



DOCUMENTAÇÃO API

<BACKEND>

Armazenamento de Dados em Ambientes Distribuídos (ADAD)



MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES E
INFORMÁTICA (METI)

DOCENTE: JOÃO MATOS

Grupo 10

Rui Jesus 99009

Ricardo Paixão 99224

Afonso Taveira 99198

Pedro Cruz 99379

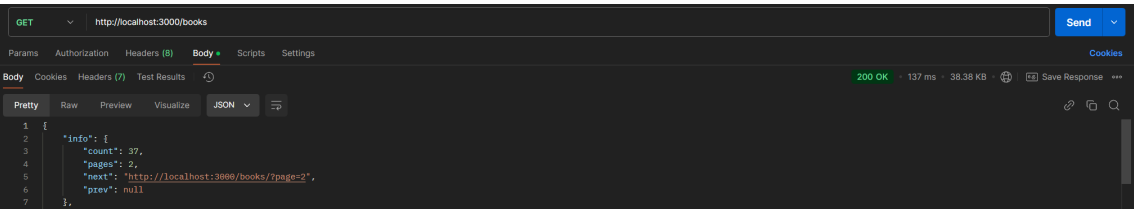
Pedro Fernandes 99262

Informação e paginação

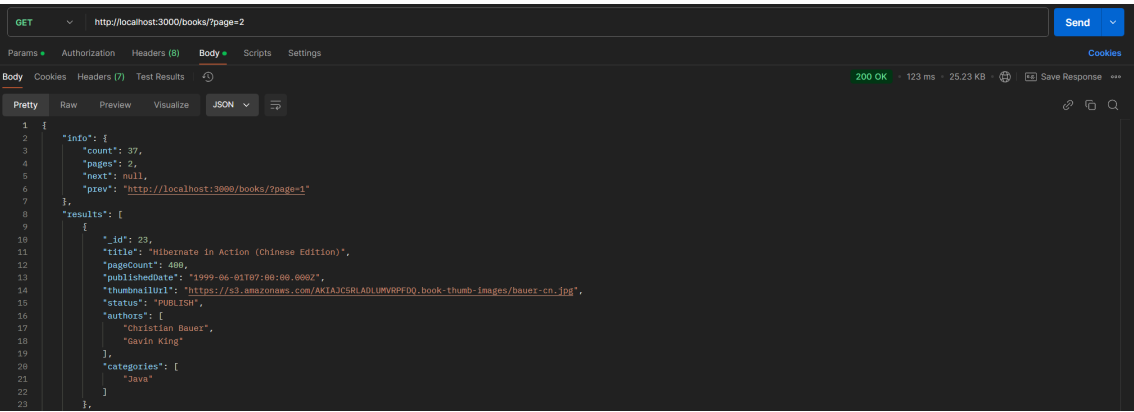
Este API faz paginação automática das respostas. São apenas enviados até 20 documentos por página.

Cada mensagem contém um objeto `info` que contém a informação sobre a resposta.

Chave	Tipo	Descrição
count	int	O número de documentos da resposta.
pages	int	O número de páginas.
next	string (url)	O link para a próxima página (se existir)
prev	string (url)	O link para a página anterior (se existir)



É possível aceder a outras páginas a partir do parâmetro `pages`. Se a página não for especificada, será sempre mostrada a primeira. Por exemplo, para aceder à página 2, adiciona-se `?pages=2` no fim do URL.



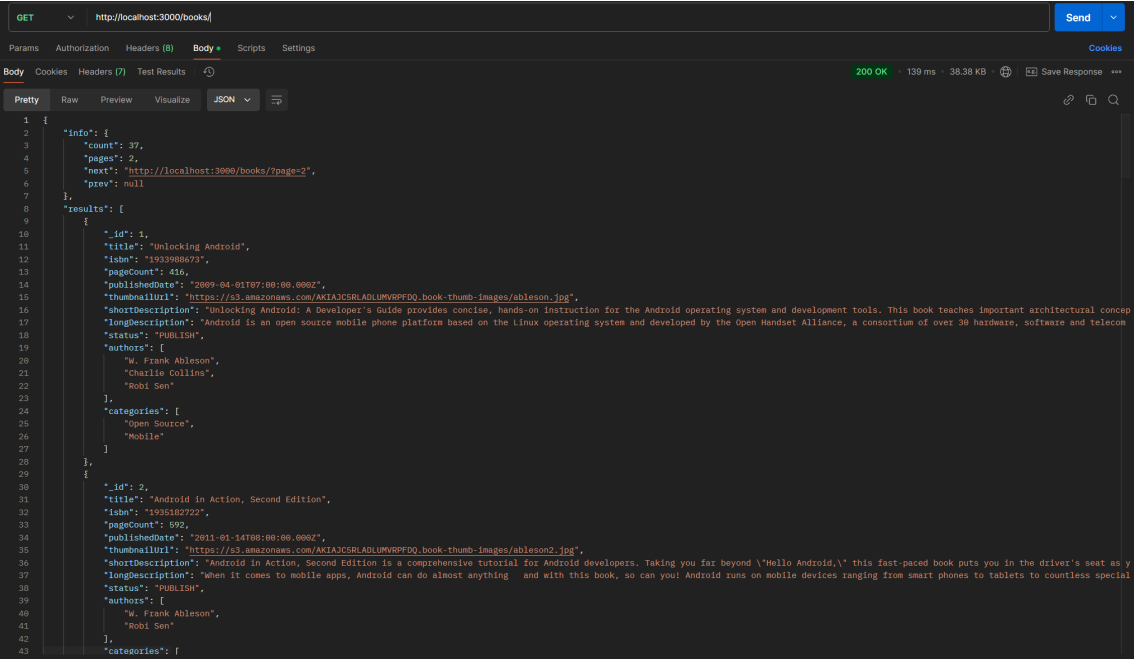
Book

Book schema

Chave	Tipo	Descrição
_id	int	O id do livro.
title	string	O título do livro.
isbn	string	O isbn do livro.
pageCount	int	O número de páginas do livro.
publishedDate	date	A data de publicação do livro.
thumbnailUrl	String (url)	Link para o livro.
shortDescription	string	Curta descrição do livro.
longDescription	string	Longa descrição do livro.
status	string	Estado do livro.
authors	array	Lista de autores do livro.
categories	array	Lista de categorias do livro.

Obter todos os livros

É possível obter todos os livros a partir do endpoint `/books`

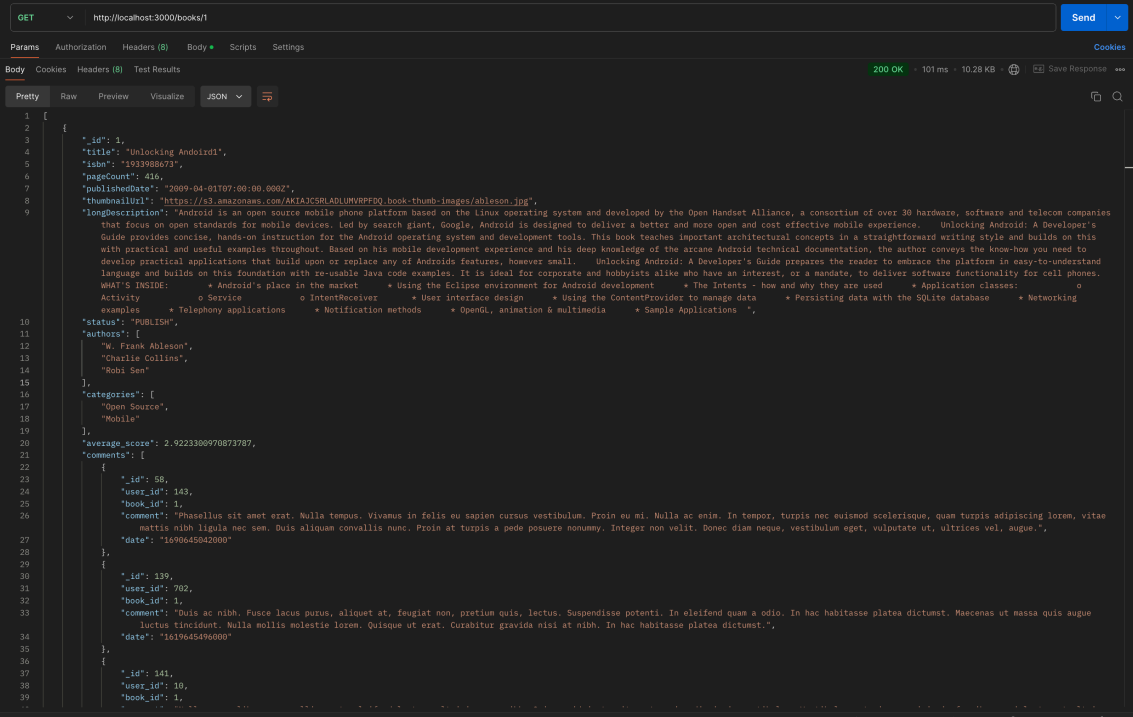


The screenshot shows a REST client interface with a GET request to `http://localhost:3000/books/` sent successfully. The response is a JSON array of two book objects. The first book is 'Unlocking Android' by W. Frank Ableson, and the second is 'Android in Action, Second Edition' by W. Frank Ableson and Robi Sen. The response status is 200 OK, and the body is displayed in a pretty-printed JSON format.

```
1 {
2   "info": {
3     "count": 37,
4     "pages": 2,
5     "next": "http://localhost:3000/books/?page=2",
6     "prev": null
7   },
8   "results": [
9     {
10      "_id": 1,
11      "title": "Unlocking Android",
12      "isbn": "1933988973",
13      "pageCount": 416,
14      "publishedDate": "2009-04-01T07:00:00.000Z",
15      "thumbnailUrl": "https://s3.amazonaws.com/AKIAJCSRLADLUMVRPFDQ_book-thumb-images/ableson.jpg",
16      "shortDescription": "Unlocking Android: A Developer's Guide provides concise, hands-on instruction for the Android operating system and development tools. This book teaches important architectural concepts and provides a comprehensive tutorial for Android developers.",
17      "longDescription": "Android is an open source mobile phone platform based on the Linux operating system and developed by the Open Handset Alliance, a consortium of over 30 hardware, software and telecom companies.",
18      "status": "PUBLISHED",
19      "authors": [
20        "W. Frank Ableson",
21        "Charlie Collins",
22        "Robi Sen"
23      ],
24      "categories": [
25        "Open Source",
26        "Mobile"
27      ]
28    },
29    {
30      "_id": 2,
31      "title": "Android in Action, Second Edition",
32      "isbn": "1935182722",
33      "pageCount": 592,
34      "publishedDate": "2011-01-14T08:00:00.000Z",
35      "thumbnailUrl": "https://s3.amazonaws.com/AKIAJCSRLADLUMVRPFDQ_book-thumb-images/ableson2.jpg",
36      "shortDescription": "Android in Action, Second Edition is a comprehensive tutorial for Android developers. Taking you far beyond 'Hello Android,' this fast-paced book puts you in the driver's seat as you learn to build real-world applications.",
37      "longDescription": "When it comes to mobile apps, Android can do almost anything - and with this book, so can you! Android runs on mobile devices ranging from smart phones to tablets to countless special-purpose devices.",
38      "status": "PUBLISHED",
39      "authors": [
40        "W. Frank Ableson",
41        "Robi Sen"
42      ],
43      "categories": [
```

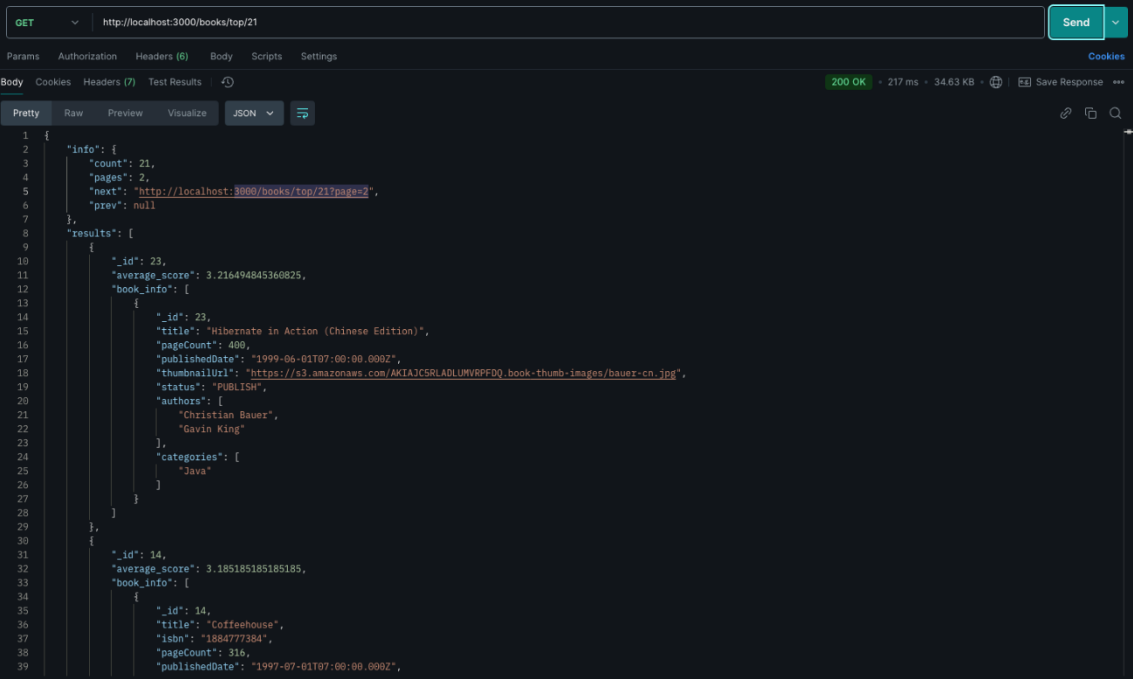
Obter um único livro

Podemos obter um único livro ao adicionar o `_id` como parâmetro: `/books/1`. Vamos receber as informações desse livro com o average score dos scores das reviews dos utilizadores e todos os comentários desse livro



Obter lista de livros com maior média de score

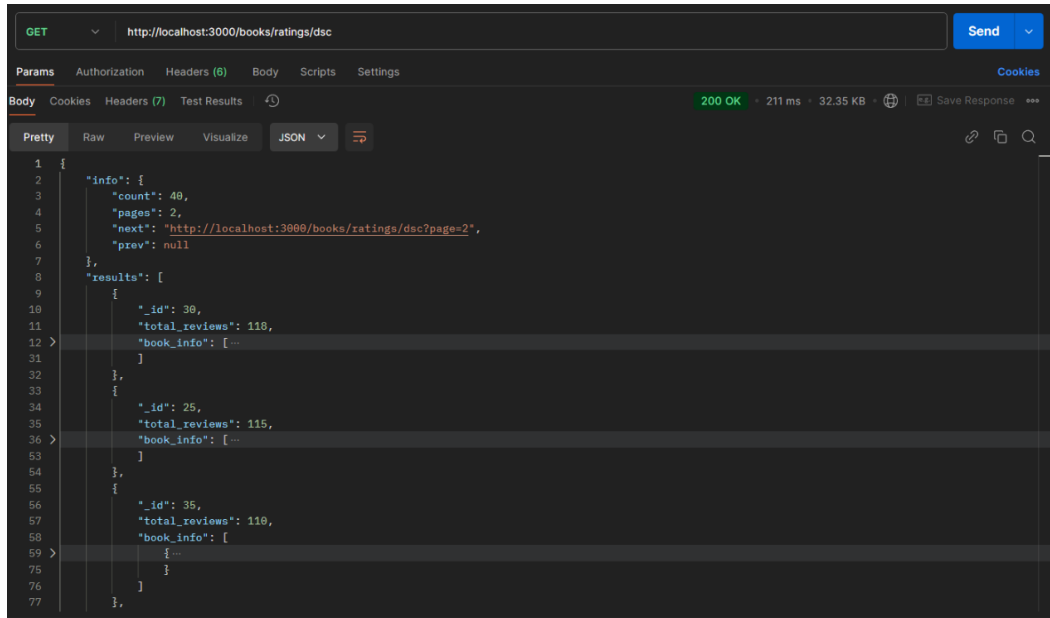
É apresentada uma lista de livros, limitada, utilizando o seguinte endpoint, `/books/top/:limit`



Obter lista de livros ordenado pelo número total de reviews

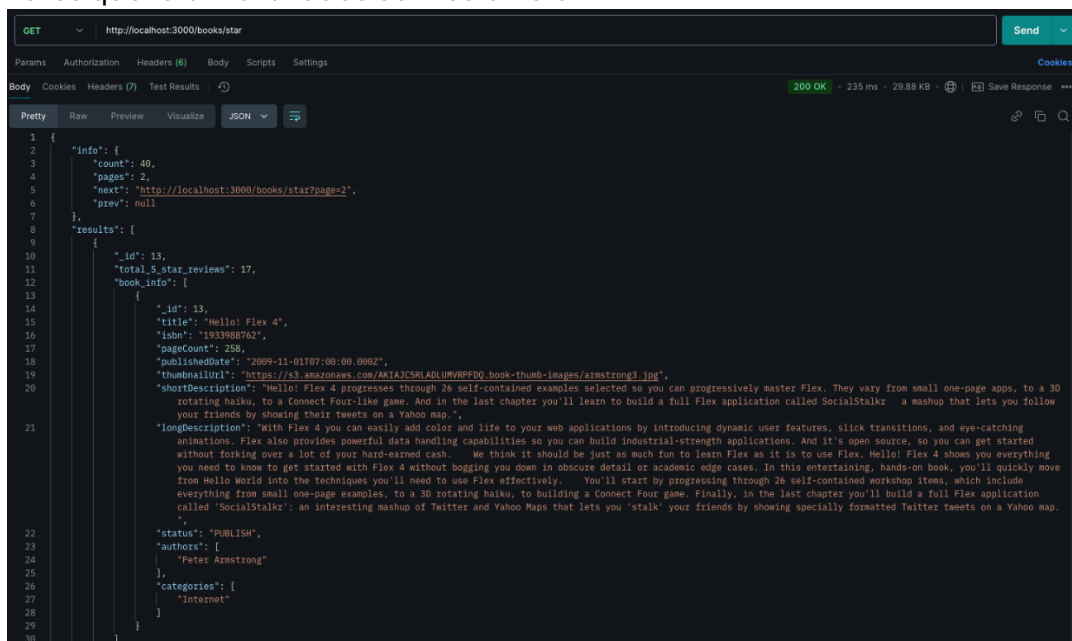
É apresentada uma lista de livros utilizando o seguinte endpoint: `/books/ratings/:order`.

Esta lista pode ser ordenada de forma ascendente (asc) ou decendente (dsc), consoante o número de reviews. Apenas são aceites os parâmetros exatos de “asc” e “dsc”



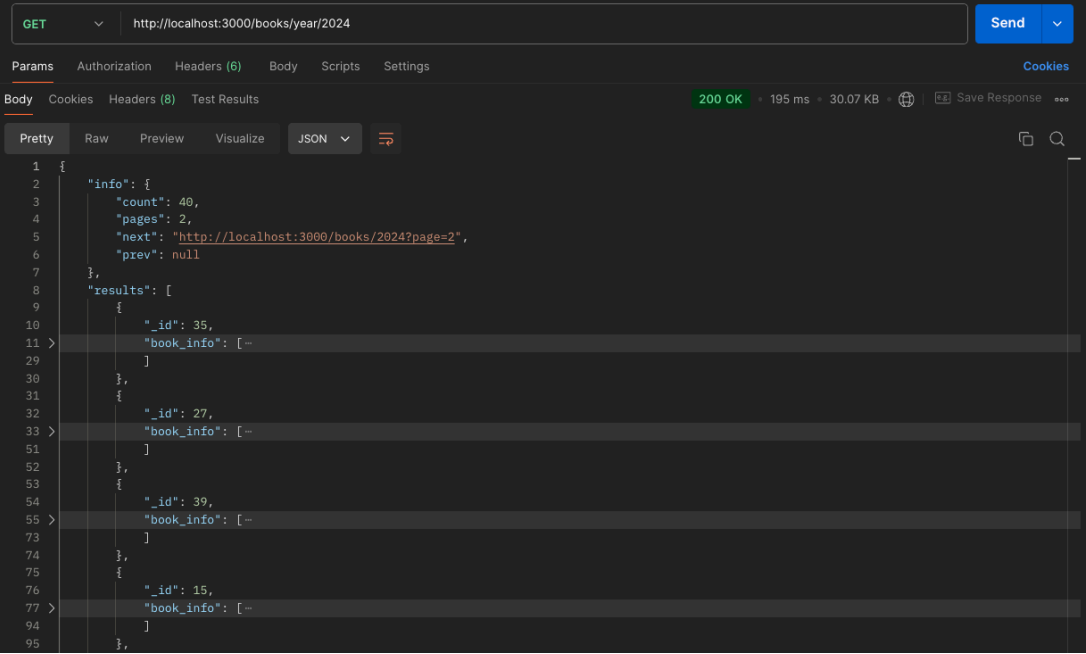
Obter lista de livros com score 5 estrelas

É apresentada uma lista de livros utilizando o seguinte endpoint, `/books/star`. Apenas os livros com reviews de 5 estrelas são apresentados e ainda o total de vezes que foram avaliados com esta nota



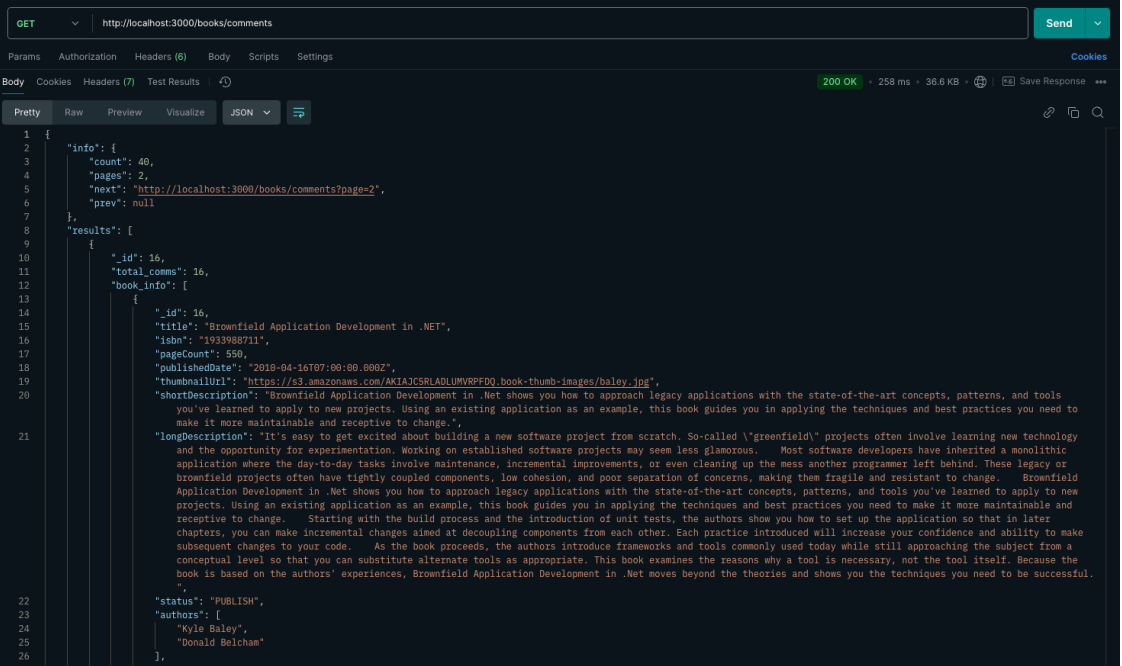
Obter lista de livros avaliados num certo ano

É apresentada uma lista de livros avaliados utilizando o seguinte endpoint: `books/year/:year`. Todos os livros que tiveram uma review no ano pretendido irão aparecer na lista. Serão descartados os pedidos que não têm 4 dígitos ou que não sejam números



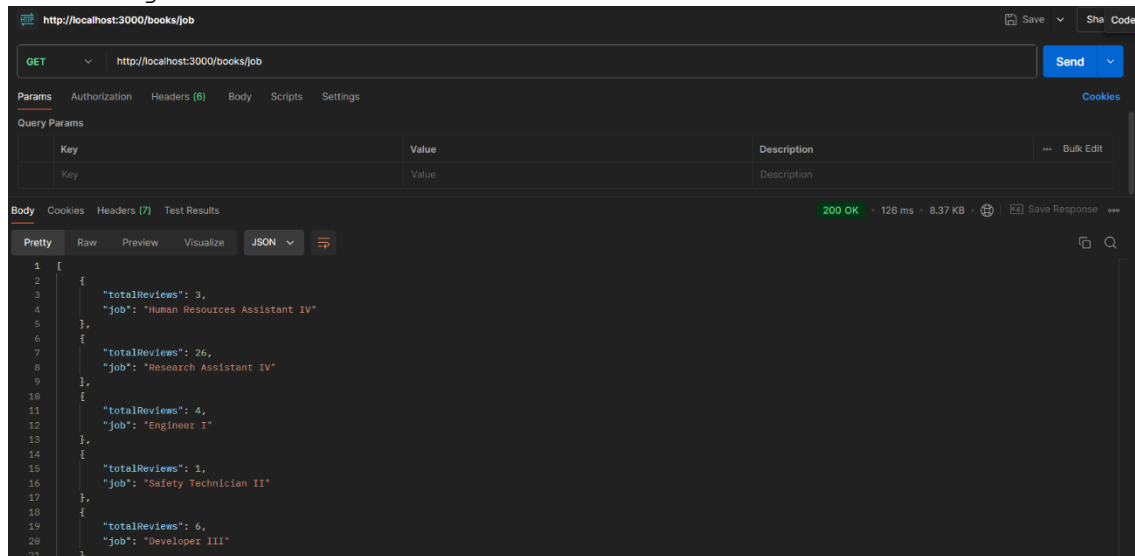
Obter lista de livros com comentários

É apresentada uma lista de livros utilizando o seguinte endpoint, `/books/comments`. Apenas o id dos livros com comentários são apresentados e estão ordenados pelo número de comentários de forma crescente



Obter o número total de reviews por “job”

Podemos obter o número total de reviews por cada “job”, através do endpoint `/books/job`



Obter uma lista de livros filtrada por preço, categoria e/ou autor

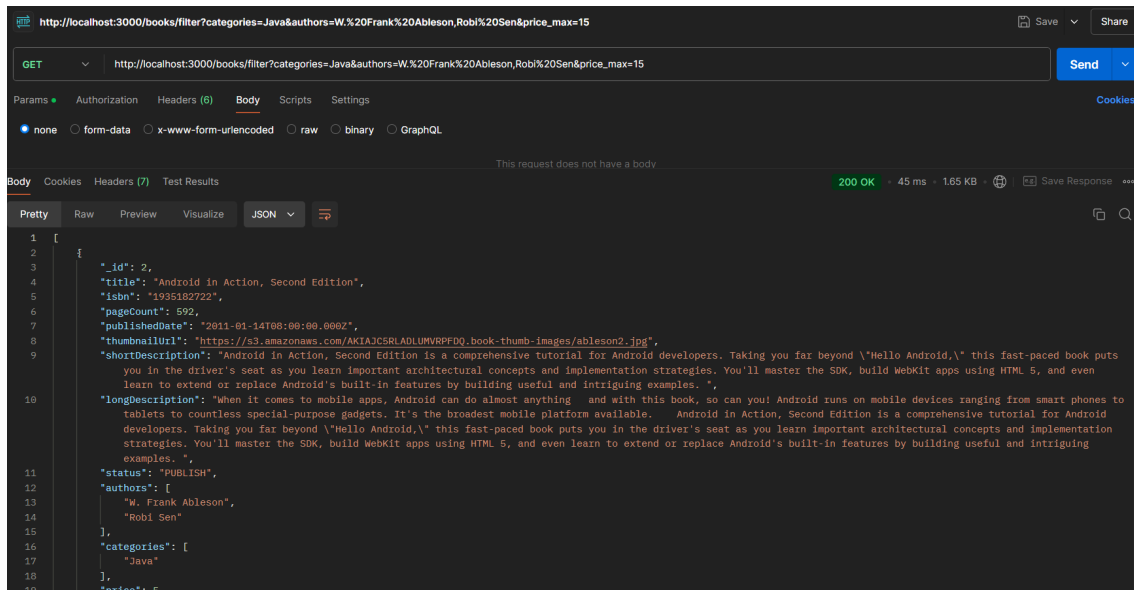
Podemos obter uma lista de livros filtrada por preço, categoria e/ou autor, ao adicionarmos um `?` seguido da query `<query>=<value>`. Podemos encadear várias queries usando o `&`. Podemos ainda ter mais de um valor em cada query, separados usando o `,`

Por exemplo, se quisermos ver quantos livros o author Robi Sen tem, tem de se adicionar

`?categories=Java&authors=W.%20Frank%20Ableson,Robi%20Sen&price_max=15` ao URL

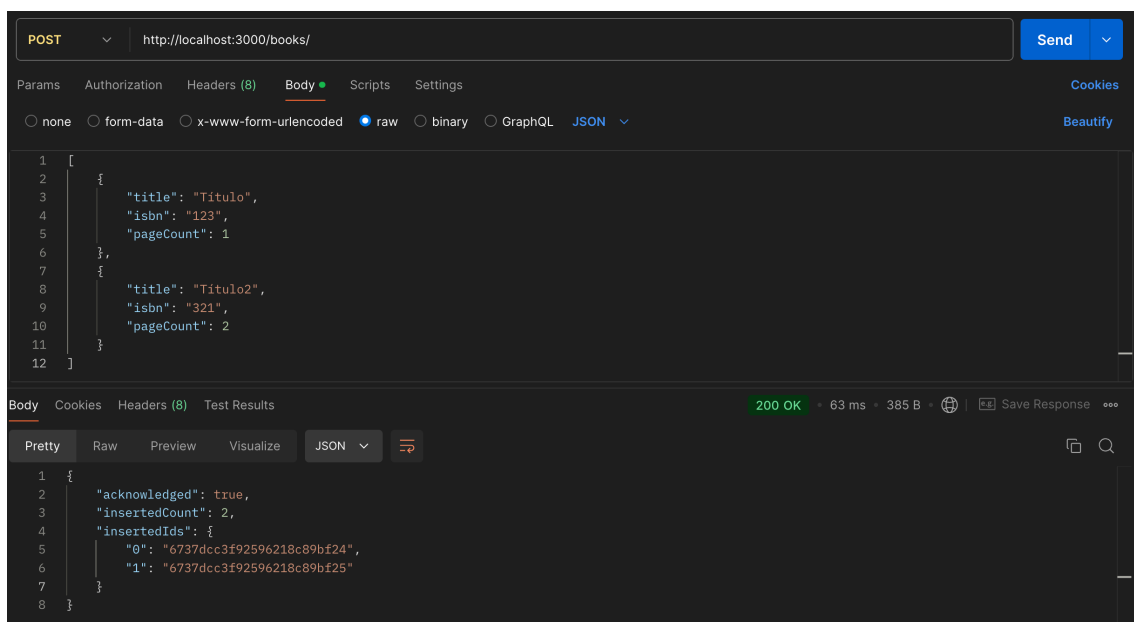
Parâmetros disponíveis:

- `authors`: filtrar pelo nome do autor.
- `categories`: filtrar pela categoria.
- `price`: filtrar pelo preço.
- `price_max`: filtrar pelo preço máximo.
- `price_min`: filtrar pelo preço máximo.



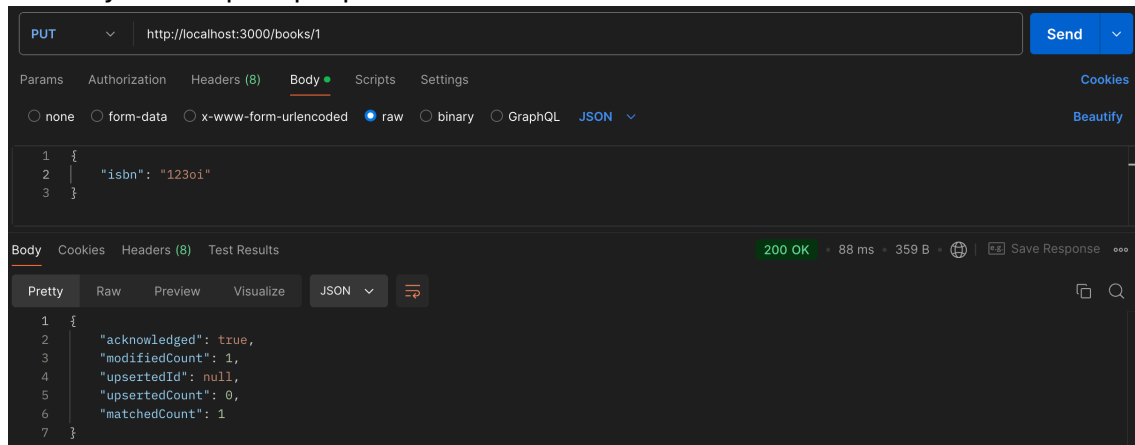
Inserir um ou vários livros

Podemos inserir um ou vários livros, ao adicionarmos um body com os dados a ser inseridos ao request, através do endpoint `/books`. Temos de inserir sempre um título, senão vamos receber um erro



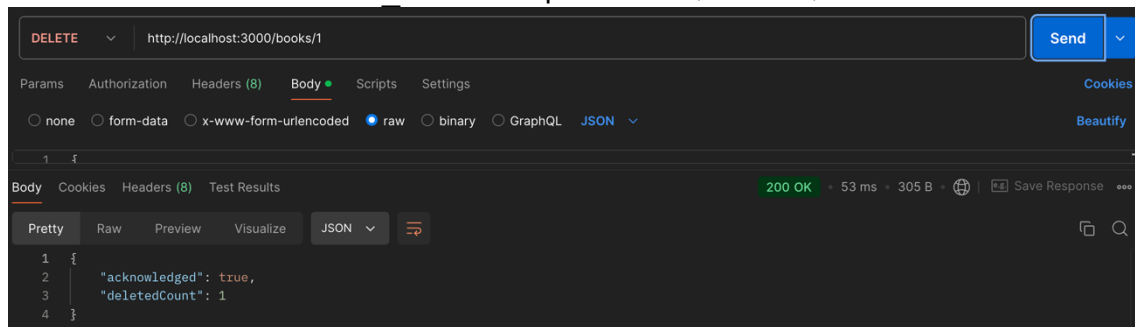
Modificar um livro

Podemos modificar um livro `_id` como parâmetro `/books/1`, ao adicionarmos no body os campos que pretendemos alterar



Eliminar um livro

Podemos eliminar um livro `_id` como parâmetro `/books/1`



User

User schema

Chave	Tipo	Descrição
_id	int	O id do utilizador.
first_name	string	O primeiro nome do utilizador.
last_name	string	O último nome do utilizador.
year_of_birth	int	O ano de nascimento do utilizador.
job	string	O emprego do utilizador.
reviews	array	Lista de reviews do utilizador.

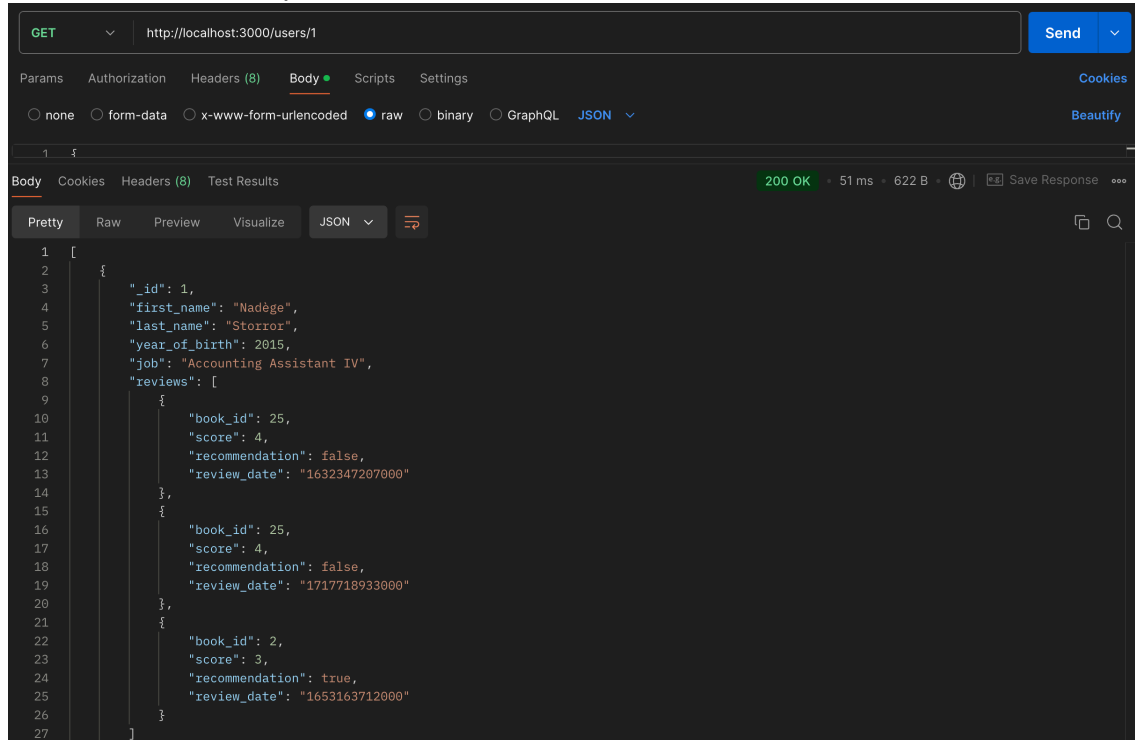
Obter todos os utilizadores

É possível obter todos os utilizadores a partir do endpoint `/users`

```
GET http://localhost:3000/users
200 OK · 165 ms · 8.73 KB
{
  "info": {
    "count": 1000,
    "pages": 50,
    "next": "http://localhost:3000/users/?page=2",
    "prev": null
  },
  "results": [
    {
      "_id": 1,
      "first_name": "Nadège",
      "last_name": "Storror",
      "year_of_birth": 2015,
      "job": "Accounting Assistant IV",
      "reviews": [
        {
          "book_id": 2,
          "score": 3,
          "recommendation": true,
          "review_date": "1653163712000"
        },
        {
          "book_id": 25,
          "score": 4,
          "recommendation": false,
          "review_date": "1717718933000"
        },
        {
          "book_id": 25,
          "score": 4,
          "recommendation": false,
          "review_date": "1632347207000"
        },
        {
          "book_id": 33,
          "score": 2,
          "recommendation": false,
          "review_date": "1663746770000"
        }
      ]
    },
    {
      "book_id": 13,
      "score": 2,
      "recommendation": false
    }
  ]
}
```

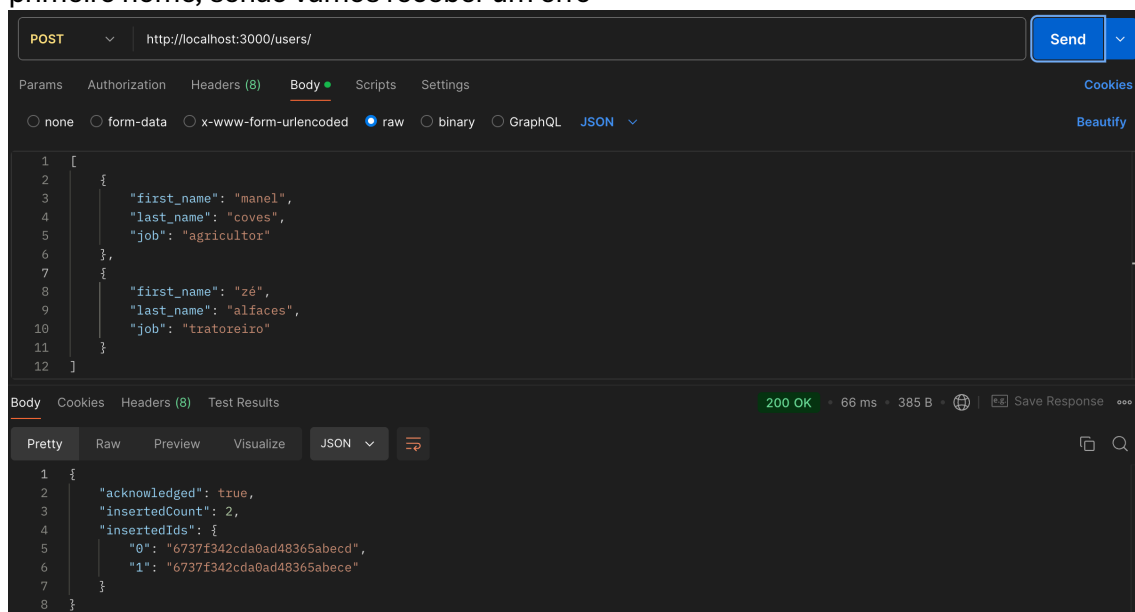
Obter um único utilizador

Podemos obter um único utilizador ao adicionar o `_id` como parâmetro: `/users/1`. Vamos receber as informações desse utilizador com as top 3 reviews do utilizador, ordenadas pelo score



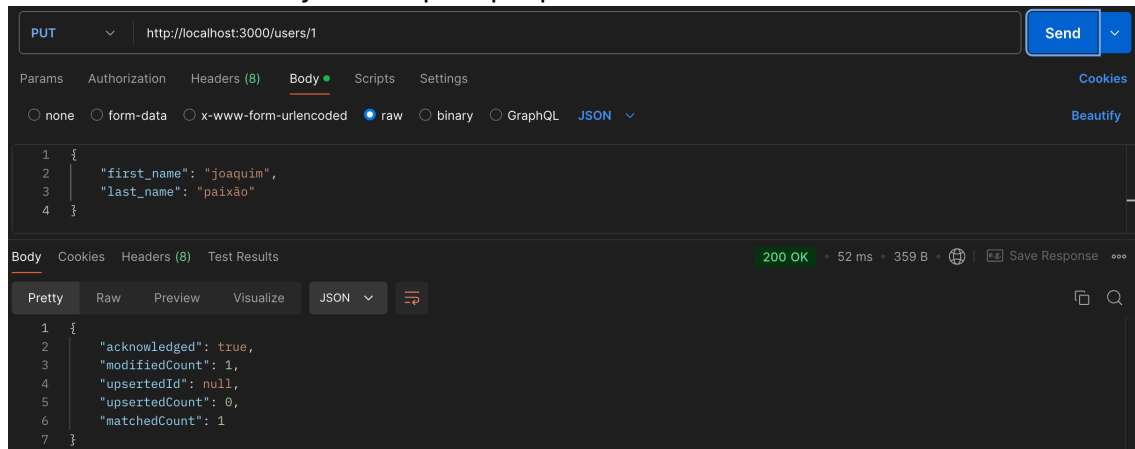
Inserir um ou vários utilizadores

Podemos inserir um ou vários utilizadores, ao adicionarmos um body com os dados a ser inseridos ao request, através do endpoint `/users`. Temos de inserir sempre um primeiro nome, senão vamos receber um erro



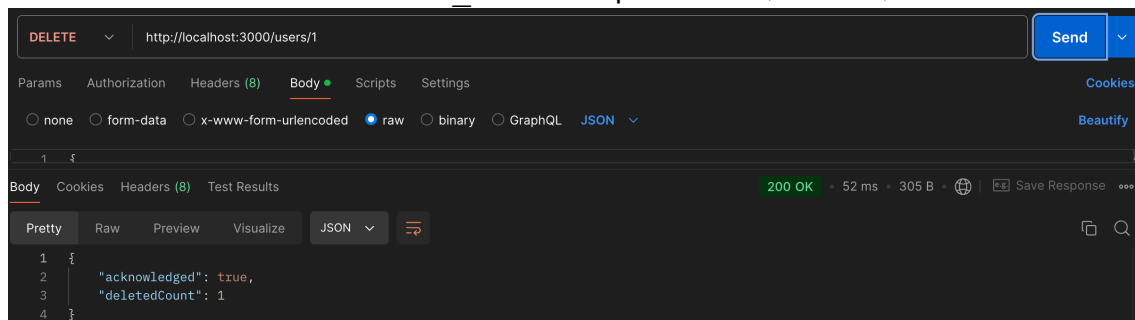
Modificar um utilizador

Podemos modificar um utilizador `_id` como parâmetro: `/users/1`, ao adicionarmos no body os campos que pretendemos alterar



Eliminar um utilizador

Podemos eliminar um utilizador `_id` como parâmetro `/users/1`



Comment

Comment schema

Chave	Tipo	Descrição
_id	int	O id do comentário.
user_id	int	O id do utilizador que fez o comentário.
book_id	int	O id do livro ao qual o comentário é feito.
comment	string	O comentário em si.
date	string	A data de publicação do comentário.

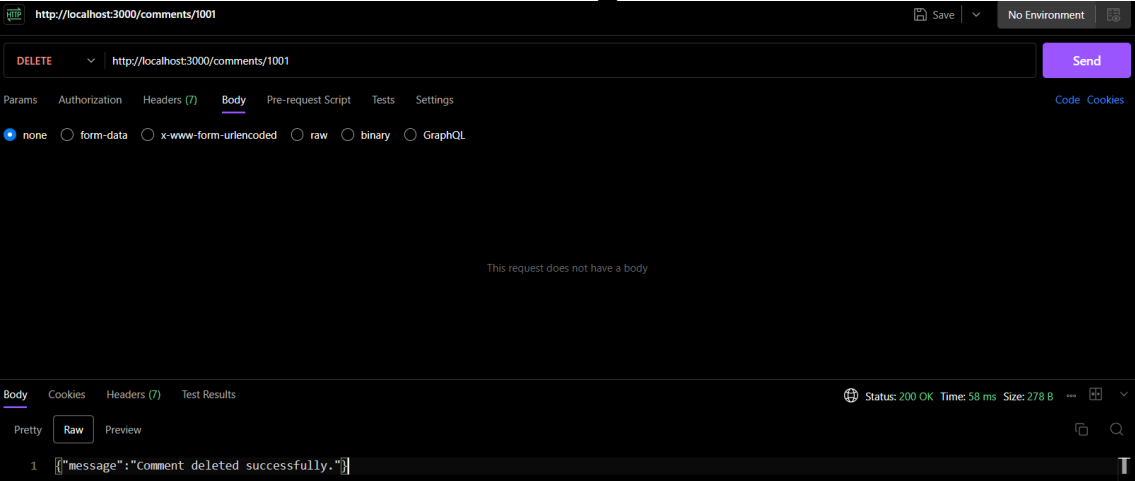
Adicionar um novo comentário a um livro

Podemos inserir um comentário, ao adicionarmos um body com os dados a ser inseridos ao request, através do endpoint `/comments`. Temos de inserir sempre um `book_id`, um `user_id` e um comentário, senão vamos receber um erro



Remover um comentário

Podemos eliminar um comment com o seu `_id` como parâmetro `/comments/1`



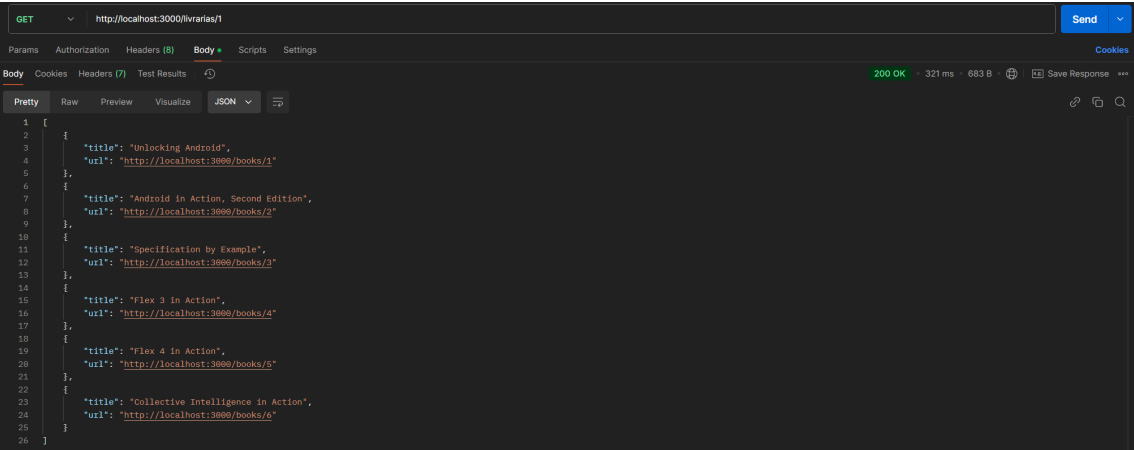
Livraria

Livraria schema

Chave	Tipo	Descrição
_id	int	O id da livreria.
type	string	O tipo de objeto GeoJSON.
geometry	object	Tipo e coordenadas da geometria objeto GeoJSON.
properties	object	Propriedades da livreria.
books	array	Lista de livros que pertencem à livreria.

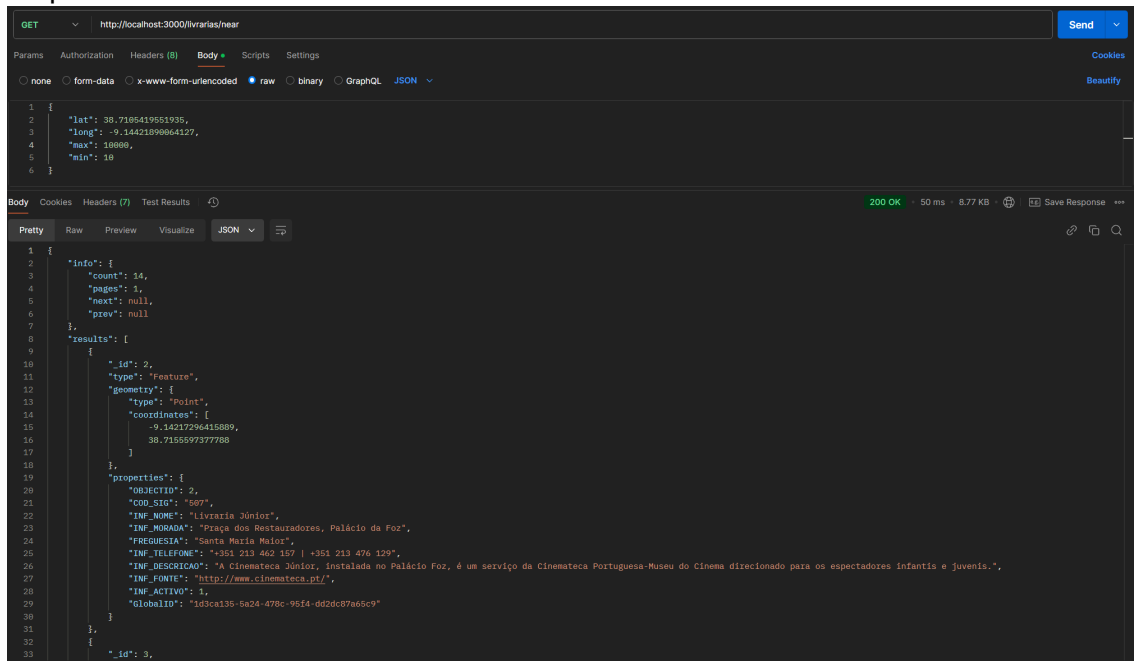
Obter todos os livros de uma livreria

Podemos obter todos os livros de uma livreria, a partir do seu `_id` como parâmetro: `/livrarias/1`



Listar as livrarias perto de uma localização

Podemos obter todas as livrarias perto de uma localização pelo endpoint `/livrarias/near`. No body da mensagem é necessário enviar em formato json as variáveis `lat` (latitude) e `long` (longitude). Também é possível enviar as variáveis `max` para especificar a distância máxima e `min` para especificar a distância mínima ao ponto



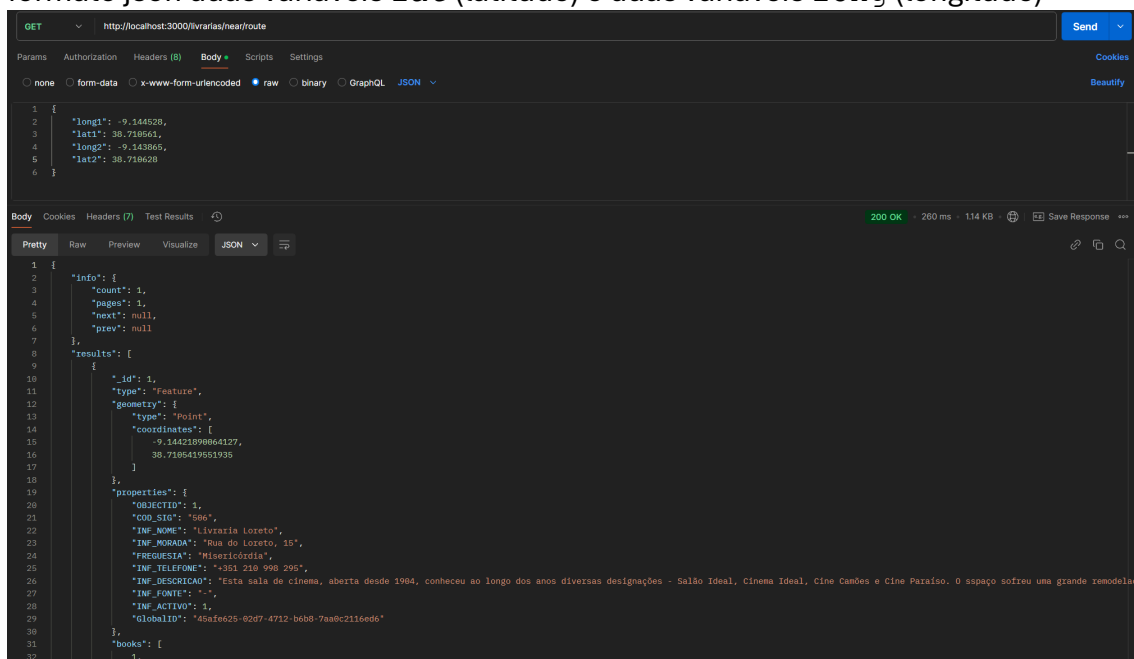
```
GET http://localhost:3000/livrarias/near

{
  "lat": 38.7186419531935,
  "long": -9.14421898864127,
  "max": 10000,
  "min": 10
}
```

```
{
  "info": {
    "count": 14,
    "pages": 1,
    "next": null,
    "prev": null
  },
  "results": [
    {
      "_id": 2,
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -9.14217286415889,
          38.7155597377788
        ]
      },
      "properties": {
        "OBJECTID": 2,
        "COD_SIG": "509",
        "INF_NOME": "Livraria Júnior",
        "INF_MORADA": "Praça dos Restauradores, Palácio da Foz",
        "FREQUENCIA": "Sexta-feira",
        "INF_TELEFONO": "+351 213 462 157 | +351 213 476 129",
        "INF_DESCRICAO": "A Cinemateca Júnior, instalada no Palácio Foz, é um serviço da Cinemateca Portuguesa-Museu do Cinema direcionado para os espectadores infantis e juvenis.",
        "INF_FONTES": "http://www.cinemateca.pt/",
        "INF_ACTIVADO": 1,
        "GlobalID": "1d3ca135-5a24-478c-9514-d82dc87a65c9"
      }
    }
  ]
}
```

Listar as livrarias perto de uma rota

Podemos obter todas as livrarias perto de uma rota pelo endpoint `/livrarias/near/route`. No body da mensagem é necessário enviar em formato json duas variáveis `lat` (latitude) e duas variáveis `long` (longitude)



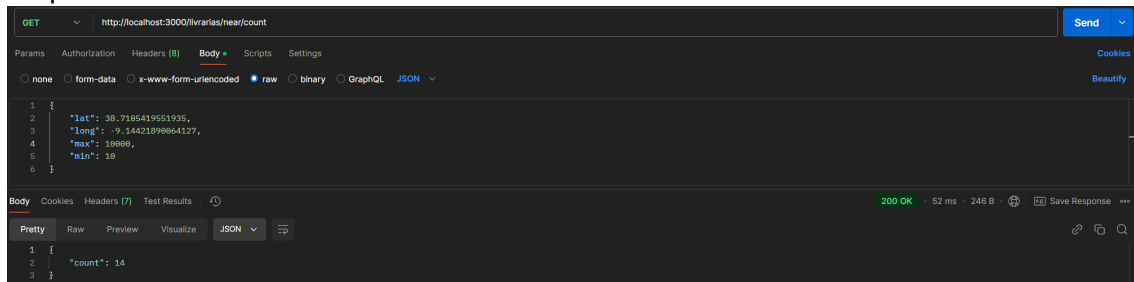
```
GET http://localhost:3000/livrarias/near/route

{
  "long1": -9.1446828,
  "lat1": 38.718641,
  "long2": -9.143865,
  "lat2": 38.718628
}
```

```
{
  "info": {
    "count": 1,
    "pages": 1,
    "next": null,
    "prev": null
  },
  "results": [
    {
      "_id": 1,
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -9.14421898864127,
          38.7186419531935
        ]
      },
      "properties": {
        "OBJECTID": 1,
        "COD_SIG": "586",
        "INF_NOME": "Livraria Loreto",
        "INF_MORADA": "Rua do Loreto, 15",
        "FREQUENCIA": "Miscelânea",
        "INF_TELEFONO": "+351 210 998 295",
        "INF_DESCRICAO": "Esta sala de cinema, aberta desde 1984, conheceu ao longo dos anos diversas designações - Salão Ideal, Cinema Ideal, Cine Camões e Cine Paraíso. O espaço sofreu uma grande remodelação",
        "INF_FONTES": "-",
        "INF_ACTIVADO": 1,
        "GlobalID": "45afe625-0287-4712-b6b8-7a8dc2116ed6"
      }
    }
  ],
  "books": [
    1
  ]
}
```

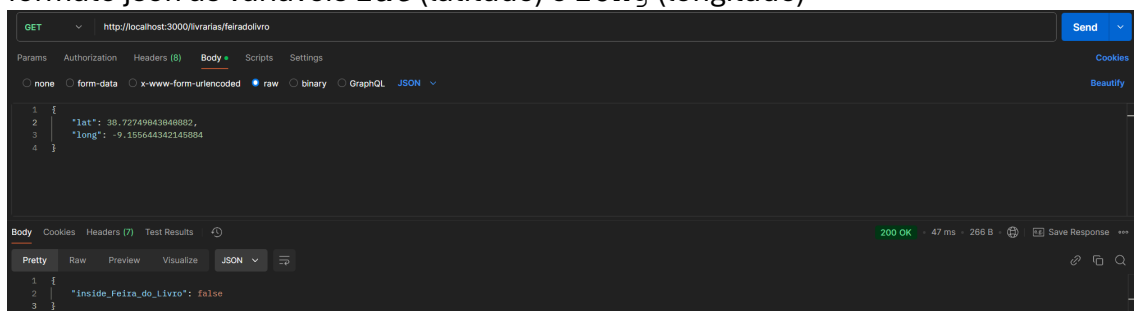
Número de livrarias perto de uma localização

Podemos obter o número de livrarias perto de uma localização pelo endpoint `/livrarias/count`. No body da mensagem é necessário enviar em formato json as variáveis `lat` (latitude) e `long` (longitude). Também é possível enviar as variáveis `max` para especificar a distância máxima e `min` para especificar a distância mínima ao ponto



Verificar se um utilizador está dentro da feira do livro

Podemos verificar se um utilizador está dentro da feira do livro pelo endpoint `/livrarias/feiradolivro`. No body da mensagem é necessário enviar em formato json as variáveis `lat` (latitude) e `long` (longitude)



Adicionar livros a uma livraria

Podemos adicionar livros a uma livraria, a partir do seu `_id` como parâmetro: `/livraria/2`. No body da mensagem deve ir uma array `books` em formato json que contém os `_id` dos livros que pretende adicionar à livraria

