usuário de uma Sala Específica

DELETE /rooms/:roomId/users/:userId - Remove um usuário específico de uma sala de chat.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat.
- `userId` (string, obrigatório): ID do usuário a ser removido da sala.

Respostas:

- 200 - OK: O usuário foi removido da sala de chat com sucesso.
- 404 - Não Encontrado: Se a sala de chat não for encontrada.

Listar Todas as Salas Ativas

GET /rooms/active - Lista todas as salas de chat ativas.

Respostas:

- 200 - OK: Retorna um array com todas as salas de chat ativas.
- 404 - Não Encontrado: Se não houver salas de chat ativas.

EndPoints - Mensagens em Salas de Chat

Enviar uma Mensagem para uma Sala de Chat

POST /rooms/:roomId/messages - Envia uma mensagem para uma sala de chat específica.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat.
- `userId` (string, obrigatório): ID do usuário que está enviando a mensagem.
- `message` (string, obrigatório): Conteúdo da mensagem.

Respostas:

- 204 - Sem Conteúdo: A mensagem foi enviada com sucesso.

Receber Mensagens de uma Sala de Chat

GET /rooms/:roomId/messages - Obtém todas as mensagens de uma sala de chat específica.

Parâmetros:

- `roomId` (string, obrigatório): ID da sala de chat.

Respostas:

- 200 - OK: Retorna um array com todas as mensagens da sala de chat.

Mensagens Diretas

Enviar uma Mensagem Direta para Outro Usuário

POST /messages/direct - Envia uma mensagem direta para outro usuário.

Parâmetros:

- `recipientId` (string, obrigatório): ID do destinatário da mensagem.
- `senderId` (string, obrigatório): ID do remetente da mensagem.
- `message` (string, obrigatório): Conteúdo da mensagem direta.

Respostas:

- 200 - OK: A mensagem direta foi enviada com sucesso.
- 400 - Requisição Inválida: Se o conteúdo da mensagem não for fornecido.
- 404 - Não Encontrado: Se o destinatário ou remetente não forem encontrados.

Obter Mensagens Diretas de um Usuário

GET /messages/direct/:userId - Obtém todas as mensagens diretas de um usuário específico.

Parâmetros:

- `userId` (string, obrigatório): ID do usuário.

Respostas:

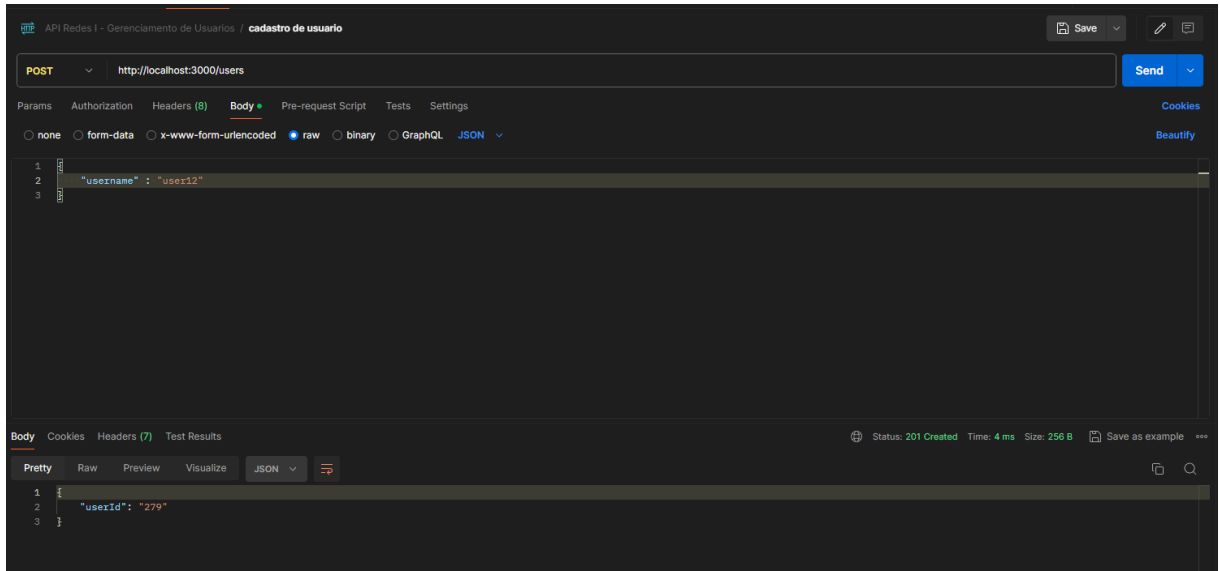
- 200 - OK: Retorna um array com todas as mensagens diretas do usuário.
- 404 - Não Encontrado: Se o usuário não for encontrado.

Testes e Resultados

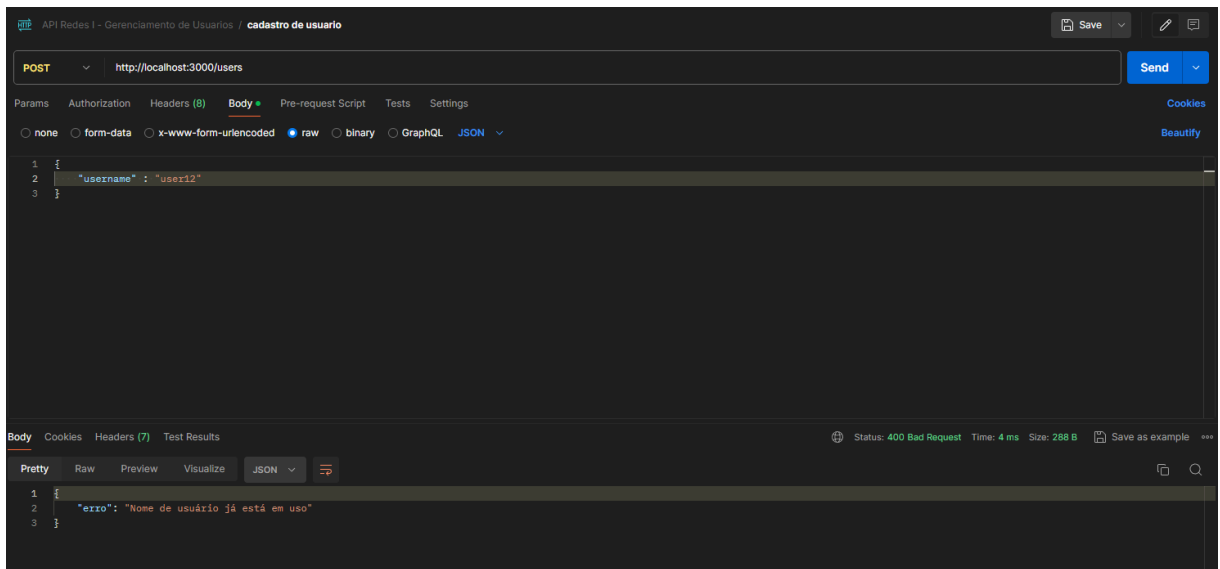
Para os testes foi usada a aplicação **Postman**, fazendo a simulação que requisições para a validação da API.

Testes nas requisições de Gerenciamento de Usuário

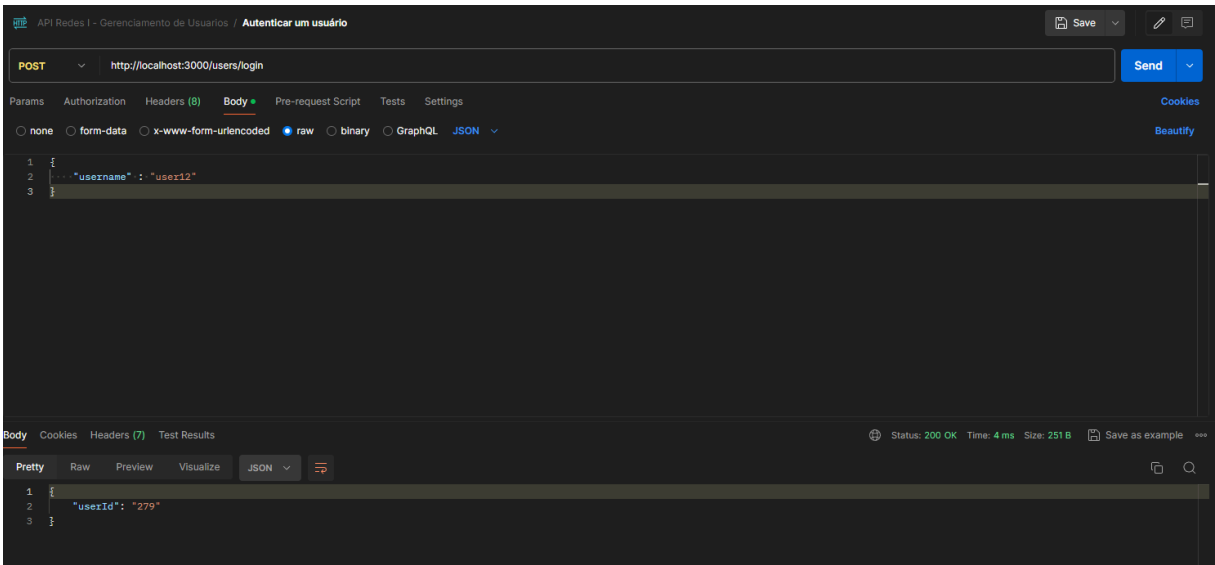
Registro de um novo usuário - criação de login



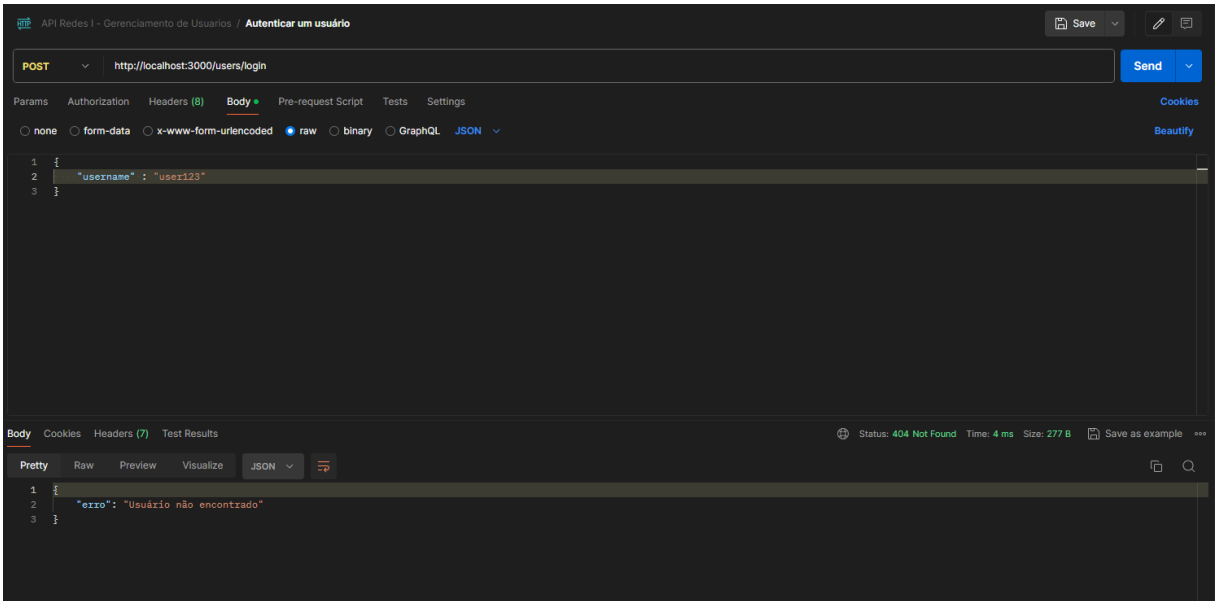
Tentativa de criação de usuário com mesmo username



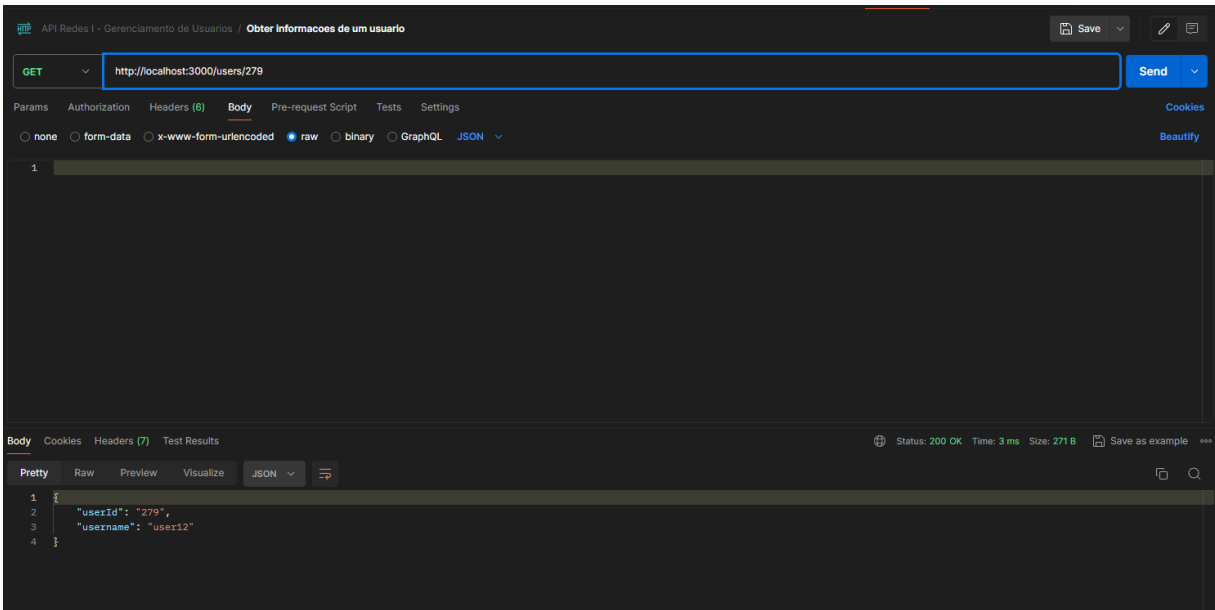
Autenticação de um usuário - autenticação de usuário já cadastrado



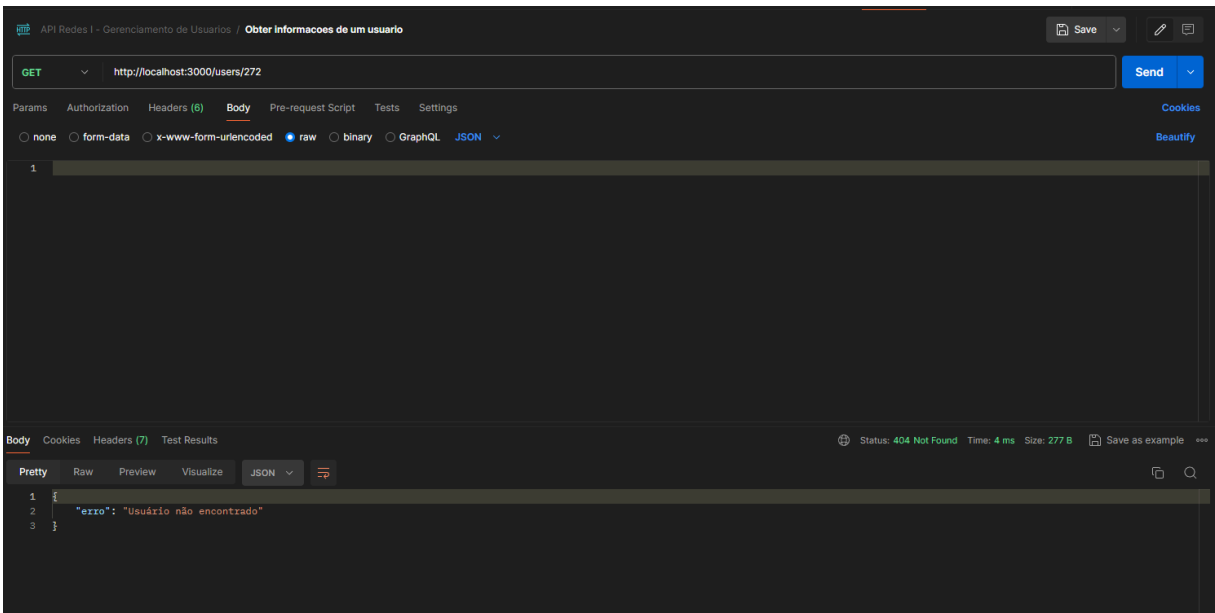
Autenticação de um usuário não cadastrado



Consulta de um usuário - consulta de um usuário já cadastrado

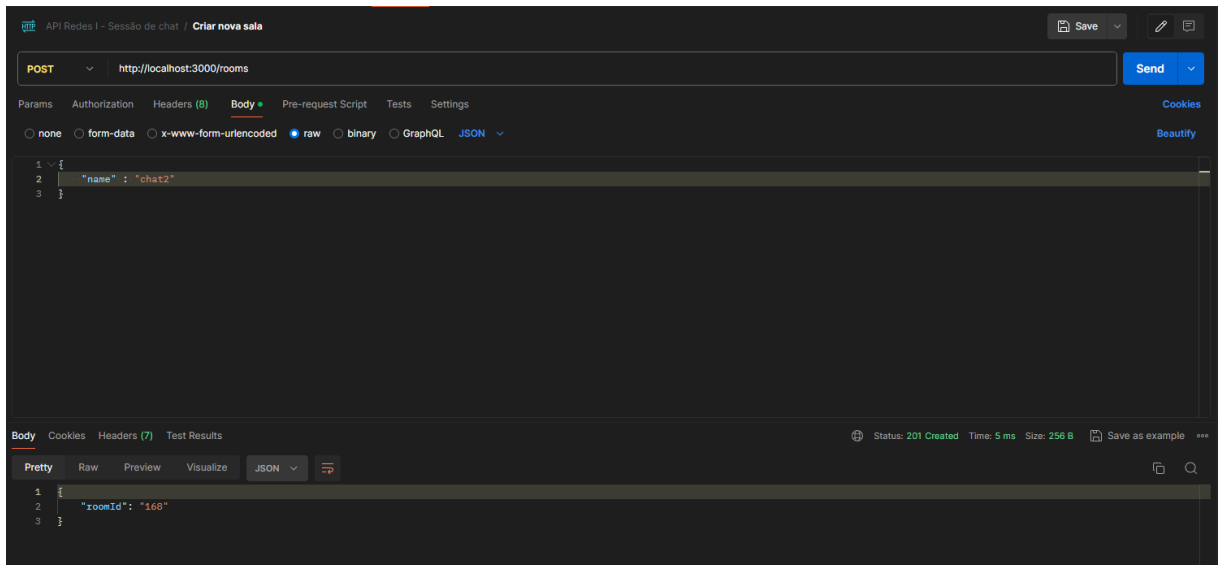


Consulta de um usuário não cadastrado

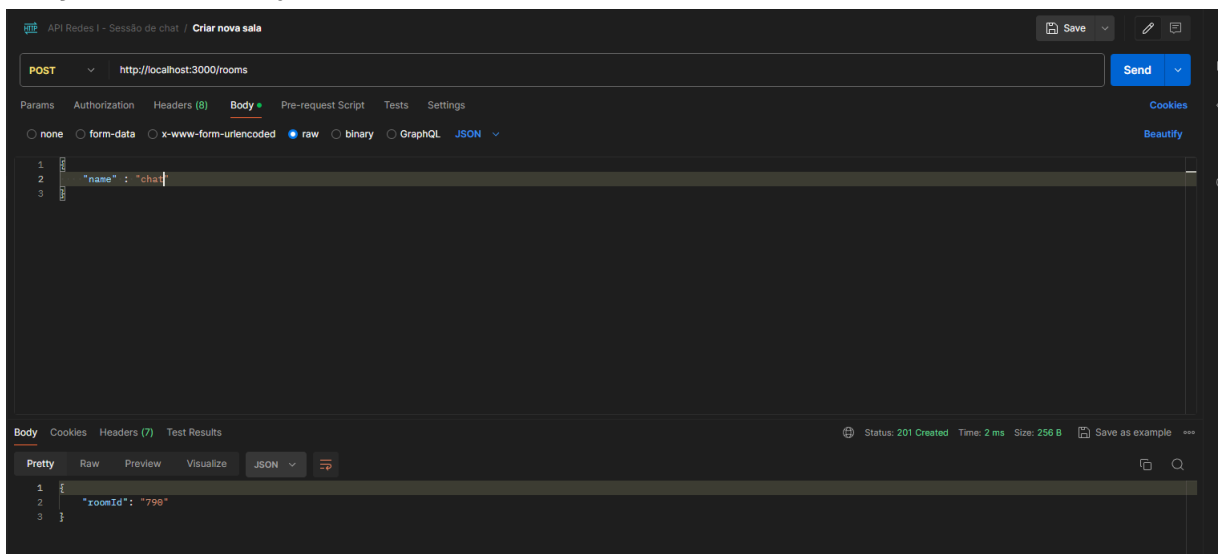


Testes nas requisições de Gerenciamento de Sessões de Chat

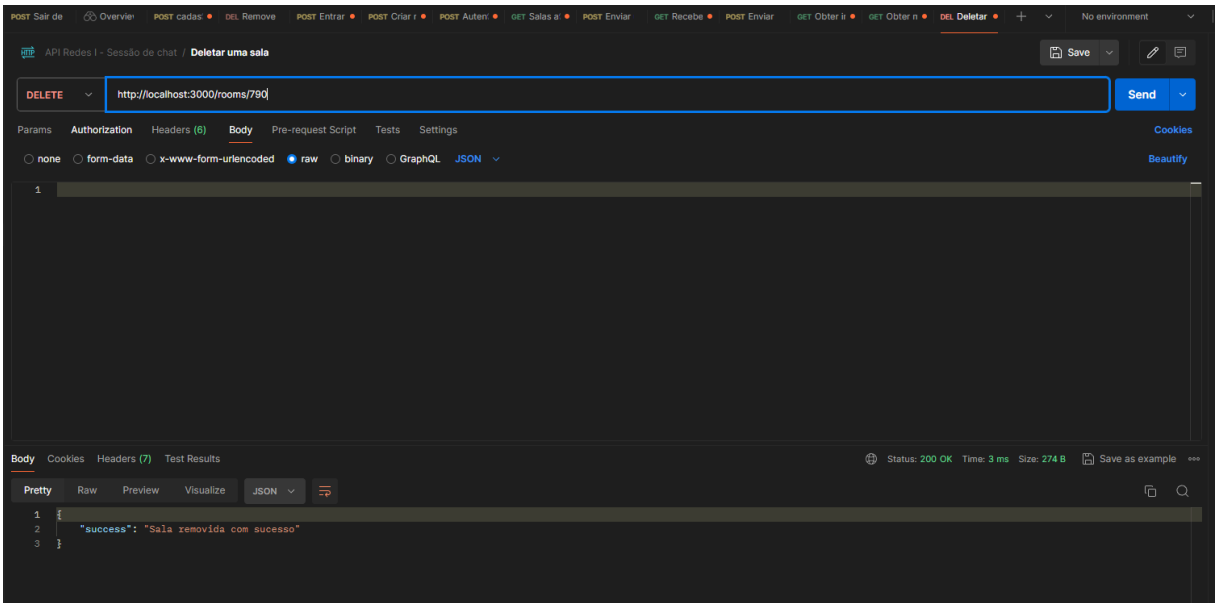
Criação de uma nova sala



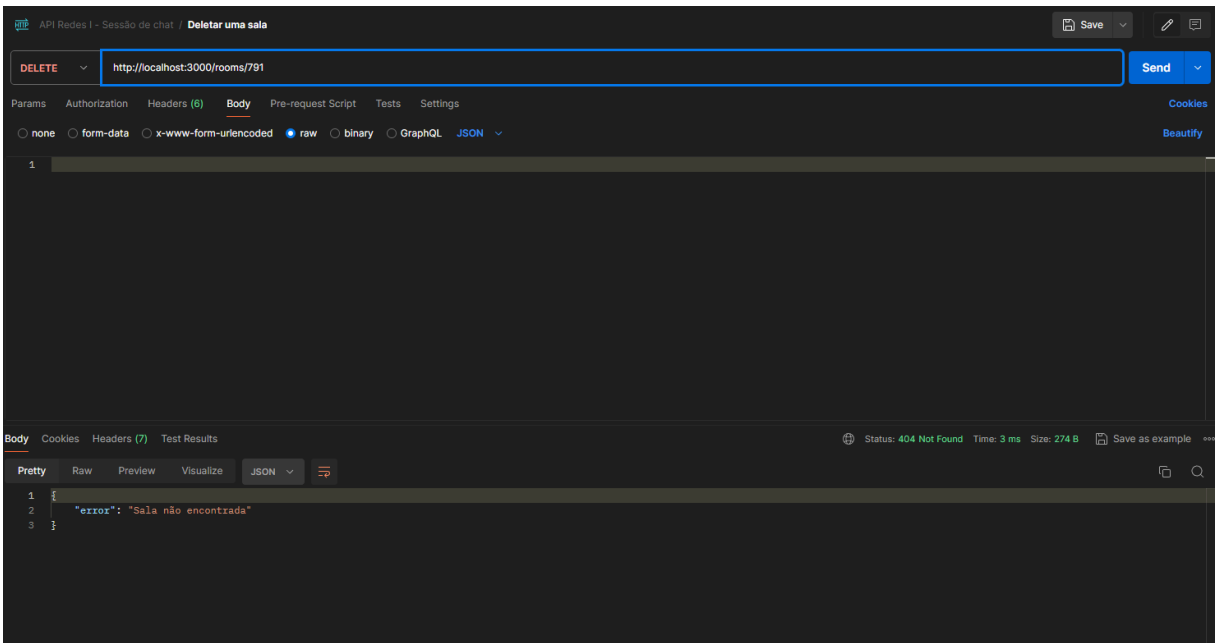
Criação de uma sala ja cadastrada



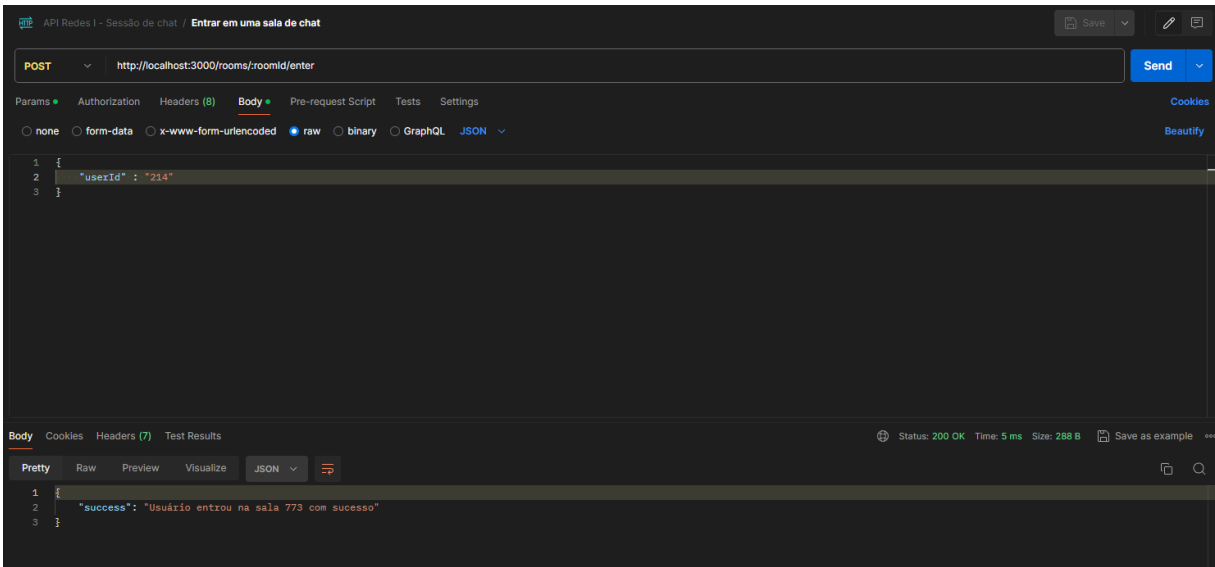
Remoção de uma nova sala - sala existente



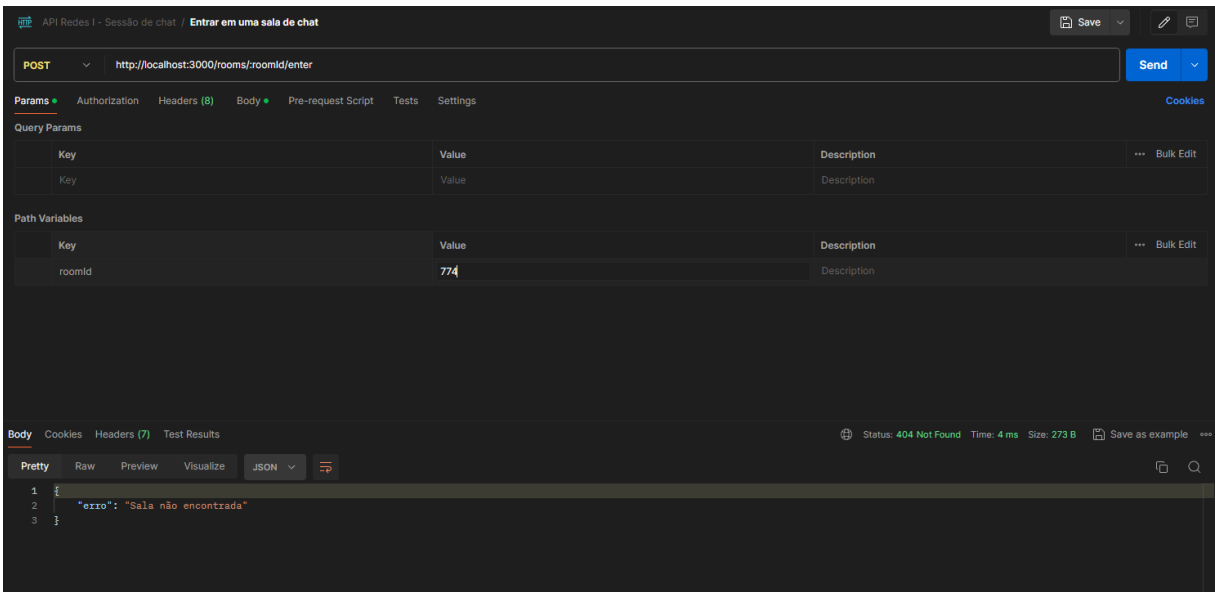
Remoção de uma nova sala - sala não existente



Login em uma sala - sala existente



Login em uma sala - sala não existente



Logout de uma sala - sala existente

API Redes I - Sessão de chat / Sair de um sala de chat

Save

POSThttp://localhost:3000/rooms/roomid/leaveSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Description		Bulk Edit
Key	Value	Description		

Path Variables

Key	Value	Description		Bulk Edit
roomid	773	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 4 msSize: 286 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "sucesso": "Usuário saiu da sala 773 com sucesso"
3 }
```

Logout de uma sala - sala não existente

API Redes I - Sessão de chat / Sair de um sala de chat

Save

POSThttp://localhost:3000/rooms/roomid/leaveSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Query Params

Key	Value	Description		Bulk Edit
Key	Value	Description		

Path Variables

Key	Value	Description		Bulk Edit
roomid	774	Description		

BodyCookiesHeaders (7)Test Results

Status: 404 Not FoundTime: 4 msSize: 273 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "erro": "Sala não encontrada"
3 }
```

Logout de um usuário específico em uma sala - usuário existente com sala existente

API Redes I - Sessão de chat / Remove um usuario de uma sala

DELETE http://localhost:3000/rooms/roomld/users/userld

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Path Variables

Key	Value	Description	Bulk Edit
roomld	773		
userld	214	Description	

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 2 ms Size: 298 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": "Usuário 214 foi removido da sala 773 com sucesso"
3 }
```

Logout de um usuário específico em uma sala - usuário não existente com uma sala existente

API Redes I - Sessão de chat / Remove um usuario de uma sala

DELETE http://localhost:3000/rooms/roomld/users/userld

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Path Variables

Key	Value	Description	Bulk Edit
roomld	787		
userld	215	Description	

Body Cookies Headers (7) Test Results

Status: 404 Not Found Time: 3 ms Size: 277 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "erro": "Usuário não encontrado"
3 }
```

Logout de um usuário específico em uma sala - usuário existente com uma não sala existente

API Rest Client - Sessão de chat / Remove um usuário de uma sala

DELETE http://localhost:3000/rooms/roomId/users/userId

Params: Authorization, Headers (6), Body, Pre-request Script, Tests, Settings

Query Params

Key	Value	Description
Key	Value	Description

Path Variables

Key	Value	Description
roomId	773	
userId	215	Description

Body: Cookies, Headers (7), Test Results

Status: 404 Not Found Time: 4 ms Size: 273 B

Body (JSON)

```
1 {
2   "erro": "Sala não encontrada"
3 }
```

Logout de um usuário específico em uma sala - usuário não está sala existente

API Rest Client - Sessão de chat / Remove um usuário de uma sala

DELETE http://localhost:3000/rooms/roomId/users/userId

Params: Authorization, Headers (6), Body, Pre-request Script, Tests, Settings

Query Params

Key	Value	Description
Key	Value	Description

Path Variables

Key	Value	Description
roomId	787	
userId	824	Description

Body: Cookies, Headers (7), Test Results

Status: 404 Not Found Time: 4 ms Size: 293 B

Body (JSON)

```
1 {
2   "erro": "Usuário não está na sala especificada"
3 }
```