



Produtos

User Story 1: Listagem de todos os produtos

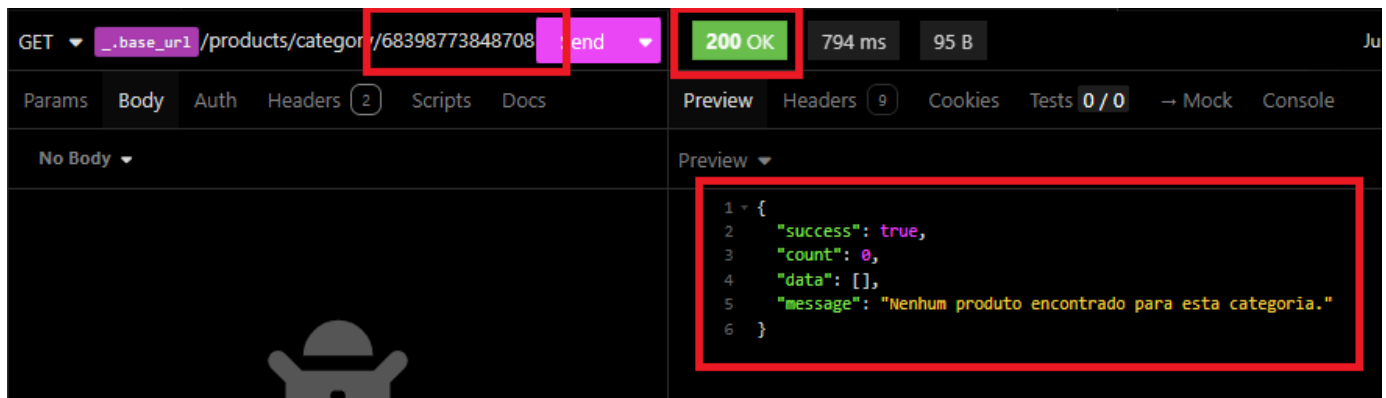
Como um cliente, eu quero listar todos os produtos disponíveis para que eu possa ver o catálogo completo da loja.

- **Critérios de Aceitação:**

- Ao realizar uma requisição **GET** para **/produtos**, a API **deve** listar todos os produtos cadastrados e retornar o status code **200 OK**.
- O corpo da resposta **deve** conter um array JSON com a lista de produtos.
- Cada objeto de produto no array **deve** conter os campos: **id** (inteiro), **nome** (string), **preço** (número), **descrição** (string), **quantidade** (inteiro) e **categoria** (string ou objeto).
- Se não houver produtos cadastrados, a API **deve** retornar um array vazio (**[]**) com status **200 OK** e a mensagem: **Nenhum produto encontrado para essa categoria.**

The screenshot shows a REST client interface with a GET request to `_.base_url/products` that has been sent. The response status is **200 OK** with a response time of 808 ms and a body size of 1130 B. The response body is displayed in JSON format, showing an array of two product objects. The first product is 'Smartphone Luffy' with a price of 799.99 and quantity of 499. The second product is 'Smartphone Y' with a price of 1299.99 and quantity of 50. Both products are categorized under 'Eletrônicos e Computadores'.

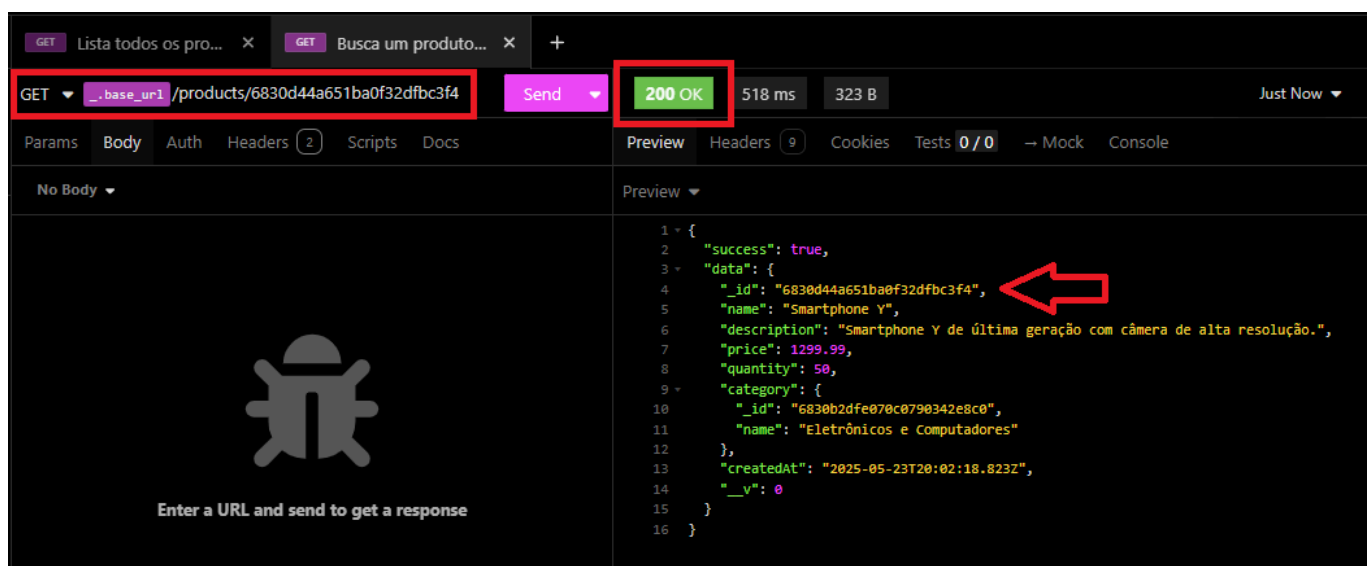
```
1 {
2   "success": true,
3   "count": 4,
4   "data": [
5     {
6       "_id": "6830b33ae070c0790342e8c5",
7       "name": "Smartphone Luffy",
8       "description": "Smartphone do Luffy de última geração com câmera de alta
9       resolução.",
10      "price": 799.99,
11      "quantity": 499,
12      "category": null,
13      "createdAt": "2025-05-23T17:41:14.515Z",
14      "__v": 0
15    },
16    {
17      "_id": "6830d44a651ba0f32dfbc3f4",
18      "name": "Smartphone Y",
19      "description": "Smartphone Y de última geração com câmera de alta
20      resolução.",
21      "price": 1299.99,
22      "quantity": 50,
23      "category": {
24        "_id": "6830b2dfe070c0790342e8c0",
25        "name": "Eletrônicos e Computadores"
26      },
27      "createdAt": "2025-05-23T20:02:18.822Z"
28    }
29  ]
30 }
```

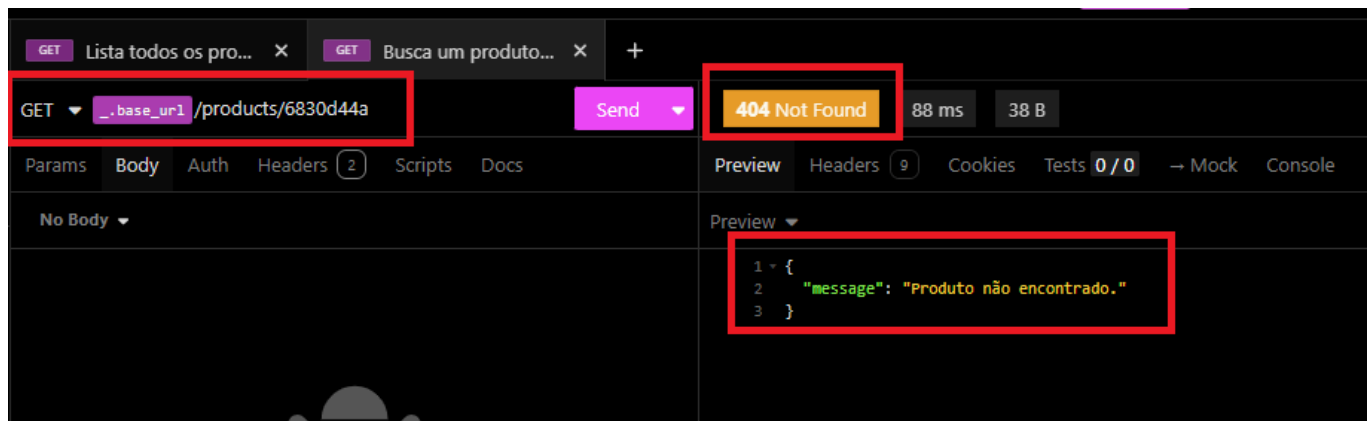


User Story 2: Busca de produto por ID

Como um cliente, **eu quero** buscar um produto específico pelo seu ID **para que** eu possa ver seus detalhes.

- **Critérios de Aceitação:**
 - **Dado** um ID de produto existente, **ao** realizar uma requisição **GET** para `/produtos/{id}`, a API **deve** retornar o status code `200 OK`.
 - O corpo da resposta **deve** conter um objeto JSON com os detalhes completos do produto solicitado correspondente ao `{id}` pesquisado.
 - **Dado** um ID de produto **inexistente**, **ao** realizar uma requisição **GET** para `/produtos/{id}`, a API **deve** retornar o status code `404 Not Found`.
 - Para um ID inexistente, o corpo da resposta **deve** conter a mensagem de erro `{"message": "Produto não encontrado"}`.

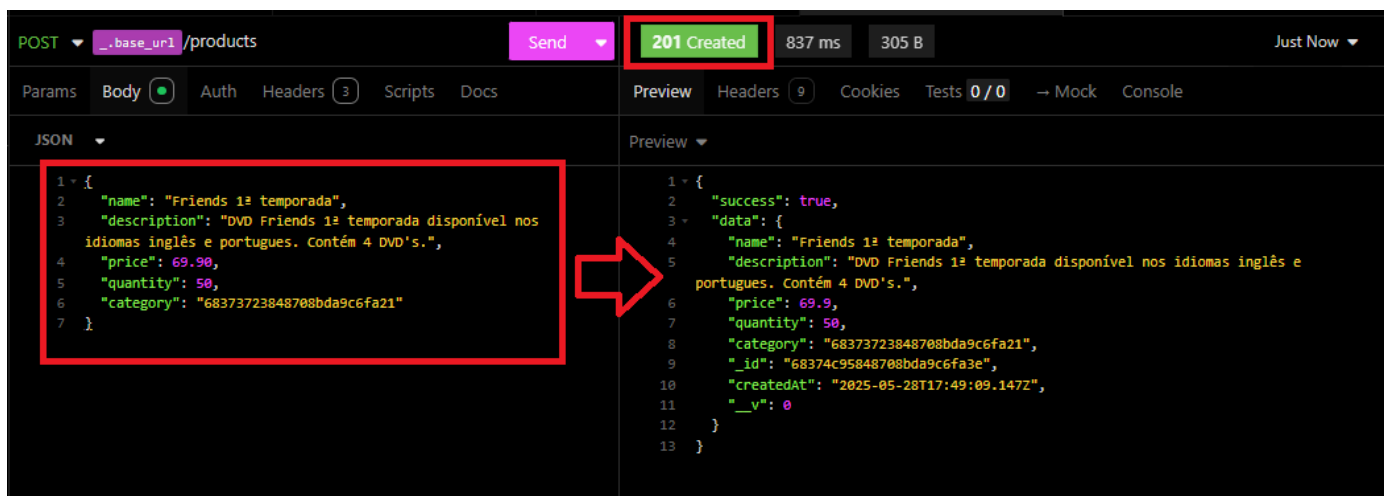




User Story 3: Cadastro de um novo produto

Como administrador, eu quero cadastrar um novo produto no sistema para que ele fique disponível no catálogo de vendas.

- **CrITÉrios de Aceitação:**
 - Ao realizar uma requisição **POST** para **/produtos** com um payload JSON válido (contendo **nome**, **preço**, **descrição**, **quantidade** e **categoria**), a API **deve** retornar o status code **201 Created**.
 - O corpo da resposta **deve** conter o objeto do produto recém-criado, incluindo o **id** gerado pelo sistema.
 - Ao tentar cadastrar um produto com um ou mais campos obrigatórios faltando no payload, a API **deve** retornar o status code **400 Bad Request**.
 - Ao tentar cadastrar um produto com tipos de dados inválidos (ex: **preço** como string), a API **deve** retornar o status code **400 Bad Request**.



POST `base_url1` /products Send 400 Bad Request 182 ms 115 B Just Now

Params Body Auth Headers (3) Scripts Docs Preview Headers (9) Cookies Tests 0/0 → Mock Console

JSON

```
1 {
2   "name": "The Vampire Diaries 10ª temporada",
3   "price": 39.90,
4   "quantity": 150,
5   "category": "68373723848708bda9c6fa21"
6 }
```

Preview

```
1 {
2   "message": "Por favor, preencha todos os campos obrigatórios: nome,
3     descrição, preço, quantidade e categoria."
4 }
```

POST `base_url1` /products Send 400 Bad Request 594 ms 87 B 2 Minutes Ago

Params Body Auth Headers (3) Scripts Docs Preview Headers (9) Cookies Tests 0/0 → Mock Console

JSON

```
1 {
2   "name": "The OC - Um estranho no paraíso 3ª
3     temporada",
4   "description": "The OC - Um estranho no paraíso
5     temporada disponível nos idiomas inglês e português.
6     contém 4 DVD's.",
7   "price": "preco",
8   "quantity": 150,
9   "category": "68373723848708bda9c6fa21"
10 }
```

Preview

```
1 {
2   "message": "Cast to Number failed for value \"preco\" (type string) at
3     path \"price\""
4 }
```

User Story 4: Atualização de um produto existente

Como administrador, eu quero atualizar as informações de um produto existente para que os dados do catálogo se mantenham corretos.

- **Critérios de Aceitação:**

- Dado um `id` de produto existente, ao realizar uma requisição PUT para `/produtos/{id}` com um payload JSON válido, a API deve retornar o status code `200 OK`.
- O corpo da resposta deve conter o objeto do produto com as informações atualizadas.
- Dado um ID de produto **inexistente**, ao realizar uma requisição PUT para `/produtos/{id}`, a API deve retornar o status code `404 Not Found`.
- Ao tentar atualizar um produto com tipos de dados inválidos nos campos, a API deve retornar o status code `400 Bad Request`.

PUT `_.base_url` `/product/6830d44a651ba0f32dfbc3f4` Send 200 OK 1.25 s 381 B Just Now

Params Body Auth Headers 3 Scripts Docs Preview Headers 9 Cookies Tests 0/0 → Mock Console

JSON

```
1 {
2   "name": "Smartphone Samsung",
3   "description": "Smartphone Samsung modelo de última geração
com câmera de alta resolução. Disponível nas cores: Black,
Blue, Pink",
4   "price": 1500.00,
5   "quantity": 250,
6   "category": "6830b2dfe070c0790342e8c0"
7 }
```

Preview

```
1 {
2   "success": true,
3   "data": {
4     "_id": "6830d44a651ba0f32dfbc3f4",
5     "name": "Smartphone Samsung",
6     "description": "Smartphone Samsung modelo de última geração com câmera de alta
resolução. Disponível nas cores: Black, Blue, Pink",
7     "price": 1500,
8     "quantity": 250,
9     "category": {
10      "_id": "6830b2dfe070c0790342e8c0",
11      "name": "Eletrônicos e Computadores"
12    },
13     "createdAt": "2025-05-23T20:02:18.823Z",
14     "_v": 0
15   }
16 }
```

PUT `_.base_url` `/products/` Send 404 Not Found 256 ms 152 B

Params Body Auth Headers 3 Scripts Docs Preview Headers 10 Cookies Tests 0/0 → M

JSON

```
1 {
2   "name": "Smartphone Samsung galaxy",
3   "description": "Smartphone Samsung modelo de última geração
com câmera de alta resolução. Disponível nas cores: Black,
Blue, Pink",
4   "price": 1500.00,
5   "quantity": 250,
6   "category": "6830b2dfe070c0790342e8c0"
7 }
```

Preview

Cannot PUT /api/products/

PUT `_.base_url` `/product/6830d44a651ba0f32dfbc3f` Send 400 Bad Request 268 ms 138 B Just Now

Params Body Auth Headers 3 Scripts Docs Preview Headers 9 Cookies Tests 0/0 → Mock Console

JSON

```
1 {
2   "name": "Smartphone Samsung galaxy",
3   "description": "Smartphone Samsung modelo de última
geração com câmera de alta resolução. Disponível nas
cores: Black, Blue, Pink",
4   "price": 1500.00,
5   "quantity": 250,
6 }
```

Preview

Bad Request

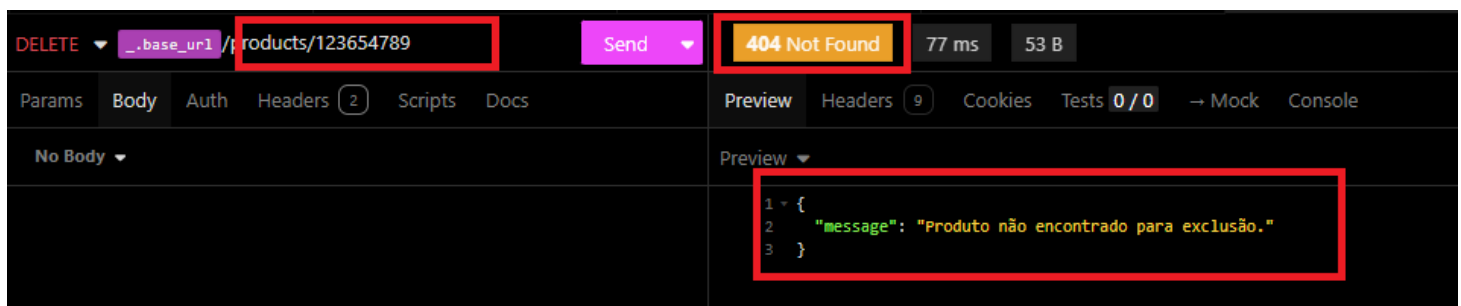
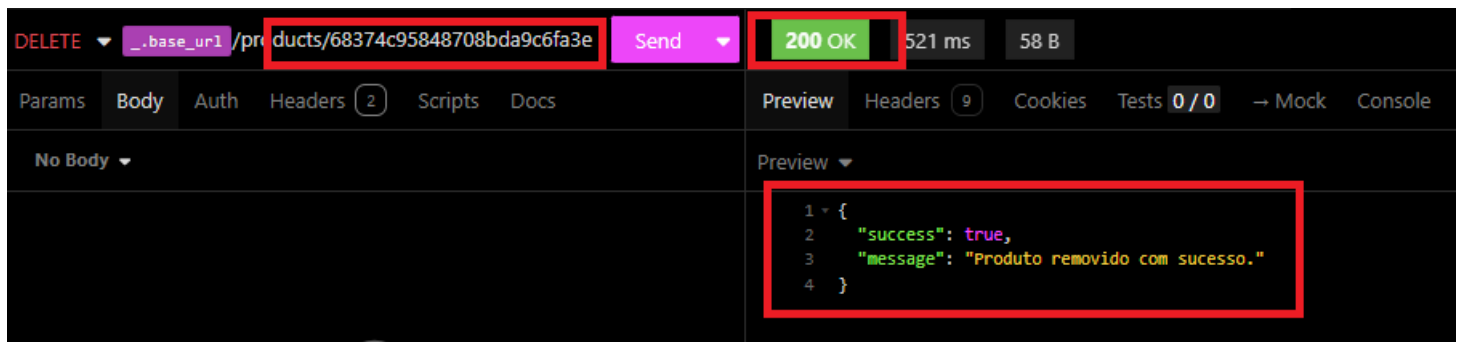
Não informa o ID categoria

User Story 5: Exclusão de um produto

Como administrador, eu quero excluir um produto do sistema para que ele não esteja mais disponível para venda.

- **Critérios de Aceitação:**

- **Dado** um ID de produto existente, **ao** realizar uma requisição **DELETE** para `/produtos/{id}`, a API **deve** retornar o status code **200 OK** e a mensagem de confirmação `{"message": "produto removido com sucesso"}`.
- **Dado** um ID de produto **inexistente**, **ao** realizar uma requisição **DELETE** para `/produtos/{id}`, a API **deve** retornar o status code **404 Not Found** e a mensagem de erro `{"message": "produto não encontrado para exclusão"}`.



User Story 6: Listagem de produtos por categoria

Como um cliente, eu quero listar todos os produtos de uma categoria específica para que eu possa encontrar itens semelhantes mais facilmente.

- **Critérios de Aceitação:**

- **Dado** um ID de categoria existente, **ao** realizar uma requisição **GET** para `/produtos/categoria/{categoriaId}`, a API **deve** retornar **200 OK** e uma lista de produtos pertencentes a essa categoria.
- **Dado** um ID de categoria **inexistente**, **ao** realizar uma requisição **GET** para `/produtos/categoria/{categoriaId}`, a API **deve** retornar **404 Not Found** e a mensagem de erro `{"message": "Categoria não encontrada"}`.
- **Dado** um ID de categoria que existe mas não possui produtos associados, a API **deve** retornar **200 OK** e um array vazio (`[]`).

GET **_.base_url** /products/category/68373723848708bda9c6fa3e **Send** **200 OK** 773 ms 1315 B Just Now ▾

Params **Body** Auth Headers (2) Scripts Docs

No Body ▾

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Introduction to Insomnia [↗](#)

Preview ▾

```
1 {
2   "success": true,
3   "count": 4,
4   "data": [
5     {
6       "_id": "68374c95848708bda9c6fa3e",
7       "name": "Friends 1ª temporada",
8       "description": "DVD Friends 1ª temporada disponível nos idiomas inglês e português. Contém 4 DVD's.",
9       "price": 69.9,
10      "quantity": 50,
11      "category": {
12        "_id": "68373723848708bda9c6fa21",
13        "name": "Séries e Filmes"
14      },
15      "createdAt": "2025-05-28T17:49:09.147Z",
16      "_v": 0
17    },
18    {
19      "_id": "68374d53848708bda9c6fa41",
20      "name": "Gossip Girl",
21      "description": "DVD Gossip Girl 5ª temporada disponível nos idiomas inglês e português. Contém 4 DVD's.",
22      "price": 49.9,
23      "quantity": 100,
24      "category": {
25        "_id": "68373723848708bda9c6fa21"
```

GET **_.base_url** /categories/78495486830b2dfe07 **Send** **404 Not Found** 209 ms 40 B

Params **Body** Auth Headers (2) Scripts Docs

No Body ▾

Categoria que não existe

Preview ▾

```
1 {
2   "message": "Categoria não encontrada."
3 }
```

GET **_.base_url** /products/category/68398773848708bda9c6fa3e **Send** **200 OK** 545 ms 95 B

Params **Body** Auth Headers (2) Scripts Docs

No Body ▾

Preview ▾

```
1 {
2   "success": true,
3   "count": 0,
4   "data": [],
5   "message": "Nenhum produto encontrado para esta categoria."
6 }
```