

Sebo Online S.A.

Essa documentação é destinada a desenvolvedores que desejam se integrar à API do Sebo Online S.A. Nela, é possível cadastrar usuários, livros e suas transações.

Início

O trabalho e seus arquivos estão disponíveis no seguinte repositório do GitHub:

<https://github.com/mcavalle/api-sebo-online>

Os demais arquivos necessários com informações sensíveis estão dentro da pasta compactada enviada. Nela, é possível baixar o arquivo .env, com os dados necessários para acesso ao banco.

Para iniciar o programa, no terminal deve ser utilizado o comando “npm start”. É preciso aparecer a mensagem de “Conectamos ao MongoDB”, informando que a conexão foi bem sucedida:

```
> sebo-online-api@1.0.2 start
> nodemon ./index.js localhost 3000

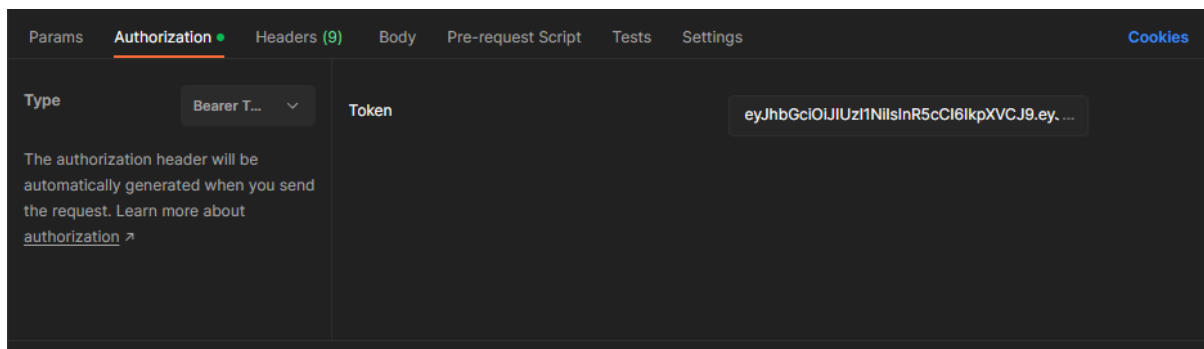
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node ./index.js localhost 3000`
Conectamos ao MongoDB
```

Autorização

Algumas rotas são privadas e exclusivas para o administrador. Para acessar essas rotas, será necessário realizar o login como usuário administrador e pegar o Token que é fornecido, conforme exemplo abaixo:

```
1  {
2    "message": "Autenticação realizada com sucesso!",
3    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY1MjAzMGVlOGE4ZmY0NDY2Y2Y5ODh0SIsIm1hdCI6MTY5Njc5NDUwNn0.e3I1JWxpxQMteS4w19I-JucZ5LkU4Cu_HEFifomcLers"
4  }
```

O tipo de autorização definido deverá ser “Bearer Token” e ter seu valor informado para ter acesso às rotas privadas:



Endpoints

Cadastrar usuários

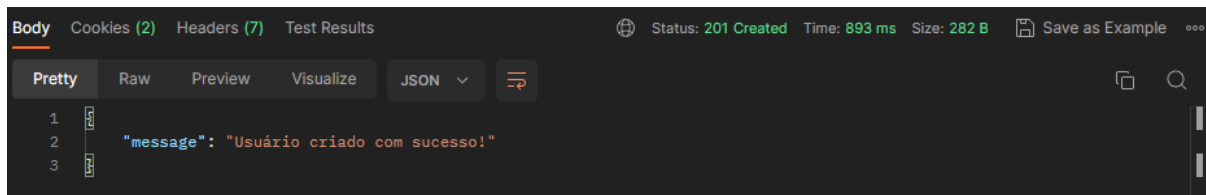
É possível cadastrar um novo usuário no sistema utilizando o método POST HTTP, pela seguinte URL: `http://localhost:3000/usuario/cadastro`

No Body da requisição, as informações do usuário devem ser fornecidas, conforme o exemplo abaixo.

```
{
  "name": "Jane Doe",
  "email": "jane@sebo.com.br",
  "password": "SPFC2023",
  "confirmPassword": "SPFC2023",
  "active": true,
  "type": "administrador",
  "date": "2022-04-05",
  "area": "Teste"
}
```

As informações de nome, e-mail, senha e se o usuário está ativo ou não são obrigatórias para todos os usuários cadastrados. Já os campos de date (data de início) e area (área de especialização) só podem ser preenchidos por usuários do tipo “administrador”.

Se a solicitação for bem sucedida, será retornado o status 201 e uma mensagem de confirmação de cadastro:



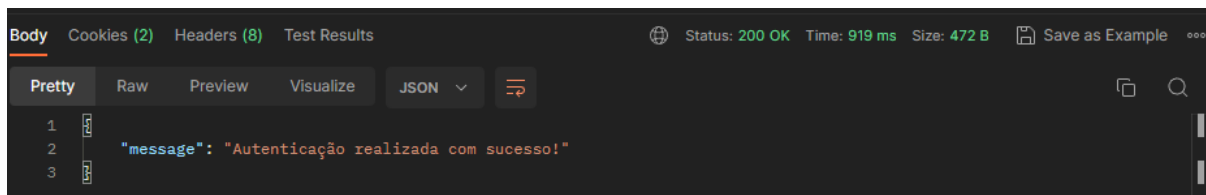
Login de usuário

O usuário pode realizar o seu login no sistema através do método POST HTTP, pela seguinte URL: <http://localhost:3000/usuario/login>

No Body da requisição, as informações de e-mail e senha devem ser fornecidas, conforme o exemplo abaixo.

```
{  
  "email": "jane@sebo.com.br",  
  "password": "SPFC2023"  
}
```

Se o login for bem sucedido, será retornado o status 200 OK e uma mensagem de login realizado com sucesso:

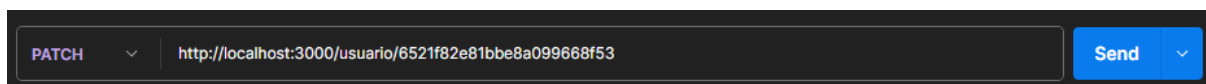


Para o login, devem ser preenchidos um e-mail já cadastrado e uma senha válida.

Alterar informações do usuário

É possível realizar alterações no usuário através do método PATCH HTTP, pela seguinte URL: <http://localhost:3000/usuario/{ID}>

Para esse método, precisamos fornecer o ID do usuário na URL, conforme o exemplo abaixo:

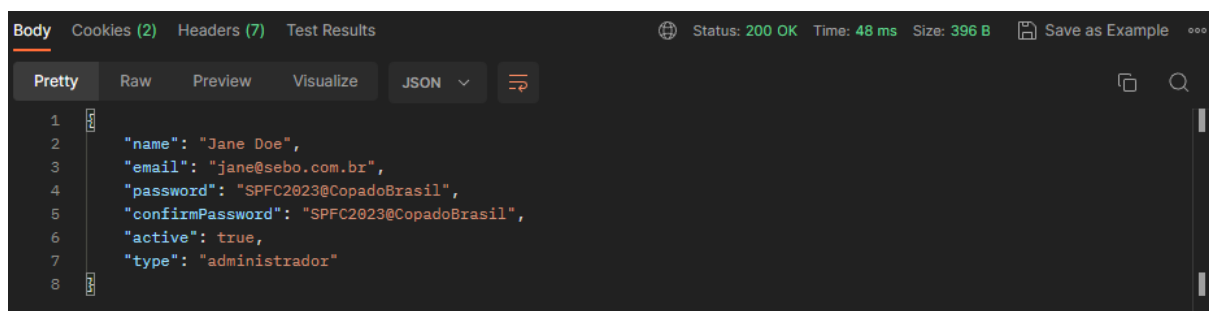


No Body, as informações do usuário devem ser fornecidas, alterando o campo desejado:

```
{  
  "name": "Jane Doe",
```

```
"email": "jane@sebo.com.br",
"password": "SPFC2023@CopadoBrasil",
"confirmPassword": "SPFC2023@CopadoBrasil",
"active": true,
"type": "administrador",
"date": "2022-04-05",
"area": "Teste"
}
```

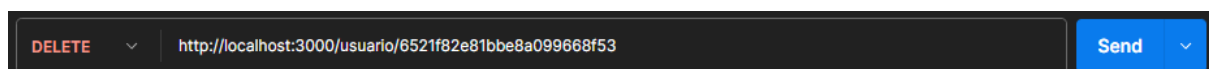
Se a atualização for bem sucedida, será retornado um status 200 OK, com os dados atualizados



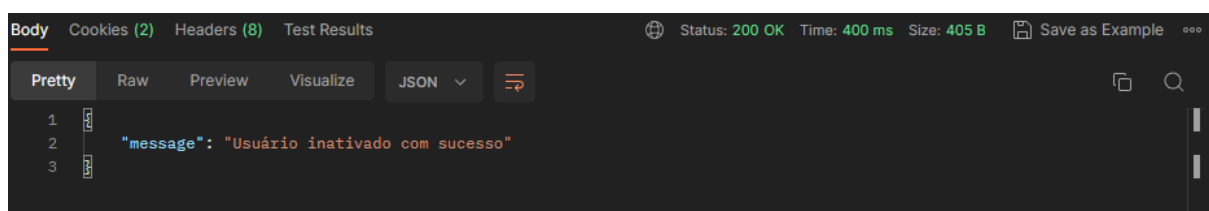
Excluir um usuário

Para essa API, os cadastros não podem ser totalmente excluídos do banco. Apenas é possível inativar os cadastros já existentes. Isso é possível através do método DELETE HTTP, pela seguinte URL: `http://localhost:3000/usuario/{ID}`

O ID do usuário a ser excluído deve ser informado na URL, conforme o exemplo:



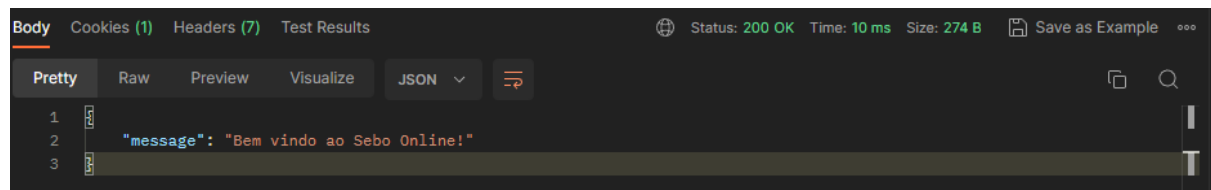
Se a requisição for bem sucedida, será retornado um status 200 OK e uma mensagem de sucesso:



Logout

Para fazer o logout, basta acessar utilizar o método GET através da seguinte URL: <http://localhost:3000/usuario/logout>

Se a requisição for bem sucedida, será retornada um status 200 OK e o usuário será retornado à página inicial:



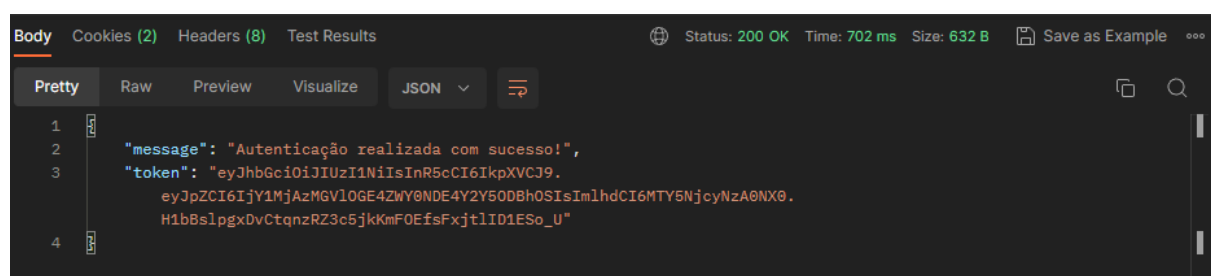
Login de administrador

Usuários administradores podem fazer login por uma rota diferente, que irá retornar um token para acesso à rotas privadas. Para isso é preciso utilizar o método POST através da seguinte URL: <http://localhost:3000/admin/login>

No Body da requisição, as informações de e-mail e senha devem ser fornecidas, conforme o exemplo abaixo.

```
{  
  "email": "jane@sebo.com.br",  
  "password": "SPFC2023"  
}
```

Se o login for bem sucedido, será retornado o status 200 OK e uma mensagem de login realizado com sucesso e o token:

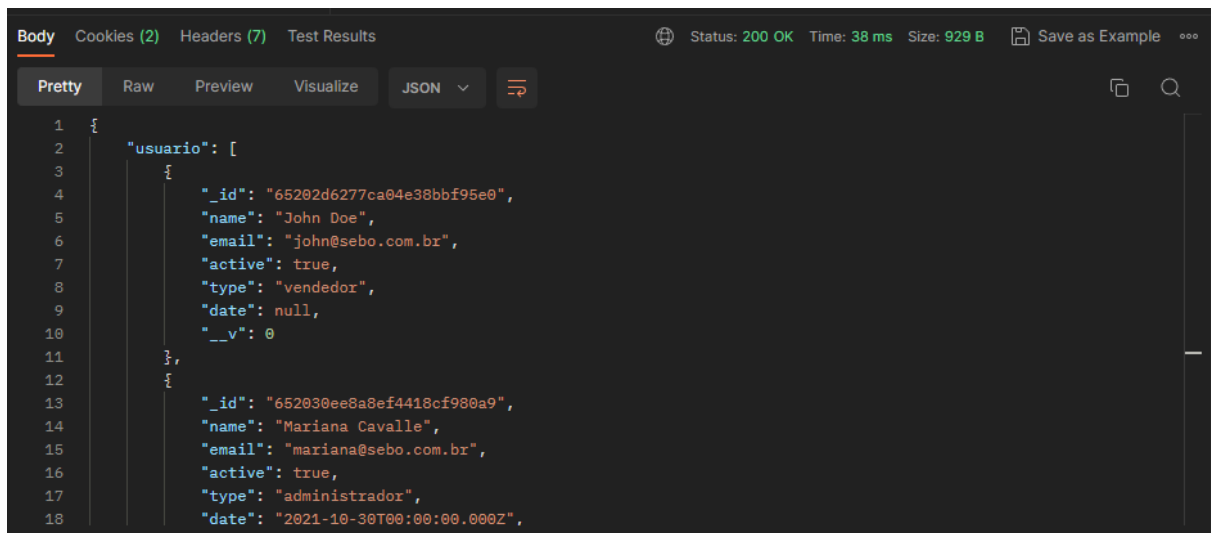


Listar usuários

É possível listar todos os usuários do sistema através do método GET, pela seguinte URL: <http://localhost:3000/admin/usuarios>

Essa é uma rota privada, portanto o token deve ser informado conforme o tópico “Autenticação”.

Se a requisição for bem sucedida, será retornado o status 200 OK e a lista dos usuários do sistema:



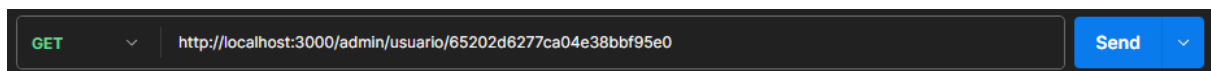
```
Body Cookies (2) Headers (7) Test Results
Status: 200 OK Time: 38 ms Size: 929 B Save as Example

Pretty Raw Preview Visualize JSON

1 {
2   "usuario": [
3     {
4       "_id": "65202d6277ca04e38bbf95e0",
5       "name": "John Doe",
6       "email": "john@sebo.com.br",
7       "active": true,
8       "type": "vendedor",
9       "date": null,
10      "__v": 0
11    },
12  ],
13  {
14    "_id": "652030ee8a8ef4418cf980a9",
15    "name": "Mariana Cavalle",
16    "email": "mariana@sebo.com.br",
17    "active": true,
18    "type": "administrador",
19    "date": "2021-10-30T00:00:00.000Z",
```

Buscar usuário por ID

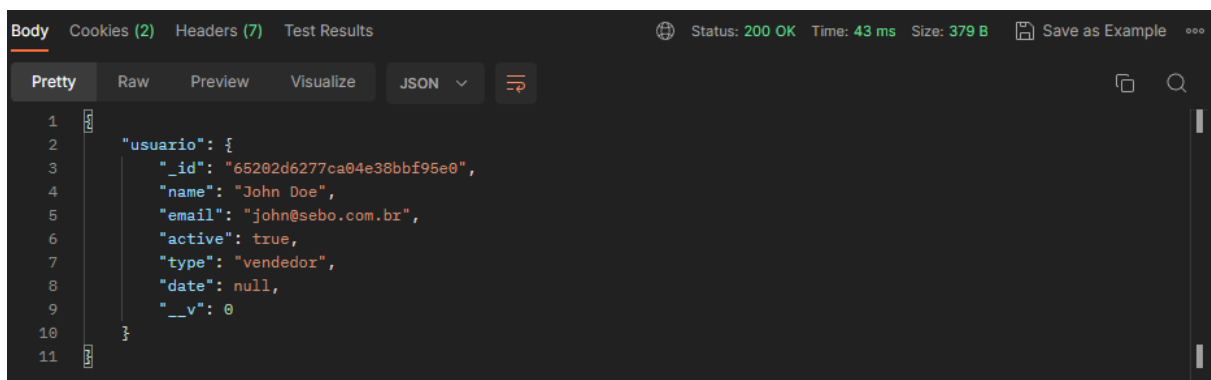
Para buscar um usuário em específico, a busca pode ser feita através do ID do usuário, fornecido na URL. Para isso, deve ser usado o método GET, pela seguinte URL: `http://localhost:3000/admin/usuario/{ID}`



```
GET http://localhost:3000/admin/usuario/65202d6277ca04e38bbf95e0 Send
```

Essa é uma rota privada, portanto o token deve ser informado conforme o tópico “Autenticação”.

Se a requisição for bem sucedida, será retornado o status 200 OK e as informações do usuário em questão:



```
Body Cookies (2) Headers (7) Test Results
Status: 200 OK Time: 43 ms Size: 379 B Save as Example

Pretty Raw Preview Visualize JSON

1 {
2   "usuario": {
3     "_id": "65202d6277ca04e38bbf95e0",
4     "name": "John Doe",
5     "email": "john@sebo.com.br",
6     "active": true,
7     "type": "vendedor",
8     "date": null,
9     "__v": 0
10  }
11 }
```

Relatórios

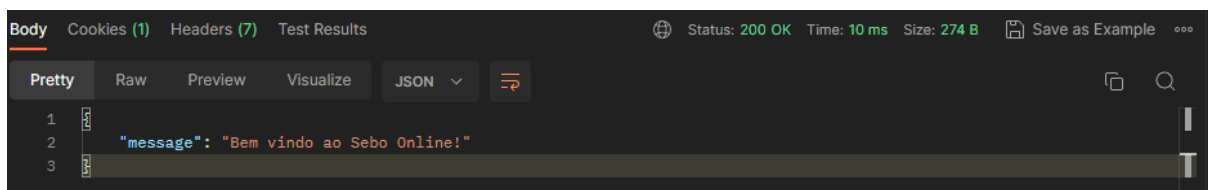
É possível gerar um relatório das transações realizadas. Para isso, deve ser usado o método GET, pela seguinte URL: <http://localhost:3000/admin/relatorios>

Essa é uma rota privada, portanto o token deve ser informado conforme o tópico “Autenticação”.

Logout administrador

Para fazer o logout, basta acessar utilizar o método GET através da seguinte URL: <http://localhost:3000/admin/logout>

Se a requisição for bem sucedida, será retornada um status 200 OK e o usuário será retornado à página inicial:



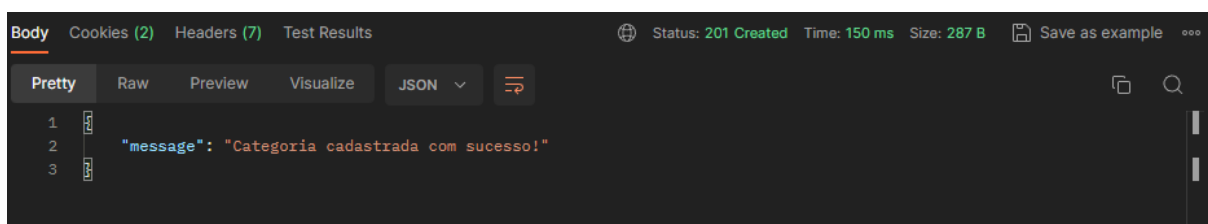
Cadastrar categorias

É possível cadastrar uma nova categoria no sistema utilizando o método POST HTTP, pela seguinte URL: <http://localhost:3000/categoria/cadastro>

No Body da requisição, as informações da categoria devem ser fornecidas, conforme o exemplo abaixo.

```
{  "name": "Fantasia",  "description": "Fantasia é um gênero da ficção em que se usa geralmente fenômenos sobrenaturais, mágicos e outros como um elemento primário do enredo, tema ou configuração.",  "active": true}
```

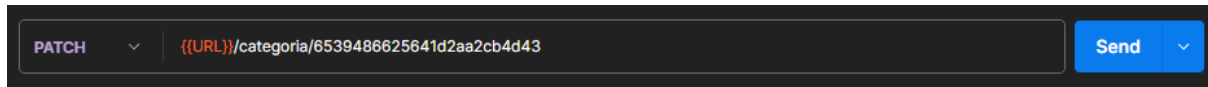
Se a solicitação for bem sucedida, será retornado o status 201 e uma mensagem de confirmação de cadastro:



Alterar informações da categoria

É possível realizar alterações no usuário através do método PATCH HTTP, pela seguinte URL: `http://localhost:3000/categoria/{ID}`

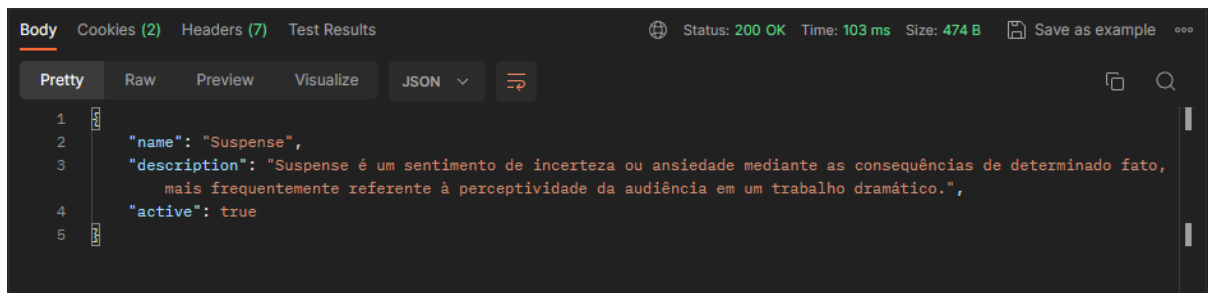
No final da URL, é preciso informar qual o ID da categoria a ser alterada, conforme o exemplo abaixo:



No Body, as informações da categoria devem ser fornecidas, alterando o campo desejado:

```
{
  "name": "Suspense",
  "description": "Suspense é um sentimento de incerteza ou ansiedade mediante as
consequências de determinado fato, mais frequentemente referente à
perceptividade da audiência em um trabalho dramático.",
  "active": true
}
```

Se a solicitação for bem sucedida, será retornado o status 200 e o cadastro atualizado:



Listar categorias

É possível listar todas as categorias pelo método GET HTTP, através da seguinte URL: `http://localhost:3000/categoria`

Se a solicitação for bem sucedida, será retornado status 200 e a listagem das categorias:


```
1 {
2   "categoria": [
3     {
4       "_id": "6539486625641d2aa2cb4d43",
5       "name": "Suspense",
6       "description": "Suspense é um sentimento de incerteza ou ansiedade mediante as consequências de determinado fato, mais frequentemente referente à perceptividade da audiência em um trabalho dramático.",
7       "__v": 0,
8       "active": true
9     },
10    {
11      "_id": "653aeeef83da4f622708f1b58",
12      "name": "Fantasia",
13      "description": "Fantasia é um gênero da ficção em que se usa geralmente fenômenos sobrenaturais, mágicos e outros como um elemento primário do enredo, tema ou configuração.",
14      "active": true,
15      "v": 0
16    }
17  ]
18 }
```

Buscar categoria por ID

É possível buscar uma categoria específica pelo seu ID, utilizando o método GET HTTP, através da seguinte URL: `http://localhost:3000/categoria/{ID}`

Basta informar o ID da categoria conforme o exemplo abaixo:

```
GET {{URL}}/categoria/6539486625641d2aa2cb4d43
```

Se a solicitação for bem sucedida, será retornado o status 200 e a categoria especificada:

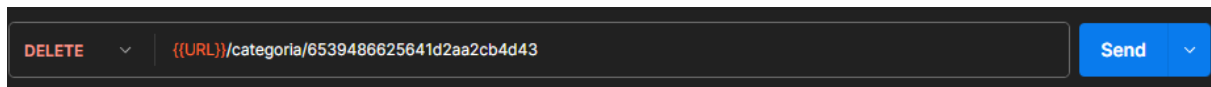
```
1 {
2   "categoria": {
3     "_id": "6539486625641d2aa2cb4d43",
4     "name": "Suspense",
5     "description": "Suspense é um sentimento de incerteza ou ansiedade mediante as consequências de determinado fato, mais frequentemente referente à perceptividade da audiência em um trabalho dramático.",
6     "__v": 0,
7     "active": true
8   }
9 }
```

Delete de categoria

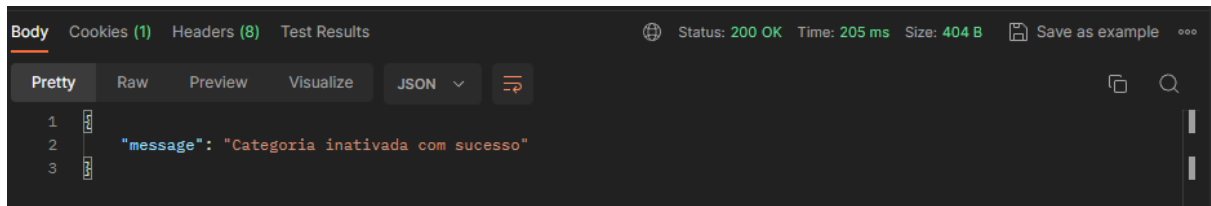
É possível fazer o soft delete de uma categoria específica pelo seu ID, utilizando o método DELETE HTTP, através da seguinte URL:

`http://localhost:3000/categoria/{ID}`

Basta informar o ID da categoria conforme o exemplo abaixo:



Se a solicitação for bem sucedida, será retornado o status 200 e uma mensagem de sucesso:



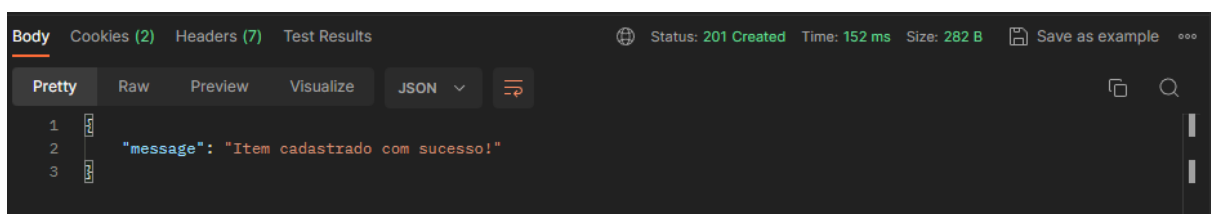
Cadastro de itens

É possível cadastrar um novo item no sistema utilizando o método POST HTTP, pela seguinte URL: <http://localhost:3000/item/cadastro>

No Body da requisição, as informações do item devem ser fornecidas, conforme o exemplo abaixo.

```
{
  "title": "Sobre os Ossos dos Mortos",
  "author": "Rick Riordan",
  "type": "Livro",
  "categoryId": "6539486625641d2aa2cb4d43",
  "price": 40,
  "description": "A aclamada autora mistura thriller e humor nesta reflexão sobre a condição humana e a natureza. Vencedora do prêmio Nobel de literatura",
  "active": true,
  "editionDate": "2019-10-30",
  "sellerId": "65202d6277ca04e38bbf95e0"
}
```

Se a solicitação for bem sucedida, será retornado o status 201 e uma mensagem de confirmação de cadastro:



Alterar item

É possível realizar alterações no usuário através do método PATCH HTTP, pela seguinte URL: `http://localhost:3000/item/{ID}`

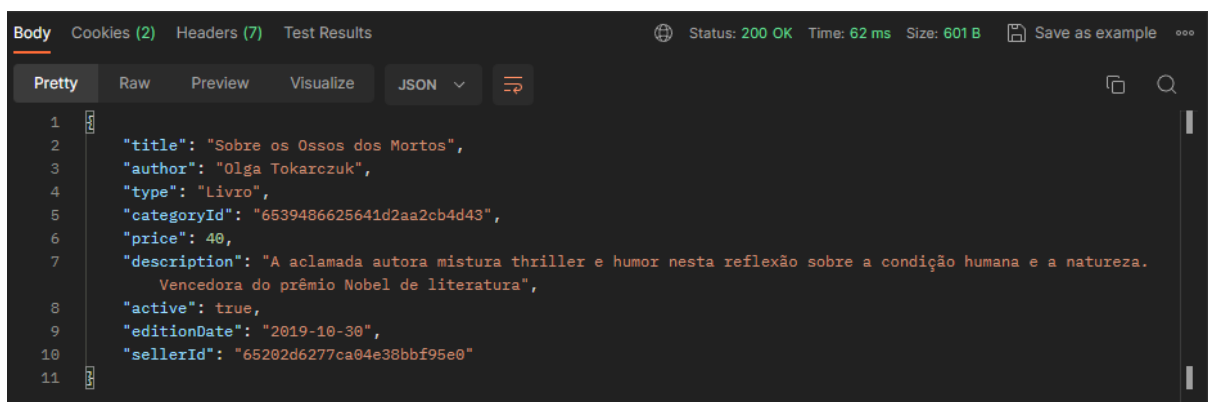
No final da URL, é preciso informar qual o ID do item a ser alterado, conforme o exemplo abaixo:



No Body, as informações do item devem ser fornecidas, alterando o campo desejado:

```
{
  "title": "Sobre os Ossos dos Mortos",
  "author": "Olga Tokarczuk",
  "type": "Livro",
  "categoryId": "6539486625641d2aa2cb4d43",
  "price": 40,
  "description": "A aclamada autora mistura thriller e humor nesta reflexão sobre a condição humana e a natureza. Vencedora do prêmio Nobel de literatura",
  "active": true,
  "editionDate": "2019-10-30",
  "sellerId": "65202d6277ca04e38bbf95e0"
}
```

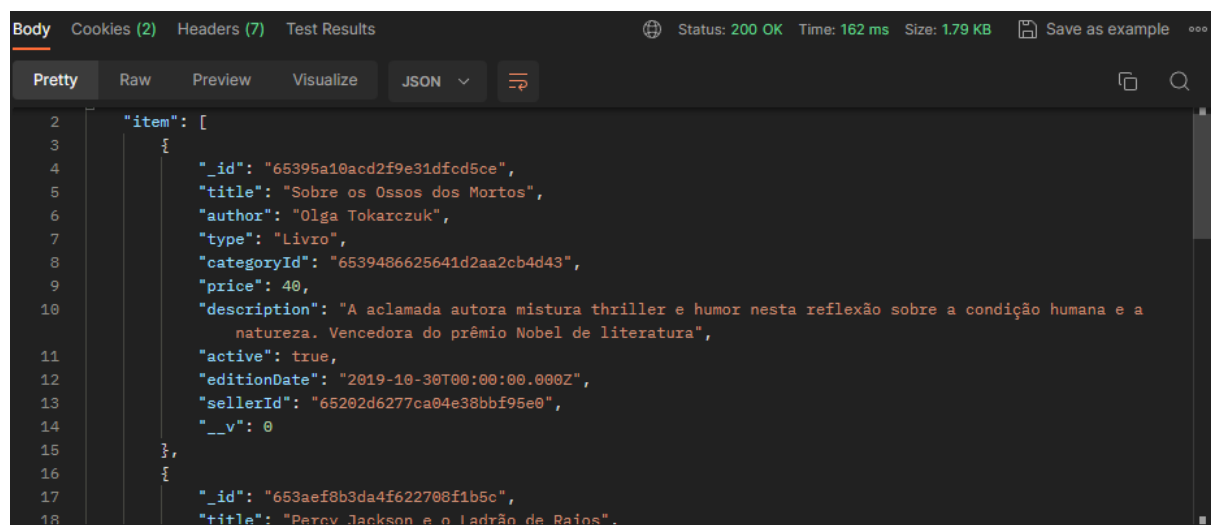
Se a solicitação for bem sucedida, será retornado o status 200 e o cadastro atualizado:



Listar itens

É possível listar todos os itens pelo método GET HTTP, através da seguinte URL: `http://localhost:3000/item`

Se a solicitação for bem sucedida, será retornado status 200 e a listagem dos itens:

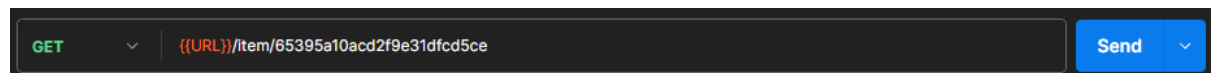


```
Body Cookies (2) Headers (7) Test Results Status: 200 OK Time: 162 ms Size: 1.79 KB Save as example
Pretty Raw Preview Visualize JSON
{
  "item": [
    {
      "_id": "65395a10acd2f9e31dfcd5ce",
      "title": "Sobre os Ossos dos Mortos",
      "author": "Olga Tokarczuk",
      "type": "Livro",
      "categoryId": "6539486625641d2aa2cb4d43",
      "price": 40,
      "description": "A aclamada autora mistura thriller e humor nesta reflexão sobre a condição humana e a natureza. Vencedora do prêmio Nobel de literatura",
      "active": true,
      "editionDate": "2019-10-30T00:00:00.000Z",
      "sellerId": "65202d6277ca04e38bbf95e0",
      "__v": 0
    },
    {
      "_id": "653aef8b3da4f622708f1b5c",
      "title": "Perceve Jackson e o Ladrão de Raios".
    }
  ]
}
```

Buscar item por ID

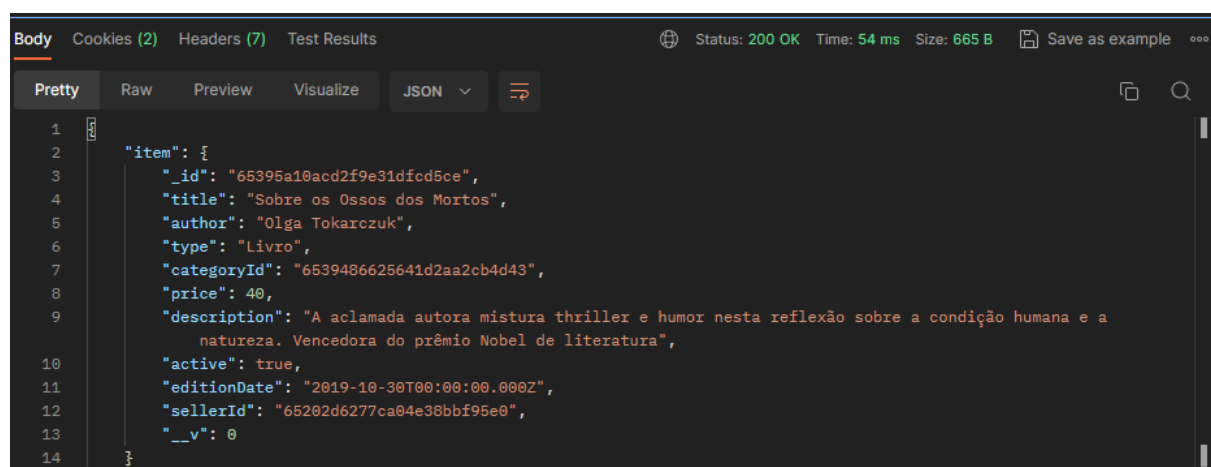
É possível buscar um item específico pelo seu ID, utilizando o método GET HTTP, através da seguinte URL: `http://localhost:3000/item/{ID}`

Basta informar o ID da categoria conforme o exemplo abaixo:



```
GET {{URL}}/item/65395a10acd2f9e31dfcd5ce Send
```

Se a solicitação for bem sucedida, será retornado o status 200 e o item especificado:



```
Body Cookies (2) Headers (7) Test Results Status: 200 OK Time: 54 ms Size: 665 B Save as example
Pretty Raw Preview Visualize JSON
{
  "item": {
    "_id": "65395a10acd2f9e31dfcd5ce",
    "title": "Sobre os Ossos dos Mortos",
    "author": "Olga Tokarczuk",
    "type": "Livro",
    "categoryId": "6539486625641d2aa2cb4d43",
    "price": 40,
    "description": "A aclamada autora mistura thriller e humor nesta reflexão sobre a condição humana e a natureza. Vencedora do prêmio Nobel de literatura",
    "active": true,
    "editionDate": "2019-10-30T00:00:00.000Z",
    "sellerId": "65202d6277ca04e38bbf95e0",
    "__v": 0
  }
}
```

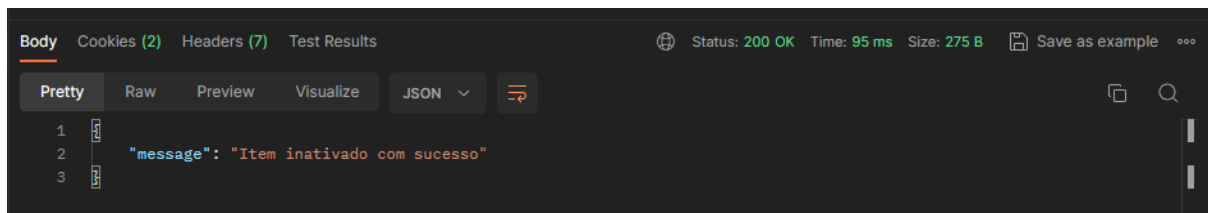
Delete de item

É possível fazer o soft delete de uma categoria específica pelo seu ID, utilizando o método DELETE HTTP, através da seguinte URL: `http://localhost:3000/item/{ID}`

Basta informar o ID do item conforme o exemplo abaixo:



Se a solicitação for bem sucedida, será retornado o status 200 e uma mensagem de sucesso:



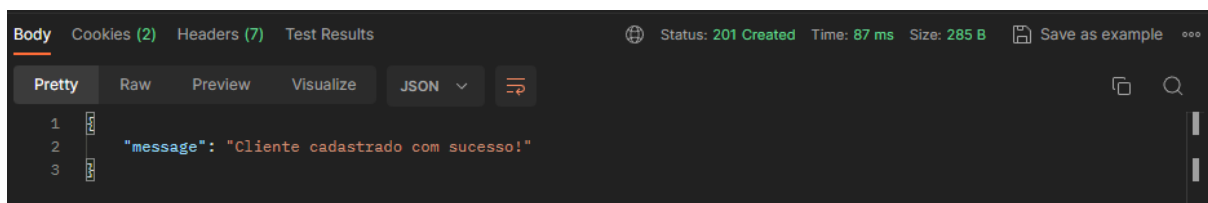
Cadastrar clientes

É possível cadastrar um novo cliente no sistema utilizando o método POST HTTP, pela seguinte URL: `http://localhost:3000/cliente/cadastro`

No Body da requisição, as informações do cliente devem ser fornecidas, conforme o exemplo abaixo.

```
{
  "name": "Pam Beesly",
  "email": "pam@dundermifflin.com.br",
  "active": true
}
```

Se a solicitação for bem sucedida, será retornado o status 201 e uma mensagem de confirmação de cadastro:



Alterar cliente

É possível realizar alterações no cliente através do método PATCH HTTP, pela seguinte URL: `http://localhost:3000/cliente/{ID}`

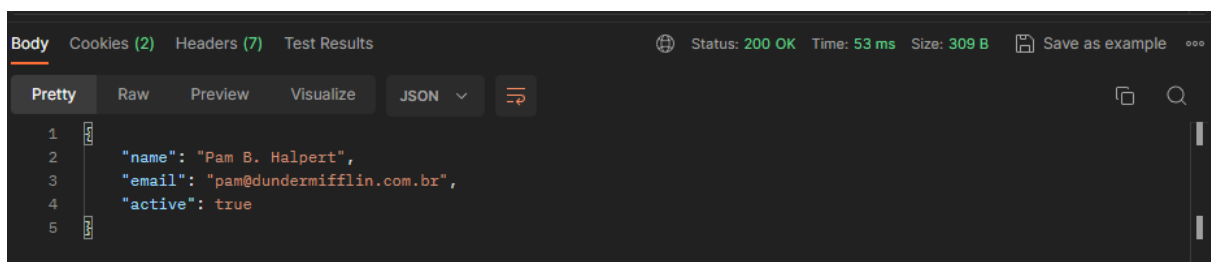
No final da URL, é preciso informar qual o ID do cliente a ser alterado, conforme o exemplo abaixo:



No Body, as informações da categoria devem ser fornecidas, alterando o campo desejado:

```
{
  "name": "Pam B. Halpert",
  "email": "pam@dundermifflin.com.br",
  "active": true
}
```

Se a solicitação for bem sucedida, será retornado o status 200 e o cadastro atualizado:



Listar clientes

É possível listar todos os clientes pelo método GET HTTP, através da seguinte URL:
<http://localhost:3000/cliente>

Se a solicitação for bem sucedida, será retornado status 200 e a listagem dos clientes:

```
1 {
2   "cliente": [
3     {
4       "_id": "6539bf3fefa3c92bea2c248a",
5       "name": "Cliente Consumidor",
6       "email": "teste@teste.com.br",
7       "active": true,
8       "__v": 0
9     },
10    {
11      "_id": "653af0323da4f622708f1b63",
12      "name": "Jim Halpert",
13      "email": "jim@dundermifflin.com.br",
14      "active": true,
15      "__v": 0
16    },
17    {
18      "_id": "654ff94565cbaba68d678efc",
```

Buscar cliente por ID

É possível buscar um cliente específico pelo seu ID, utilizando o método GET HTTP, através da seguinte URL: `http://localhost:3000/cliente/{ID}`

Basta informar o ID do cliente conforme o exemplo abaixo:

```
GET {{URL}}/cliente/6539bf3fefa3c92bea2c248a Send
```

Se a solicitação for bem sucedida, será retornado o status 200 e o cliente especificado:

```
1 {
2   "cliente": {
3     "_id": "6539bf3fefa3c92bea2c248a",
4     "name": "Cliente Consumidor",
5     "email": "teste@teste.com.br",
6     "active": true,
7     "__v": 0
8   }
9 }
```

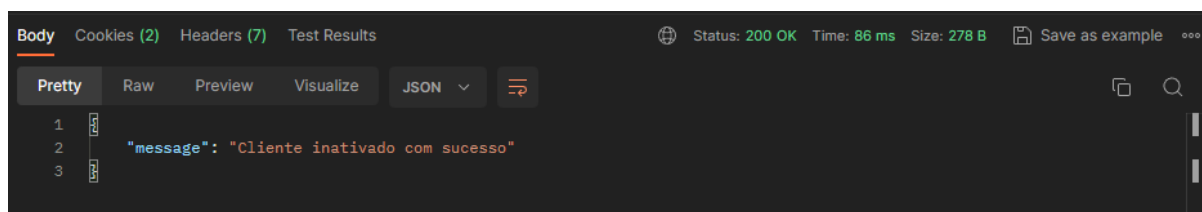
Delete de cliente

É possível fazer o soft delete de um cliente específico pelo seu ID, utilizando o método DELETE HTTP, através da seguinte URL: `http://localhost:3000/cliente/{ID}`

Basta informar o ID do cliente conforme o exemplo abaixo:

```
DELETE {{URL}}/cliente/6539bf3fefa3c92bea2c248a Send
```

Se a solicitação for bem sucedida, será retornado o status 200 e uma mensagem de sucesso:



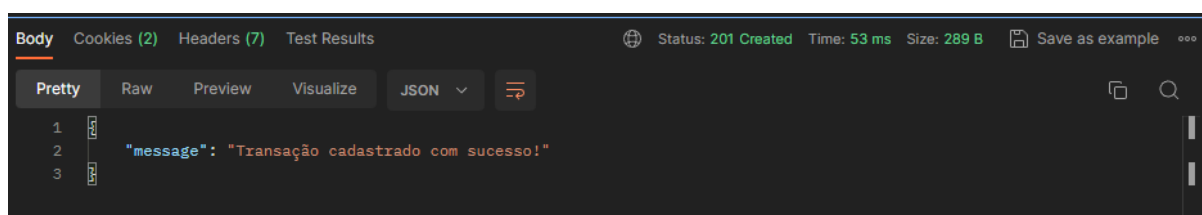
Cadastrar transação

É possível incluir transações no sistema utilizando o método POST HTTP, pela seguinte URL: <http://localhost:3000/transacao/cadastro>

No Body da requisição, as informações da transação devem ser fornecidas, conforme o exemplo abaixo.

```
{  "clientId": "6539bf3fefa3c92bea2c248a",  "sellerId": "652030ee8a8ef4418cf980a9",  "itemId": "653af0033da4f622708f1b5f",  "price": 70,  "transactionDate": "2021-10-26"}
```

Se a solicitação for bem sucedida, será retornado o status 201 e uma mensagem de confirmação de cadastro:



Listar transações

É possível listar todas as transações feitas no sistema pelo método GET HTTP, através da seguinte URL: <http://localhost:3000/transacao>

Se a solicitação for bem sucedida, será retornado status 200 e a listagem das transações:


```
Body Cookies (2) Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1
2  "transacao": [
3    {
4      "_id": "653af12a3da4f622708f1b67",
5      "clientId": "6539bf3fefa3c92bea2c248a",
6      "sellerId": "652030ee8a8ef4418cf980a9",
7      "itemId": "653af0033da4f622708f1b5f",
8      "price": 55,
9      "transactionDate": "2023-10-26T00:00:00.000Z",
10     "__v": 0
11   },
12   {
13     "_id": "653af1703da4f622708f1b6b",
14     "clientId": "6539bf3fefa3c92bea2c248a",
15     "sellerId": "652030ee8a8ef4418cf980a9",
16     "itemId": "653af0033da4f622708f1b5f",
17     "price": 70,
18     "transactionDate": "2021-10-26T00:00:00.000Z",
```

Buscar transação por vendedor

É possível filtrar transações de um vendedor específico, utilizando o método GET HTTP, através da seguinte URL: `http://localhost:3000/vendedor/{ID}`

Basta informar o ID do vendedor conforme o exemplo abaixo:

```
GET {{URL}}/transacao/vendedor/652030ee8a8ef4418cf980a9 Send
```

Se a solicitação for bem sucedida, será retornado o status 200 e o cliente especificado:

```
Body Cookies (2) Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1
2  "transacao": [
3    {
4      "_id": "653af12a3da4f622708f1b67",
5      "clientId": "6539bf3fefa3c92bea2c248a",
6      "sellerId": "652030ee8a8ef4418cf980a9",
7      "itemId": "653af0033da4f622708f1b5f",
8      "price": 55,
9      "transactionDate": "2023-10-26T00:00:00.000Z",
10     "__v": 0
11   },
12   {
13     "_id": "653af1703da4f622708f1b6b",
14     "clientId": "6539bf3fefa3c92bea2c248a",
15     "sellerId": "652030ee8a8ef4418cf980a9",
```

Otimização de buscas

Para a otimização de buscas, utilizaria algoritmos de QuickSort para ordenar os dados de forma mais eficiente e Elasticsearch para realizar buscas mais eficiente

Referências

Para o desenvolvimento do projeto, foram utilizadas as referências abaixo:

Crie uma API RESTful com Node.js e MongoDB | CRUD com Node, Express e Mongoose. Link para acesso: <https://www.youtube.com/watch?v=K5QaTfE5yIk>

Autenticação com Node.js e MongoDB com JWT - Login e Registro com Node.js. Link para acesso: <https://www.youtube.com/watch?v=qEBoZ8lJR3k>

Build a REST API with Node JS and Express | CRUD API Tutorial. Link para acesso: <https://www.youtube.com/watch?v=l8WPWK9mS5M>

Cookies HTTP em NodeJS. Link para acesso: <https://acervolima.com/cookies-http-em-node-js/>

Cookie. Link para acesso: <https://www.npmjs.com/package/cookie>

QuickSort Algorithm in JavaScript. Link para acesso: <https://www.guru99.com/quicksort-in-javascript.html>

Elasticsearch. Link para acesso: <https://www.elastic.co/guide/en/cloud/current/ec-getting-started-node-js.html>