API-RESTful-chat para Gerenciamento de Salas de Chat

Autores: Arthur Lara Panzera
Bernardo Rezende Santos Pereira
Henrique Brocco Marra
Lucas Jose de Freitas

2024

Sumário

Prefácio

O objetivo deste trabalho é colocar em prática os conhecimentos obtidos pela disciplina de Redes I pela PUC MINAS, ministrada pelo Dr.Ricardo Carlini Sperandio. Para isso, este projeto se propõe a explorar o uso de uma API RESTful para o desenvolvimento de um sistema de gerenciamento de salas de chat, que foi montado utilizando o NodeJs.

O foco principal deste trabalho é a implementação de um sistema de chat que permita um gerenciamento de usuários, criação e gerência de salas de conversa e a troca de mensagens em tempo real entre usuários. Além disso, serão apresentados conceitos fundamentais de APIs RESTful, incluindo os princípios de design REST, métodos HTTP e boas práticas de desenvolvimento.

1. Introdução ao Documento

O trabalho busca desenvolver uma API RESTful explorando e implementando o protocolo HTTP e permitirá a comunicação em tempo real entre os usuários de um sistema de chat. Ela será responsável por gerenciar os usuários, as sessões de chat e as mensagens. Ele permitirá a aplicação prática dos conceitos vistos, e também proporcionará uma compreensão mais profunda de como os protocolos da camada de aplicação são usados em aplicações do mundo real.

1.1. Tema

A camada de aplicação é a responsável por fazer a comunicação entre o sistema e a interface de visualização do usuário, e ela utiliza de vários protocolos para isso, como o HTTP. Ele é responsável por permitir a transferência de dados na web. É usado principalmente para transmitir informações em formato de texto, imagens, vídeos, entre outros, entre o servidor web e o navegador de um usuário.

O HTTP opera em um modelo de solicitação-resposta, o cliente envia uma solicitação para o servidor, e o servidor retorna uma resposta, cada solicitação é independente, o que significa que o servidor não mantém informações entre solicitações diferentes, sendo um protocolo sem estado. Ele faz isso definindo métodos para tratar com as informações trafegadas, sendo as principais: GET, para recuperar informações, PUT, para atualizar informações e DELETE, para excluir informações.

1.2. Objetivo do Projeto

O objetivo do projeto é colocar em prática alguns conhecimentos adquiridos na disciplina de redes de computadores, focando na camada de aplicação do modelo TCP/IP, através do desenvolvimento de uma API RESTful para um sistema de chat em tempo real.

A API será responsável por gerenciar usuários, sessões de chat e mensagens, permitindo a comunicação em tempo real entre os usuários, explorando os recursos do protocolo HTTP.

2. Descrição Geral do Sistema

Esse projeto descreve uma API para gerenciamento de mensagens, usuários e salas de chat. Ele é dividido em três coleções distintas:

API Redes I - Gerenciamento de Mensagens

- Contém operações para manipular mensagens em salas de chat:
- Enviar uma mensagem a uma sala: Permite enviar uma mensagem para uma sala específica, fornecendo o 'userId' do remetente e o conteúdo da mensagem.
- Receber mensagem de uma sala: Essa operação provavelmente deveria ser mais detalhada, especificando como receber as mensagens de uma sala específica.
- Enviar mensagem direta a outro usuário: Permite enviar uma mensagem direta a um usuário específico, identificado pelo 'recipientId', vindo do ‘senderId’.
- Obter mensagens diretas de um usuário: Permite obter todas as mensagens diretas de um usuário com base no 'userId' fornecido.

API Redes I - Gerenciamento de Usuários

- Foca em operações relacionadas à gestão de usuários:
- Cadastro de usuário: Permite criar um novo usuário fornecendo o nome de usuário ('username').
- Obter informações de um usuário: Obtém informações específicas de um usuário com base no 'userId'.
- Autenticar um usuário: Permite autenticar um usuário fornecendo o 'username' para login.

API Redes I - Sessão de Chat

- Responsável pelo gerenciamento das salas de chat:
- Criar nova sala: Cria uma nova sala de chat com um nome específico.
- Deletar uma sala: Remove uma sala de chat com base no ID da sala ('roomId').
- Entrar em uma sala de chat: Permite que um usuário entre em uma sala de chat especificada pelo 'roomId'.

- Sair de uma sala de chat: Remove um usuário de uma sala de chat específica ('roomId') com base no 'userId'.

- Remover um usuário de uma sala: Remove um usuário específico ('userId') de uma sala específica ('roomId').

- Salas ativas: Obtém informações sobre as salas de chat ativas.

Visão Geral do Projeto

- Tecnologias Utilizadas: A API parece estar utilizando o protocolo HTTP sobre o 'localhost' na porta '3000'.

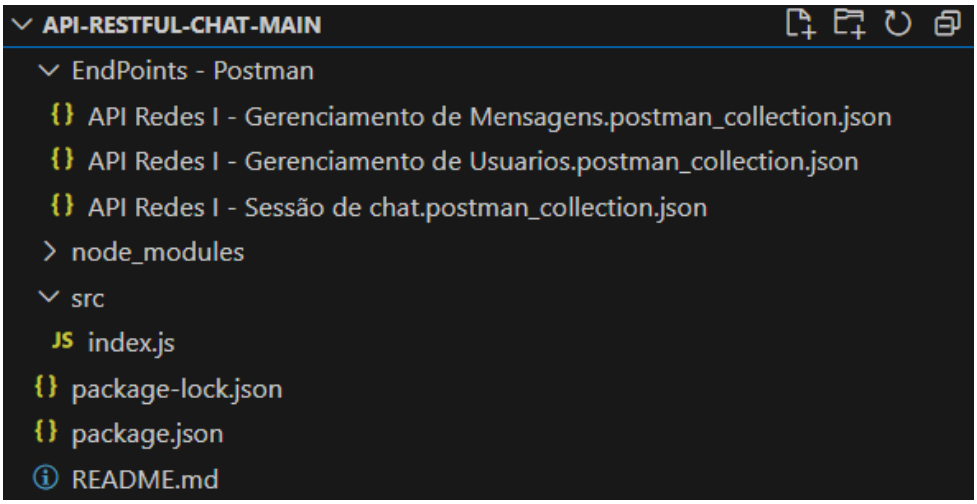
- Funcionalidades Principais: Permite o gerenciamento de mensagens em salas, a gestão de usuários e o controle das sessões de chat (salas).

- Recursos Adicionais: Seria útil ter mais detalhes sobre as respostas esperadas para cada requisição, bem como informações sobre possíveis autenticações ou autorizações implementadas.

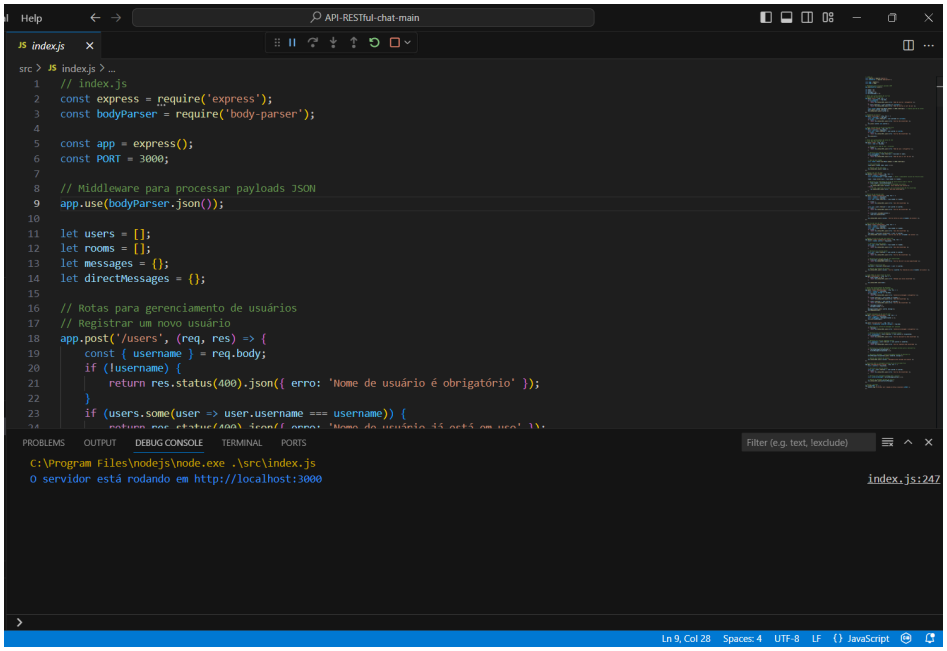
Cada coleção define diferentes endpoints (pontos finais) que podem ser utilizados para interagir com o sistema de mensagens, usuários e salas de chat. Esta estrutura permite testar e documentar de forma eficiente a funcionalidade da API usando o Postman ou ferramentas semelhantes.

3. Implementação

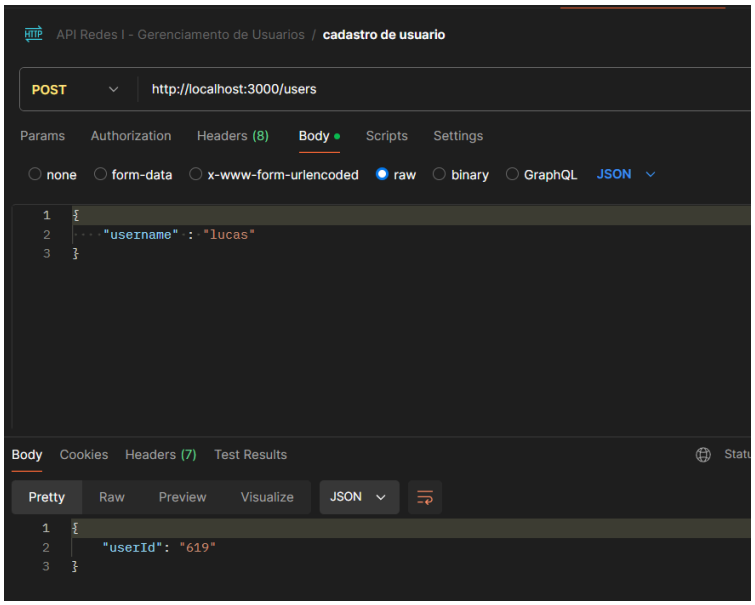
Para a elaboração e implementação foi utilizado o visual studio code (VSCode) com NodeJs, pois foi observado que era uma linguagem e uma IDE que a maioria do grupo possuía um conhecimento básico, além da infinidade de conteúdo disponível para pesquisa na internet e em livros. Para a estrutura do trabalho, foi decidido uma organização por meio de pastas, separando os módulos utilizados, os EndPoints e o arquivo index.



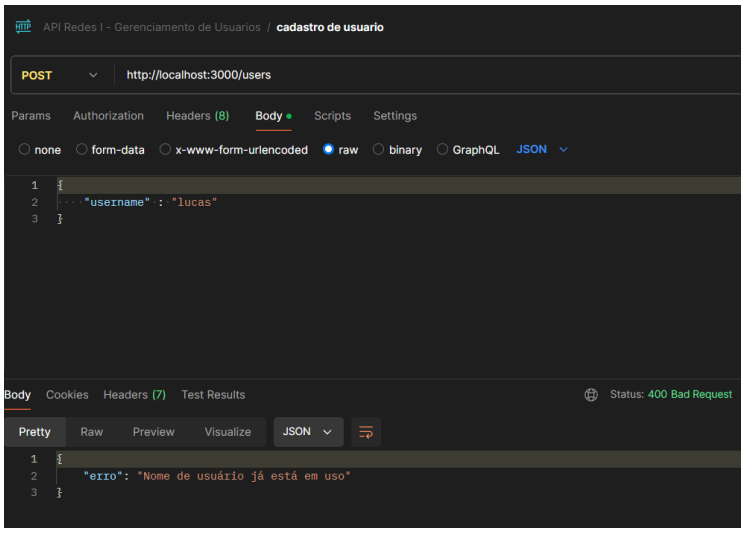
Para a realização dos testes foi utilizado o software Postman, para executar os arquivos JSON, juntamente com o VSCode executando a API, abaixo temos os exemplos dos testes:



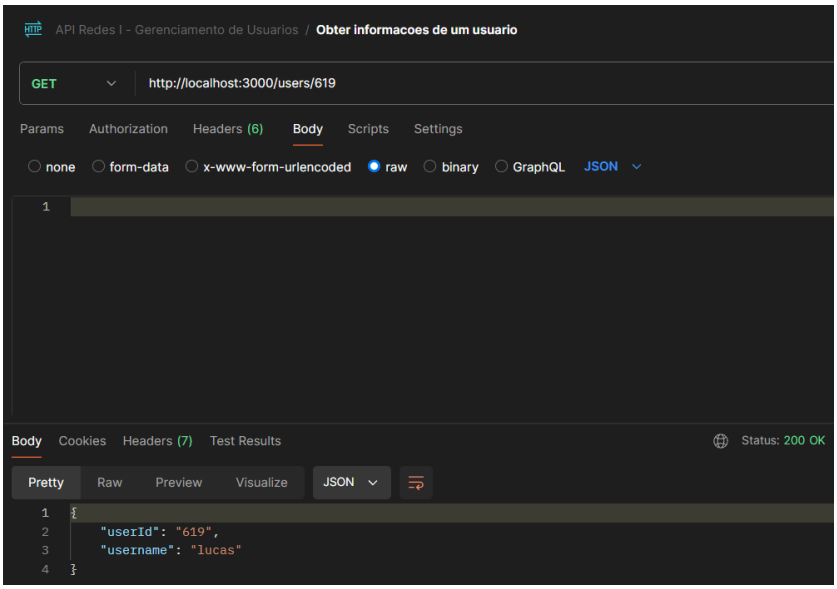
Ao cadastrar um usuário com o nome “lucas”, temos o seguinte retorno:



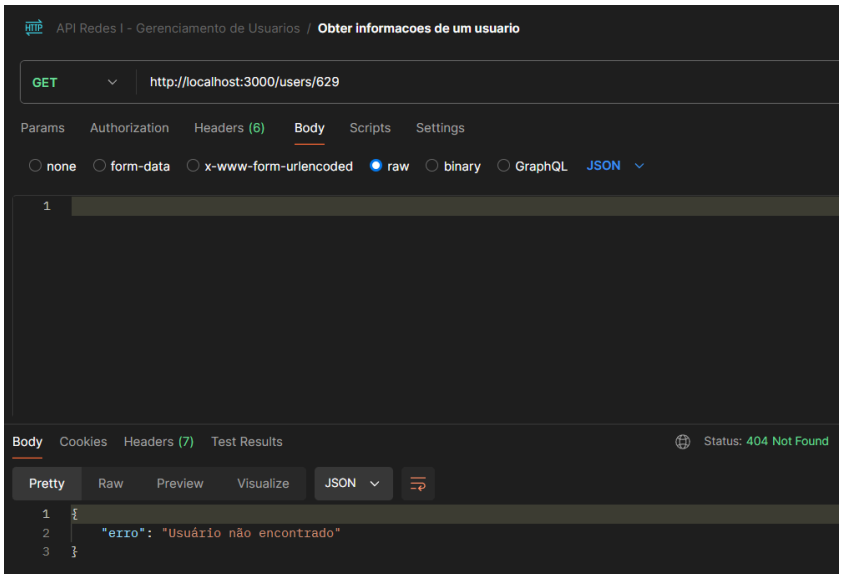
Ao efetuarmos um cadastro novamente com o mesmo nome, obtemos um status 400, pois o nome já está sendo utilizado.



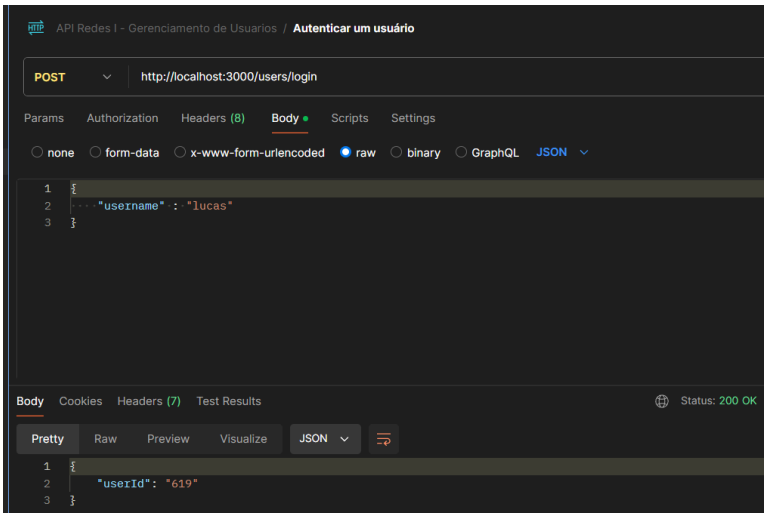
Ao realizamos um get para visualizar o usuário, quando inserimos o id correto:



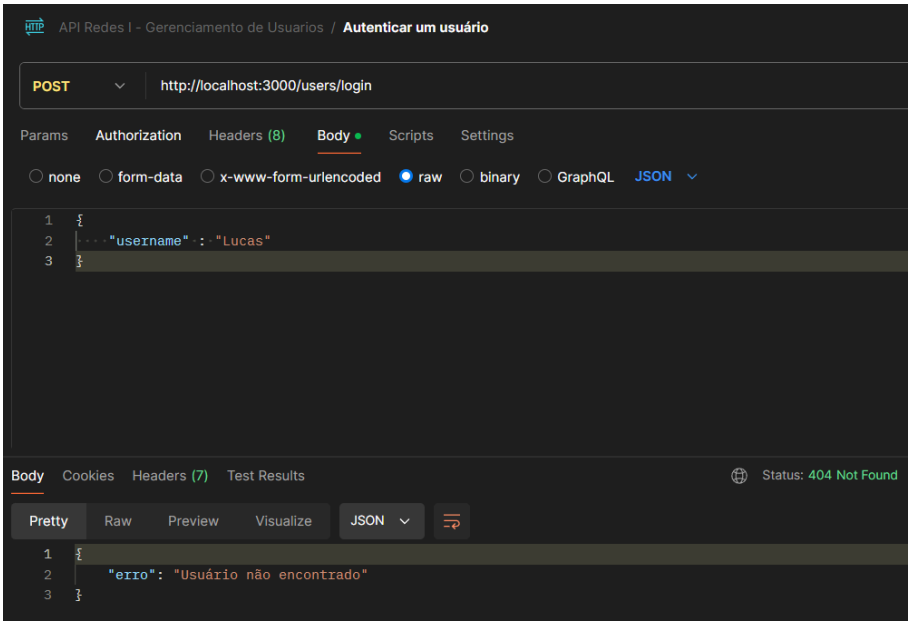
Quando inserimos o id incorreto, obtemos um erro 404 - Not found:



Quando vamos realizar uma autenticação é necessário passar o username, quando passamos um que existe temos um retorno de 200 - Ok:

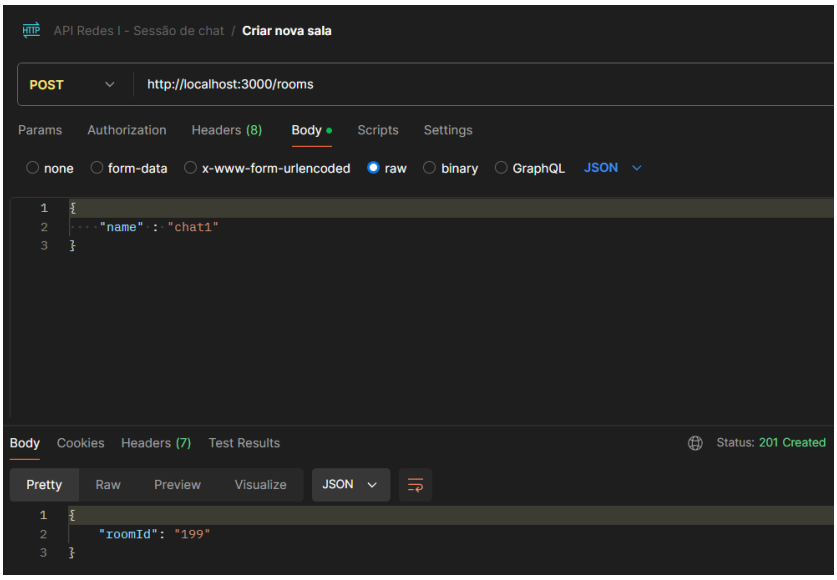


Ao tentar realizar a autenticação com um usuário que não existe não obtemos sucesso, e sim, um erro 404 de usuário não encontrado.

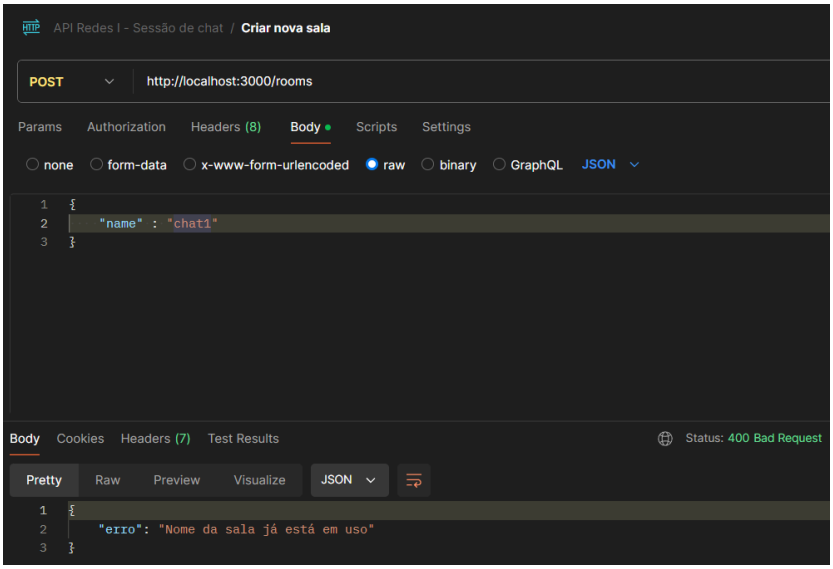


Para os endpoints de Sessão de chat temos 6 possibilidades, criação de uma nova sala, remoção de uma sala, entrar ou sair de uma sala de chat, remover um usuário de uma sala e ver quais salas estão ativas:

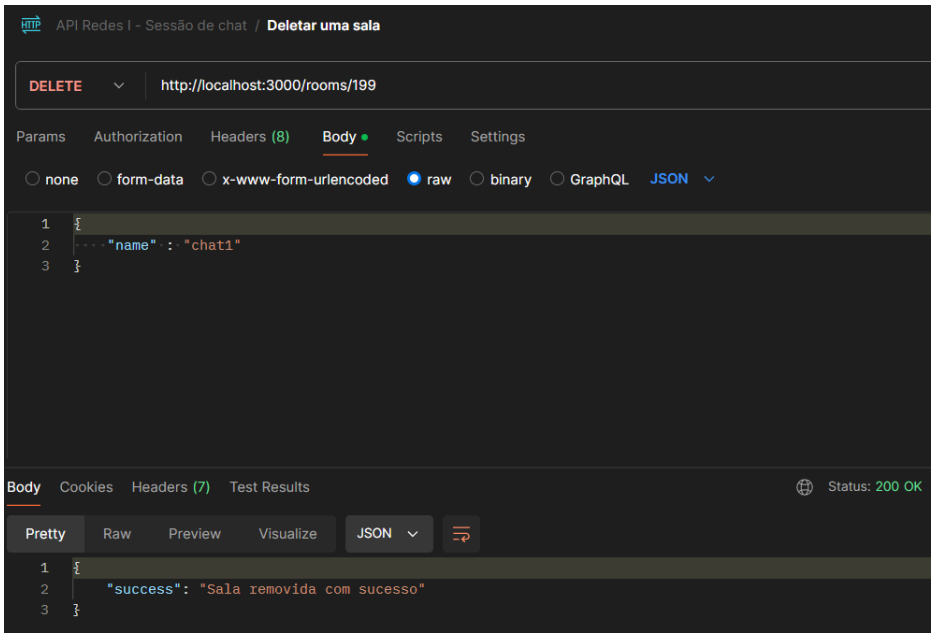
Criar nova sala 201:



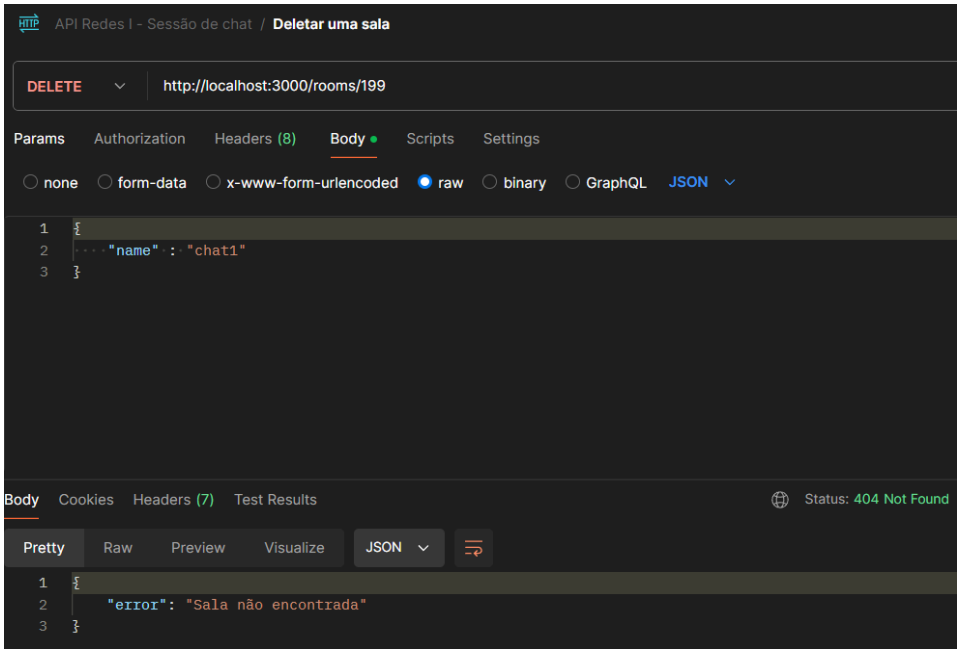
Sala 400:



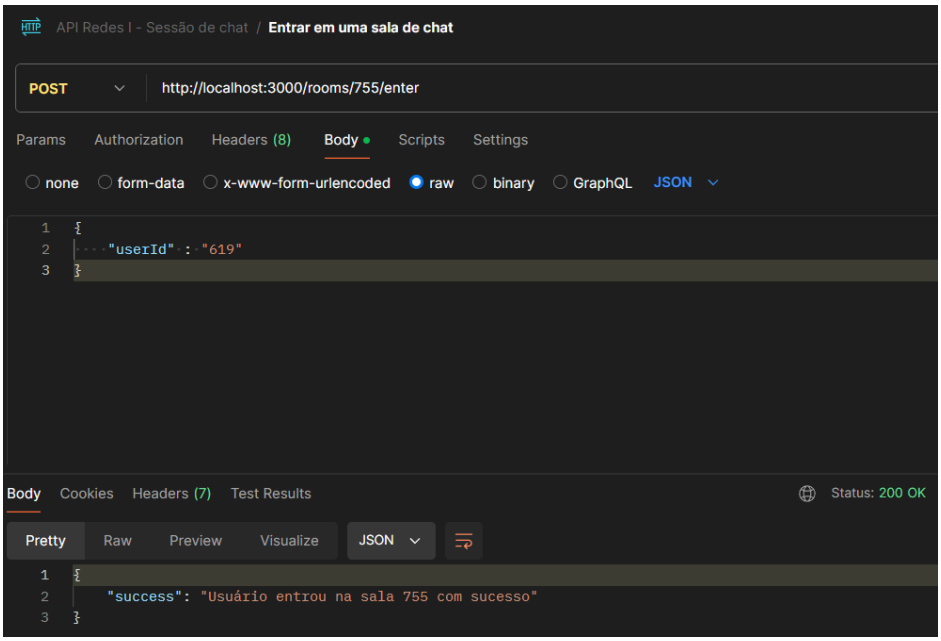
Delete 200:



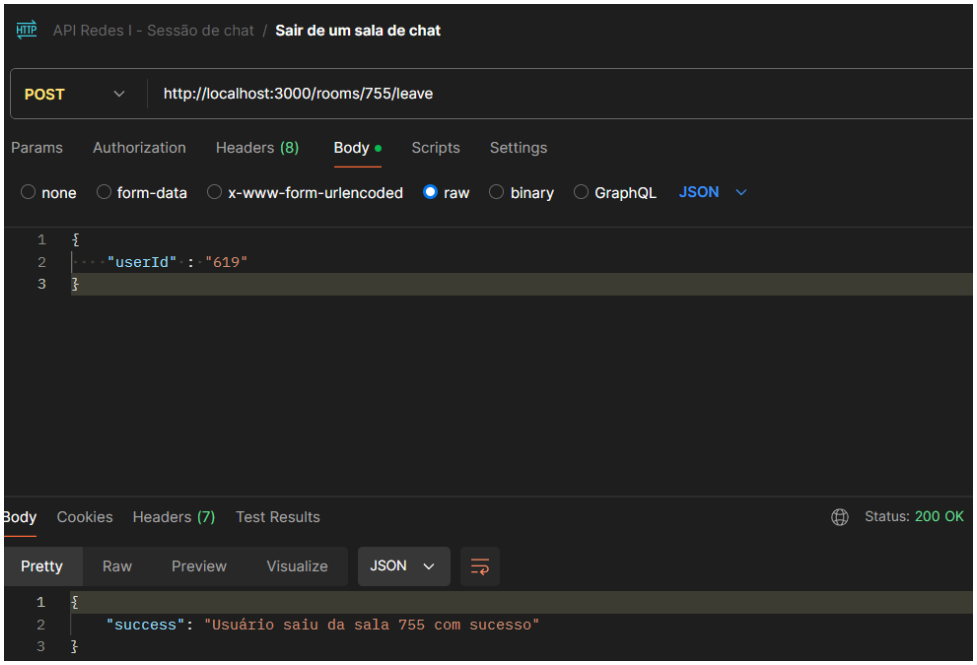
Delete 404:



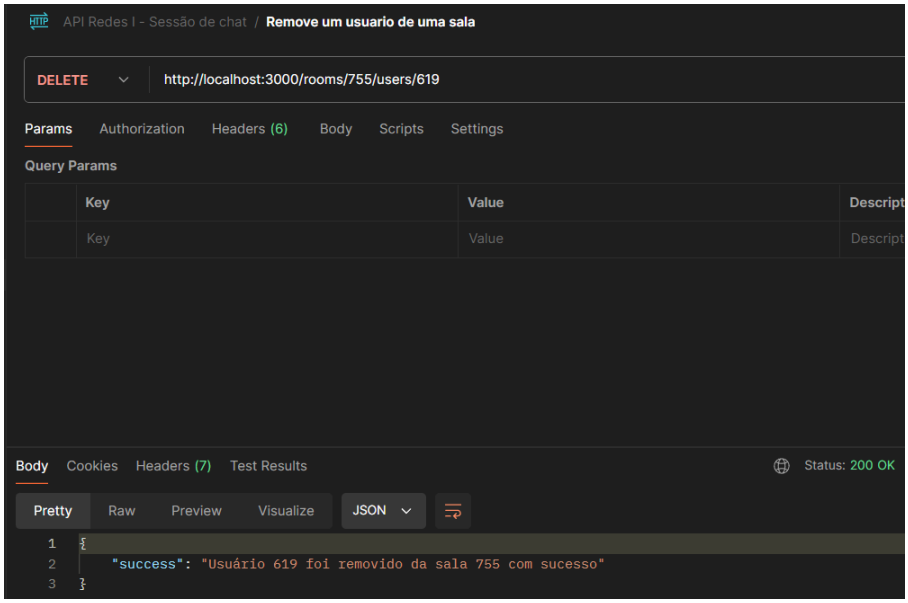
Para realizar a inserção em uma sala é necessário o Id do usuário e o id da sala, caso um dos dois ou ambos estejam incorretos dará erro 404 not found. Caso ambos estejam certos temos sucesso 200, conforme a imagem abaixo:



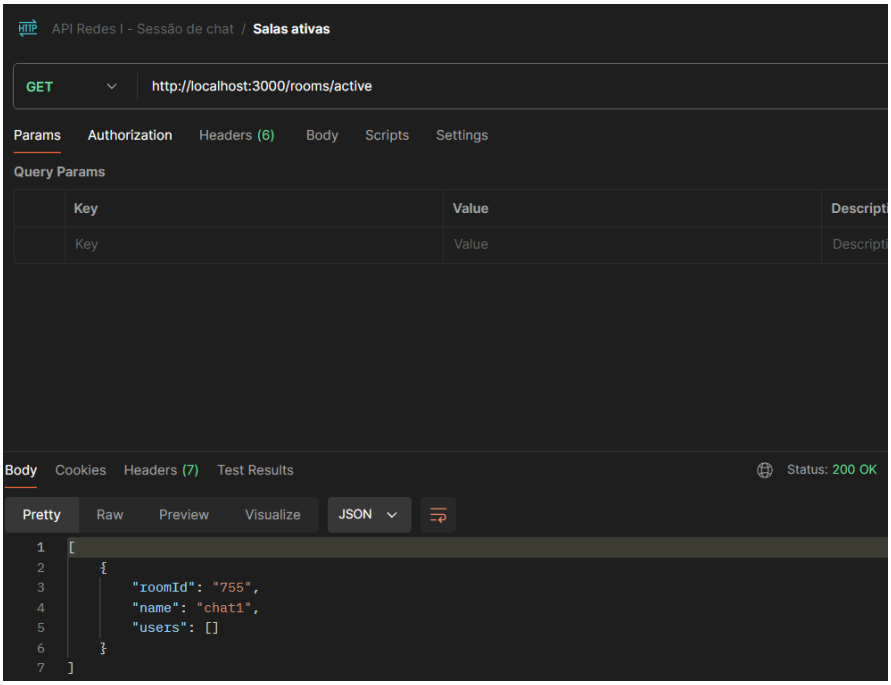
O mesmo comportamento é esperado para se sair da sala de chat:



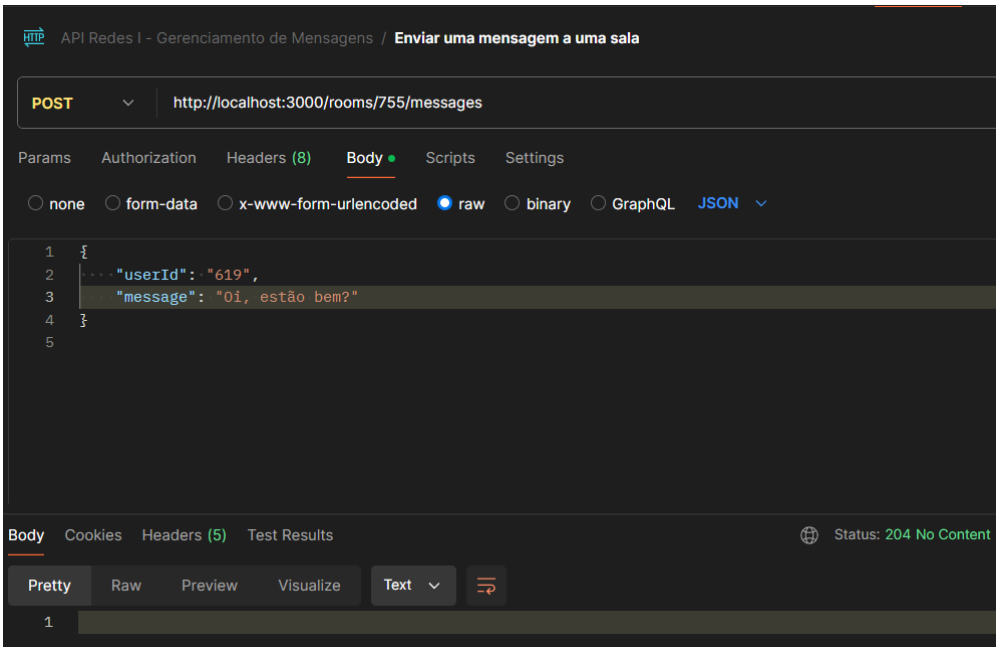
Para remover um usuário de uma sala é exigido os mesmos parâmetros dos dois casos anteriores:



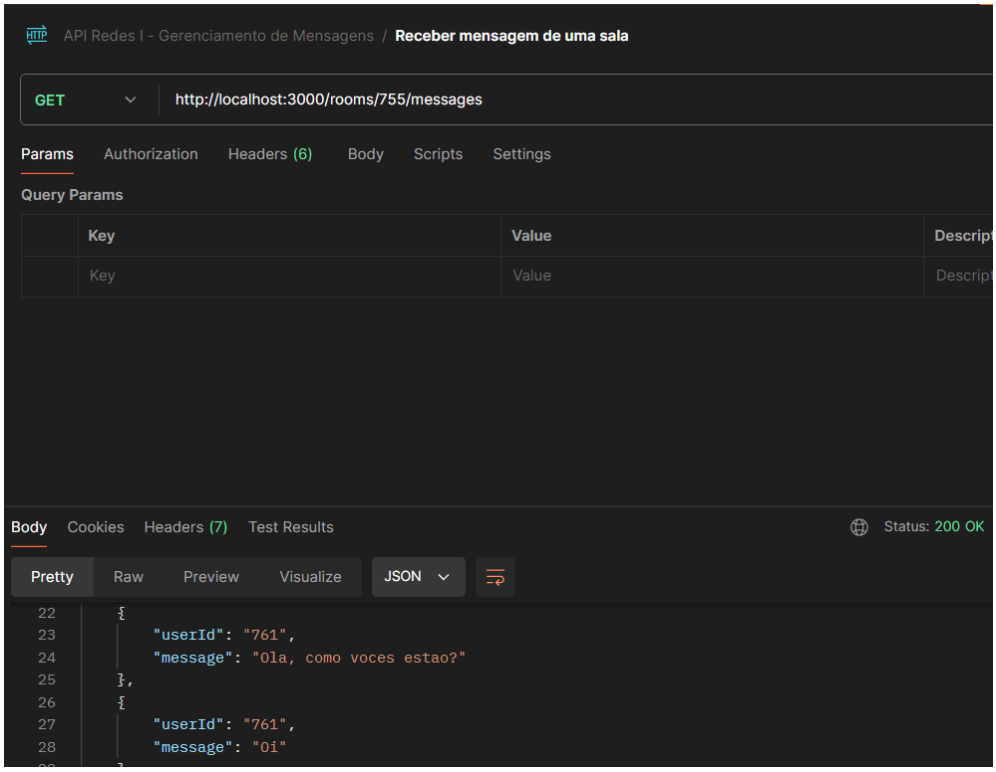
Para visualizar as salas ativas é só realizar o GET:



Por fim, temos o envio e o recebimento das mensagens, tendo 4 vertentes, envio e recebimento de mensagens diretas ou comuns. Para o envio, é necessário o corpo da mensagem, o usuário que está enviando e a sala em que está sendo enviada a mensagem, conforme abaixo:



Para receber precisamos apenas do id sala:



As mensagens diretas funcionam da mesma forma que as mensagens comuns, mas direcionando para o id do usuário de destino.

4. Conclusões e Considerações Finais

Concluindo este trabalho acadêmico, desenvolvemos um sistema abrangente para gerenciamento de usuários, salas de chat e mensagens. O sistema oferece funcionalidades essenciais para interação entre usuários em um ambiente virtual.

Para o gerenciamento de usuários, implementamos endpoints que permitem registro de novos usuários, autenticação e obtenção de informações de usuário específico. O sistema também facilita a criação e administração de salas de chat, permitindo que usuários entrem e saiam de salas, além de adicionar ou remover participantes. Operações como listagem de salas ativas e exclusão de salas também foram contempladas.

A funcionalidade principal de troca de mensagens foi implementada para permitir o envio direto entre usuários e dentro de salas de chat, com histórico de mensagens mantido para referência futura. Os endpoints dedicados a mensagens suportam o envio, recebimento e recuperação de mensagens, possibilitando uma experiência completa de comunicação em tempo real.

Este sistema é flexível e pode ser expandido com recursos adicionais, como notificações, privilégios de administrador para salas, busca avançada de mensagens, entre outros. É fundamental garantir a segurança e a escalabilidade do sistema, especialmente em ambientes de produção, utilizando boas práticas de desenvolvimento e gerenciamento de dados.

Em suma, este trabalho demonstra um projeto sólido para uma aplicação de chat, oferecendo uma base robusta para a interação entre usuários online, com potencial para evolução e personalização conforme as necessidades do ambiente e dos usuários.

Bibliografia

Pinto, Pedro. RESTful API com node.js. Medium, 2017. Disponível em: <https://pedrompinto.medium.com/restful-api-com-node-js-e7e8a710256d>. Acesso em: 03/05/2024.

NodeJS Learning. Node JS. Disponível em: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>. Acesso em: 03/05/2024.

Postman. Postman. Disponível em: <https://www.postman.com>. Acesso em: 05/05/2024.

Visual Studio Code. Visual Studio Code. Disponível em: <https://code.visualstudio.com>. Acesso em: 01/05/2024.

Prado, Jean. Como fazer referência bibliográfica de site nas normas ABNT. Tecnoblog, 2019. Disponível em: <https://tecnoblog.net/responde/referencia-site-abnt-artigos/>. Acesso em: 10/05/2024.

Stack Overflow. Stack Overflow. Disponível em: <https://stackoverflow.com> . Acesso em: 03/05/2024.