

Registrar Usuário e Logar

Registrando usuário com papel de **USER** informando username, password e role

The screenshot shows the Postman interface for a REST client. On the left, a sidebar lists various API endpoints under the 'Produtos' collection. The main panel displays a POST request to 'localhost:8080/auth/register'. The request body is a JSON object:

```
{  "username": "joao",  "password": "testesenha1@USER",  "role": "USER"}
```

. The response status is '200 OK' with a message 'Usuário registrado com sucesso'.

Registrando usuário com papel de **ADM** informando username, password e role

This screenshot shows a similar POST request in Postman, but for an administrator user. The request body is:

```
{  "username": "gerenteadm",  "password": "testesenha1@admin",  "role": "ADMIN"}
```

. The response is also '200 OK' with the message 'Usuário registrado com sucesso'.

Realizando autenticação com o perfil **ADM** criado anteriormente e gerando token de autenticação.

The screenshot shows a POST request to 'localhost:8080/auth/login' in Postman. The request body is the same administrator credentials as the previous request:

```
{  "username": "gerenteadm",  "password": "testesenha1@admin",  "role": "ADMIN"}
```

. The response status is '200 OK' and the body contains a JWT token:

```
{  "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJnZXJlbnRlYWRTIiwiaWF0IjoxNzQ0MDY1MDg2LjLjLjEHA10jE3NDQwNjg2ODZ9.c8eQfAtR2ktxFzL2APefhELtn9gvBx2eRkdQReJ46M"}
```

Criar Produtos

Criando produto com nível de acesso de usuário **ADM**

+

...

Produtos

GET loginGerarTokenAdmin

GET loginGerarTokenUsuario

POST registerUser

POST CriarProduto

POST criar pedido

POST AlterarProduto

POST Deletar Produto

POST BuscarProdutoPorId

POST buscar top5

POST MediaPedidoPorUsuario

POST FaturamentoMensal

POST buscarProdutosElasticPersonal...

GET https://fdceefeb47dc4a48920...

Produtos / CriarProduto

Save Share

POST localhost:8080/api/products

Send

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "nome": "FANTA",
3   "descricao": "REFRIGERANTE FANTA LATA",
4   "preco": 3.00,
5   "categoria": "BEBIDAS",
6   "quantidadeEstoque": 10,
7   "dataCriacao": "2025-04-04T17:00:00",
8   "dataAtualizacao": "2025-04-04T17:00:00"
9 }
```

Body Cookies Headers (11) Test Results

200 OK 588 ms 582 B Save Response

JSON Preview Visualize

```
1 {
2   "id": "45d8a0ba-e244-41ef-8931-8aa62d8d69cb",
3   "nome": "FANTA",
4   "descricao": "REFRIGERANTE FANTA LATA",
5   "preco": 3.00,
6   "categoria": "BEBIDAS",
7   "quantidadeEstoque": 10,
8   "dataCriacao": "2025-04-07T19:34:40.1054576",
9   "dataAtualizacao": "2025-04-07T19:34:40.1054576"
10 }
```

Exibindo Produto Criado no ElasticSearch Cloud.

Field	Contents
k id	"45d8a0ba-e244-41ef-8931-8aa62d8d69cb"
t nome	"FANTA"
t descricao	"REFRIGERANTE FANTA LATA"
# preco	3
k categoria	"BEBIDAS"
# quantidadeEstoque	10
📅 dataCriacao	[2025, 4, 7, 19, 34, 40, 105457600]
📅 dataAtualizacao	[2025, 4, 7, 19, 34, 40, 105457600]

Exibindo Produto criado no Banco de Dados.

	id	categoria	data_atualizacao	data_criacao	descricao	nome
	BLOB	BEBIDAS	2025-04-07 19:25:13.938243	2025-04-07 19:25:13.938243	REFRIGERANTE SPRITE 2L	SPRITE
	BLOB	BEBIDAS	2025-04-07 19:24:36.017899	2025-04-07 19:24:36.017899	COCA COLA 2L	COCA COLA
	BLOB	ALIMENTO	2025-04-06 22:21:24.518544	2025-04-06 01:37:51.183657	ARROZ ANCELE 5KG	Arroz
▶	BLOB	BEBIDAS	2025-04-07 19:34:40.105458	2025-04-07 19:34:40.105458	REFRIGERANTE FANTA LATA	FANTA
	BLOB	BEBIDAS	2025-04-07 19:25:26.913007	2025-04-07 19:25:26.913007	REFRIGERANTE FANTA 2L	FANTA
	BLOB	ALIMENTO	2025-04-06 22:21:24.518544	2025-04-06 01:38:34.342019	FEIJAO SÃO JOÃO 2KG	Feijão
	BLOB	higiene e limpeza	2025-04-06 01:22:41.801143	2025-04-06 00:06:13.545791	sabonete lux	sabonete
	BLOB	higiene e limpeza	2025-04-06 01:22:41.801143	2025-04-06 00:06:26.024369	sabonete nivea	sabonete
	BLOB	higiene e limpeza	2025-04-06 01:32:49.758459	2025-04-06 00:06:52.742424	shampoo Vela	sabonete

Alterar Produto

Realizada alteração do produto anteriormente criado como Fanta para Pepsi+descricao+preço
Product id: 45d8a0ba-e244-41ef-8931-8aa62d8d69cb

The screenshot shows a REST client interface with a sidebar on the left listing various API endpoints. The main area displays a PUT request to the endpoint `localhost:8080/api/products/45d8a0ba-e244-41ef-8931-8aa62d8d69cb`. The request body is a JSON object with the following fields: `nome` (PEPSI), `descricao` (REFRIGERANTE PEPSI LATA), `preco` (3.50), `categoria` (BEBIDAS), `quantidadeEstoque` (10), `dataCriacao` (2025-04-04T17:00:00), and `dataAtualizacao` (2025-04-04T17:00:00). The response status is 200 OK, and the response body is a JSON object with the same fields, but with the `id` field included: `id` (45d8a0ba-e244-41ef-8931-8aa62d8d69cb).

Exibindo Produto Alterado no ElasticSearch Cloud.

Field	Contents
k id	"45d8a0ba-e244-41ef-8931-8aa62d8d69cb"
t nome	"PEPSI"
t descricao	"REFRIGERANTE PEPSI LATA"
# preco	3.5
k categoria	"BEBIDAS"
# quantidadeEstoque	10
dataCriacao	[2025, 4, 4, 17, 0]
dataAtualizacao	[2025, 4, 4, 17, 0]

Exibindo Produto Alterado no Banco de Dados.

	id	categoria	data_atualizacao	data_criacao	descricao	nome
	BL08	BEBIDAS	2025-04-07 19:25:13.938243	2025-04-07 19:25:13.938243	REFRIGERANTE SPRITE 2L	SPRITE
	BL08	BEBIDAS	2025-04-07 19:24:36.017899	2025-04-07 19:24:36.017899	COCA COLA 2L	COCA COLA
	BL08	ALIMENTO	2025-04-06 22:21:24.518544	2025-04-06 01:37:51.183657	ARROZ ANCELE 5KG	Arroz
	BL08	BEBIDAS	2025-04-07 19:46:01.323176	2025-04-04 17:00:00.000000	REFRIGERANTE PEPSI LATA	PEPSI
	BL08	BEBIDAS	2025-04-07 19:25:26.913007	2025-04-07 19:25:26.913007	REFRIGERANTE FANTA 2L	FANTA
	BL08	ALIMENTO	2025-04-06 22:21:24.518544	2025-04-06 01:38:34.342019	FEIJAO SÃO JOÃO 2KG	Feijão
	BL08	higiene e limpeza	2025-04-06 01:22:41.801143	2025-04-06 00:06:13.545791	sabonete lux	sabonete
	BL08	higiene e limpeza	2025-04-06 01:22:41.801143	2025-04-06 00:06:26.024369	sabonete nivea	sabonete
	BL08	higiene e limpeza	2025-04-06 01:32:49.758459	2025-04-06 00:06:52.742424	shampoo Vela	sabonete
*	NULL	NULL	NULL	NULL	NULL	NULL

Busca Produto por ID

Buscando Produto Criado anteriormente por ID: 45d8a0ba-e244-41ef-8931-8aa62d8d69cb

The screenshot shows a REST client interface with a sidebar on the left listing various endpoints. The main panel displays a GET request to `localhost:8080/api/products/45d8a0ba-e244-41ef-8931-8aa62d8d69cb`. The response is a 200 OK status with a JSON body containing product details.

```
{
  "id": "45d8a0ba-e244-41ef-8931-8aa62d8d69cb",
  "nome": "PEPSI",
  "descricao": "REFRIGERANTE PEPSI LATA",
  "preco": 3.50,
  "categoria": "BEBIDAS",
  "quantidadeEstoque": 10,
  "dataCriacao": "2025-04-04T17:00:00",
  "dataAtualizacao": "2025-04-07T19:46:01.323176"
}
```

Buscar Produtos filtrando pelo Nome(ElasticSearch)

Buscar produtos filtrados pelo nome permitindo um limite de erros.

The screenshot shows a REST client interface with a sidebar on the left. The main panel displays a GET request to `localhost:8080/api/products/search?nome=PEPI`. The response is a 200 OK status with a JSON body containing a list of product details.

Key	Value	Description	Bulk Edit
<input type="checkbox"/> categoria	higiene e lim		
<input type="checkbox"/> minPreco	2		
<input type="checkbox"/> maxPreco	20		
<input checked="" type="checkbox"/> nome	PEPI		

```
[
  {
    "id": "45d8a0ba-e244-41ef-8931-8aa62d8d69cb",
    "nome": "PEPSI",
    "descricao": "REFRIGERANTE PEPSI LATA",
    "preco": 3.50,
    "categoria": "BEBIDAS",
    "quantidadeEstoque": 10,
    "dataCriacao": "2025-04-04T17:00:00",
    "dataAtualizacao": "2025-04-04T17:00:00"
  }
]
```

Buscar Produtos filtrando pela Categoria(ElasticSearch)

Buscar produtos filtrados pela Categoria permitindo um limite de erros.

Produtos / buscarProdutosElasticPersonalizado

SaveShare

GETlocalhost:8080/api/products/search?categoria=BEBIDA

Send

ParamsAuthorizationHeaders (10)BodyScriptsSettingsCookies

<input type="checkbox"/>	Key	Value	Description	Bulk Edit
<input type="checkbox"/>	nome	PEPI		
<input checked="" type="checkbox"/>	categoria	BEBIDA		
<input type="checkbox"/>	minPreco	2		
<input type="checkbox"/>	maxPreco	5		
	Key	Value	Description	

BodyCookiesHeaders (11)Test Results200 OK189 ms1.28 KBSave Response

{ } JSONPreviewVisualize

```
1  [
2    {
3      "id": "1b6123de-4593-4cb7-a534-870c72bd077e",
4      "nome": "COCA COLA",
5      "descricao": "COCA COLA 2L",
6      "preco": 12.00,
7      "categoria": "BEBIDAS",
8      "quantidadeEstoque": 10,
9      "dataCriacao": "2025-04-07T19:24:36.0178987",
10     "dataAtualizacao": "2025-04-07T19:24:36.0178987"
11   },
```

Buscar Produtos filtrando por faixa de preço(ElasticSearch)

Buscar produtos filtrados por faixa de valor.

The screenshot shows a REST client interface with the following details:

- URL:** `localhost:8080/api/products/search?minPreco=2&maxPreco=5`
- Method:** GET
- Params:**

Key	Value
minPreco	2
maxPreco	5
- Status:** 200 OK
- Response Body (JSON):**

```
[{"id": "de88f637-65b5-4c8b-a2bc-7075333f9ce7", "nome": "sabonete", "descricao": "shampoo Vela", "preco": 3.00, "categoria": "higiene e limpeza", "quantidadeEstoque": 8, "dataCriacao": "2025-04-06T00:06:52.742424", "dataAtualizacao": "2025-04-06T01:31:47.11656"}, {"id": "45d8a0ba-e244-41ef-8931-8aa62d8d69cb", "nome": "PEPSI", "descricao": "REFRIGERANTE PEPSI LATA", "preco": 3.50, "categoria": "BEBIDAS", "quantidadeEstoque": 10, "dataCriacao": "2025-04-04T17:00:00", "dataAtualizacao": "2025-04-04T17:00:00"}]
```

-Caso o usuário com perfil de **ADM** tentar buscar os produtos sem selecionar nenhum filtro, será retornado todos os Produtos que tenham quantidade em estoque > 0.

Criar Pedido

Nesse momento meu pedido se encontra com a quantidade 10 em estoque.

Document id: 1b6123de-4593-4cb7-a534-870c72bd077e

Field	Contents
<div>k</div> id	"1b6123de-4593-4cb7-a534-870c72bd077e"
<div>t</div> nome	"COCA COLA"
<div>t</div> descricao	"COCA COLA 2L"
<div>#</div> preco	12
<div>k</div> categoria	"BEBIDAS"
<div>#</div> quantidadeEstoque	10
<div>📅</div> dataCriacao	[2025, 4, 7, 19, 24, 36, 17898700]
<div>📅</div> dataAtualizacao	[2025, 4, 7, 19, 24, 36, 17898700]

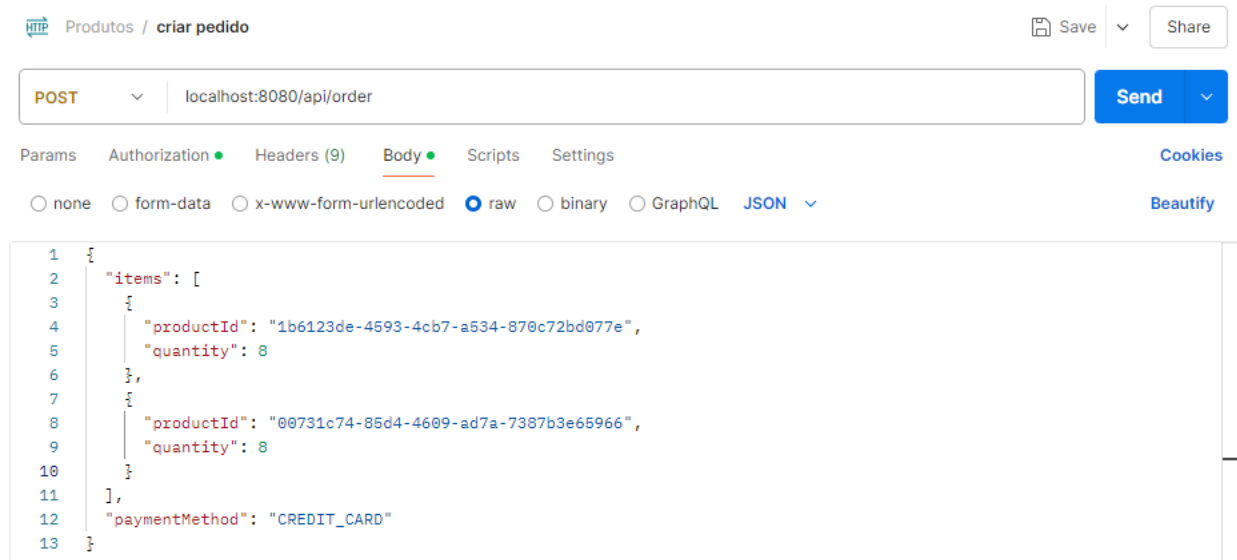
Document id: 00731c74-85d4-4609-ad7a-7387b3e65966

Show

Field	Contents
<div>k</div> id	"00731c74-85d4-4609-ad7a-7387b3e65966"
<div>t</div> nome	"SPRITE"
<div>t</div> descricao	"REFRIGERANTE SPRITE 2L"
<div>#</div> preco	8
<div>k</div> categoria	"BEBIDAS"
<div>#</div> quantidadeEstoque	10
<div>📅</div> dataCriacao	[2025, 4, 7, 19, 25, 13, 938243300]
<div>📅</div> dataAtualizacao	[2025, 4, 7, 19, 25, 13, 938243300]

Realizando Pedido e Pagamento

Requisição: nesse momento é passado o ID do Produto e a quantidade a ser Pedida.



Produtos / criar pedido

POST localhost:8080/api/order

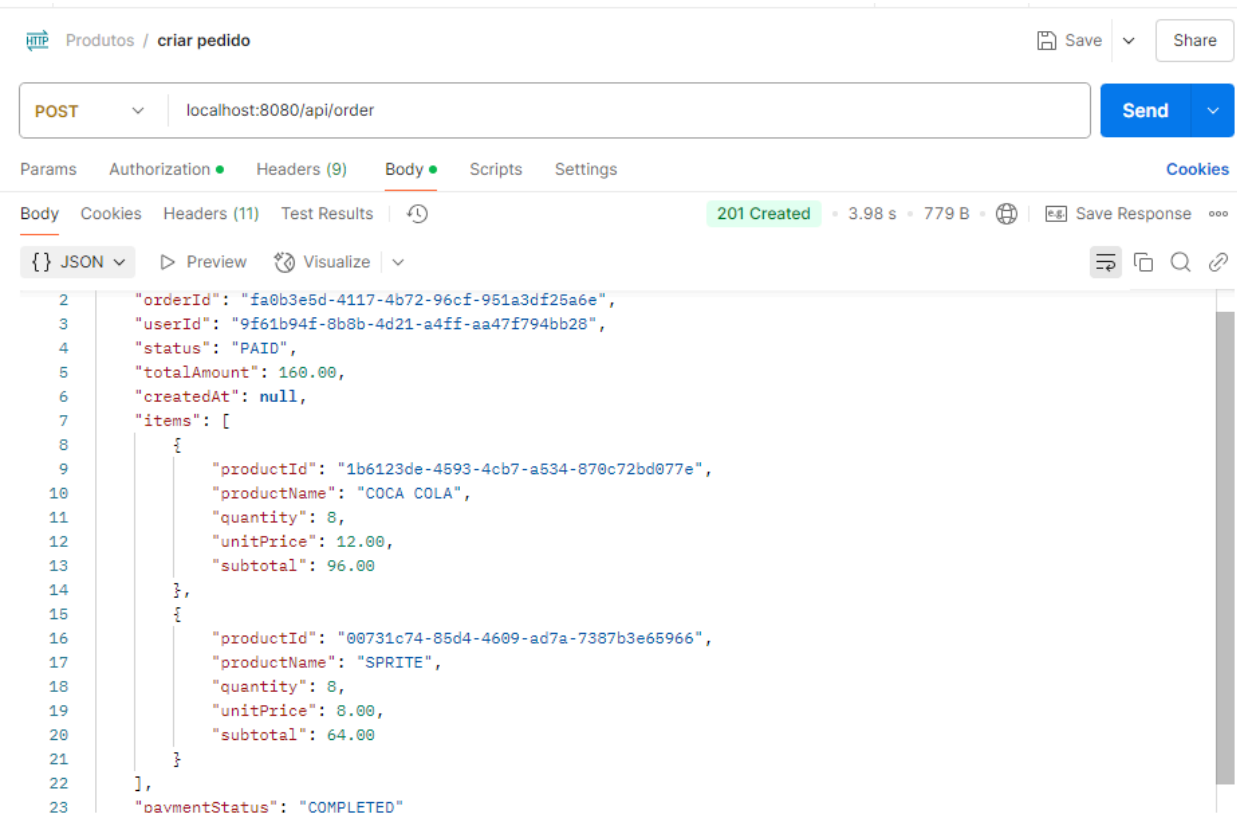
Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "items": [
3     {
4       "productId": "1b6123de-4593-4cb7-a534-870c72bd077e",
5       "quantity": 8
6     },
7     {
8       "productId": "00731c74-85d4-4609-ad7a-7387b3e65966",
9       "quantity": 8
10    }
11  ],
12  "paymentMethod": "CREDIT_CARD"
13 }
```

Resposta:



Produtos / criar pedido

POST localhost:8080/api/order

Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

Body Cookies Headers (11) Test Results

201 Created • 3.98 s • 779 B • Save Response

{ JSON Preview Visualize

```
2 "orderId": "fa0b3e5d-4117-4b72-96cf-951a3df25a6e",
3 "userId": "9f61b94f-8b8b-4d21-a4ff-aa47f794bb28",
4 "status": "PAID",
5 "totalAmount": 160.00,
6 "createdAt": null,
7 "items": [
8   {
9     "productId": "1b6123de-4593-4cb7-a534-870c72bd077e",
10    "productName": "COCA COLA",
11    "quantity": 8,
12    "unitPrice": 12.00,
13    "subtotal": 96.00
14  },
15  {
16    "productId": "00731c74-85d4-4609-ad7a-7387b3e65966",
17    "productName": "SPRITE",
18    "quantity": 8,
19    "unitPrice": 8.00,
20    "subtotal": 64.00
21  }
22 ],
23 "paymentStatus": "COMPLETED"
```

Neste momento o Pedido é Pago, é enviado uma ordem de Pagamento para o Kafka através do tópico **order.paid** que recebe o Id do pedido. Nossa aplicação consome o user do tópico, debita o valor no estoque do pedido e atualiza as informações no elasticsearch e na base de dados Mysql.

Informação no kafka com o número do pedido.

order.paid

Query with Flink

Enable Tableflow

Actions

Messages

Monitor

Data contracts

New

Configuration

Production in last hour 1 messages

Consumption in last hour 1 messages

Total messages 11

Retention time 1 week

Partition All

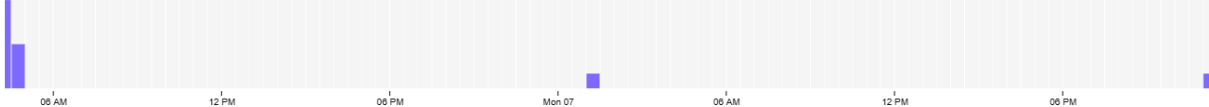
Consume Latest

Max results 1,000

CSV

JSON

Search...



Timestamp	Partition	Offset	Key	Value
2025-04-07T23:22:21.884Z	0	4		fa0b3e5d-4117-4b72-96cf-951a3df25a6e

Neste momento meu pedido se encontra assim após o pedido:

Document id: 00731c74-85d4-4609-ad7a-7387b3e65966

Field	Contents
id	"00731c74-85d4-4609-ad7a-7387b3e65966"
nome	"SPRITE"
descricao	"REFRIGERANTE SPRITE 2L"
preco	8
categoria	"BEBIDAS"
quantidadeEstoque	2
dataCriacao	[2025, 4, 7, 19, 25, 13, 938243000]
dataAtualizacao	[2025, 4, 7, 19, 25, 13, 938243000]



Document id: 1b6123de-4593-4cb7-a534-870c72bd077e



Field	Contents
id	"1b6123de-4593-4cb7-a534-870c72bd077e"
nome	"COCA COLA"
descricao	"COCA COLA 2L"
preco	12
categoria	"BEBIDAS"
quantidadeEstoque	2
dataCriacao	[2025, 4, 7, 19, 24, 36, 17899000]
dataAtualizacao	[2025, 4, 7, 19, 24, 36, 17899000]

Anteriormente havia 10 produtos de cada, gerei uma ordem para esses 2 produtos passando 8 produtos.Com isso restaram apenas 2 produtos em estoque.

Criar Pedido (Sem item no estoque)


Buscando pedidos com quantidade em estoque zerados




Document id: 00731c74-85d4-4609-ad7a-7387b3e65966		 
Field	Contents	
k id	"00731c74-85d4-4609-ad7a-7387b3e65966"	
t nome	"SPRITE"	
t descricao	"REFRIGERANTE SPRITE 2L"	
# preco	8	
k categoria	"BEBIDAS"	
# quantidadeEstoque	0	
dataCriacao	[2025, 4, 7, 19, 25, 13, 938243000]	
dataAtualizacao	[2025, 4, 7, 21, 50, 16, 196623000]	




Document id: 1b6123de-4593-4cb7-a534-870c72bd077e		  Show 5 fewer fields
Field	Contents	
k id	"1b6123de-4593-4cb7-a534-870c72bd077e"	
t nome	"COCA COLA"	
t descricao	"COCA COLA 2L"	
# preco	12	
k categoria	"BEBIDAS"	
# quantidadeEstoque	0	
dataCriacao	[2025, 4, 7, 19, 24, 36, 17899000]	
dataAtualizacao	[2025, 4, 7, 21, 50, 16, 196623000]	





Realizando chamada de pedidos passando os produtos e as quantidades.

```
1 {
2   "items": [
3     {
4       "productId": "1b6123de-4593-4cb7-a534-870c72bd077e",
5       "quantity": 1
6     },
7     {
8       "productId": "00731c74-85d4-4609-ad7a-7387b3e65966",
9       "quantity": 1
10    }
11  ]
12 }
```

Body Cookies Headers (10) Test Results 

400 Bad Request • 67 ms • 334 B •   Save Response 


Raw  Preview  Visualize 

1 Estoque insuficiente

Result Grid		Filter Rows:		Edit:	Export/Import:		Wrap C
	id	created_at	status		total_amount	user_id	
	BLOB	2025-04-06 01:41:20.119718	PAID		84.00	BLOB	
	BLOB	2025-04-07 21:50:13.653864	PAID		20.00	BLOB	
	BLOB	2025-04-06 22:21:22.006680	PAID		336.00	BLOB	
	BLOB	2025-04-06 01:17:56.283062	PAID		21.00	BLOB	
	BLOB	2025-04-06 01:21:02.070403	PAID		0.00	BLOB	
	BLOB	2025-04-06 01:32:49.384832	PAID		3.00	BLOB	
▶	BLOB	2025-04-07 22:32:04.945840	CANCELED		0.00	BLOB	

Exibindo relatório dos 5 principais usuários que mais ganham , com possibilidade de filtro por dados.

 Produtos / **buscar top5**

GETlocalhost:8080/api/reports/top-spenders

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

The authorization header will be automatically generated when you send the request. [Learn more about Bearer Token](#) authorization.

Token

eyJhbGciOiJIUzI1NiJ9.eyJzdWUiOiJnZXJI

Body

Cookies

Headers (11)

Test Results

200 OK

139 ms

776 B

Save Response

{ } JSON

Preview

Visualize

```
1  [
2    {
3      "userId": "112d4133-2a77-4785-b284-f05b45f55273",
4      "userName": "Denis",
5      "totalSpent": 336.00
6    },
7    {
8      "userId": "9f61b94f-8b8b-4d21-a4ff-aa47f794bb28",
9      "userName": "joao",
10     "totalSpent": 160.00
11   }
12 ]
```

Retornar o ticket médio dos pedidos por usuário, com possibilidade de filtro por dados

Produtos / **MediaPedidoPorUsuario**

GET | localhost:8080/api/reports/average-ticket/112d4133-2a77-4785-b284-f05b45f55273

Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type: Bearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJnZXJl

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (11) Test Results

200 OK • 80 ms • 408 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "O ticket medio do funcionario consultado é: ",
3   "average": 48.6
4 }
```

Retornar o valor total faturado no mês atual

Produtos / **FaturamentoMensal**

GET | localhost:8080/api/reports/monthly-revenue

Send

Params Authorization Headers (7) Body Scripts Settings Cookies

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJnZXJl

Body Cookies Headers (11) Test Results

200 OK • 318 ms • 443 B • Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Faturamento mensal calculado com sucesso",
3   "revenue": 686.0,
4   "currency": "BRL",
5   "period": "APRIL/2025"
6 }
```