

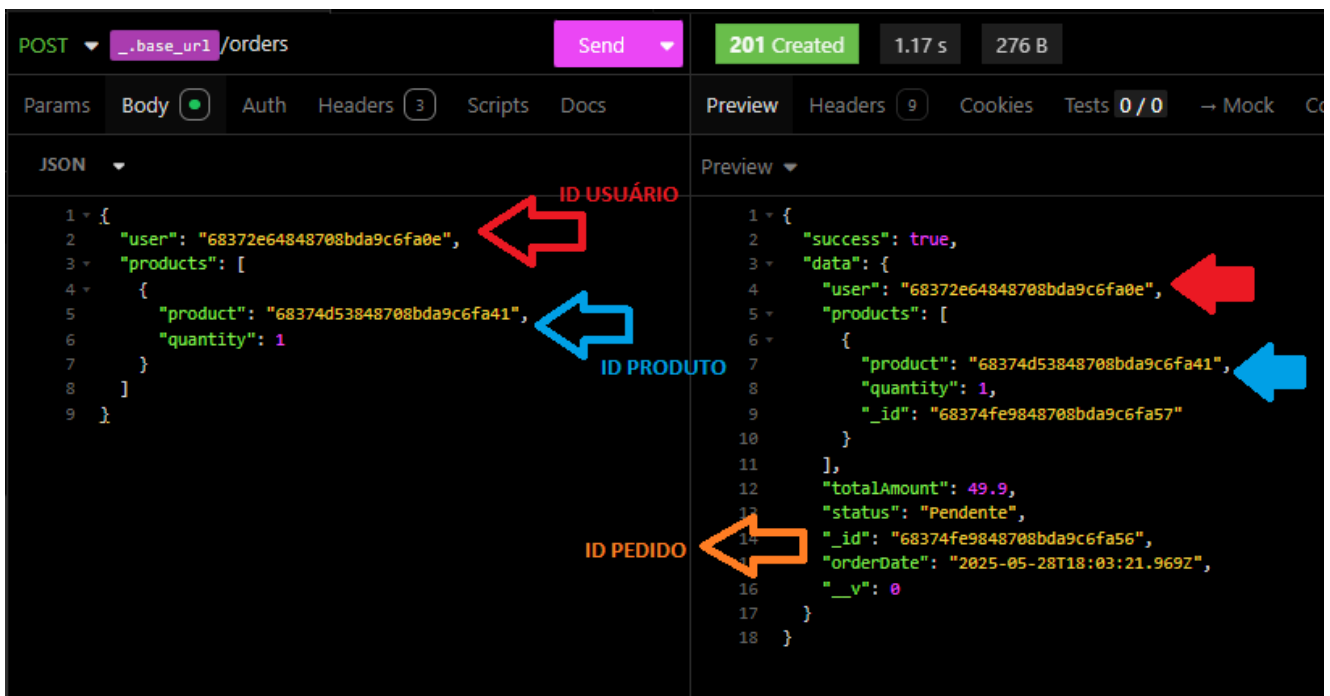
Pedidos

User Story 16: Cadastro de um novo pedido

Como um cliente, eu quero criar um novo pedido para que eu possa comprar os produtos desejados.

- **CrITÉrios de Aceitação:**

- Ao realizar uma requisição **POST** para **/pedidos** com um payload JSON válido (contendo os dados necessários para o pedido, a serem definidos ao testar a API), a API **deve** retornar o status code **201 Created**.
- A resposta **deve** conter o objeto do pedido criado, incluindo um **id** gerado pela API e outros detalhes relevantes do pedido.
- Ao tentar criar um pedido com um payload JSON incompleto (faltando informações obrigatórias), a API **deve** retornar o status code **400 Bad Request** e uma mensagem indicando os campos ausentes.
- Ao tentar criar um pedido com dados inválidos (ex: um ID de produto inexistente), a API **deve** retornar o status code **404 Not Found** com a mensagem: `{"message": "Produto com ID {id} não encontrado. Por favor, forneça um ID de produto válido e existente."}`



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

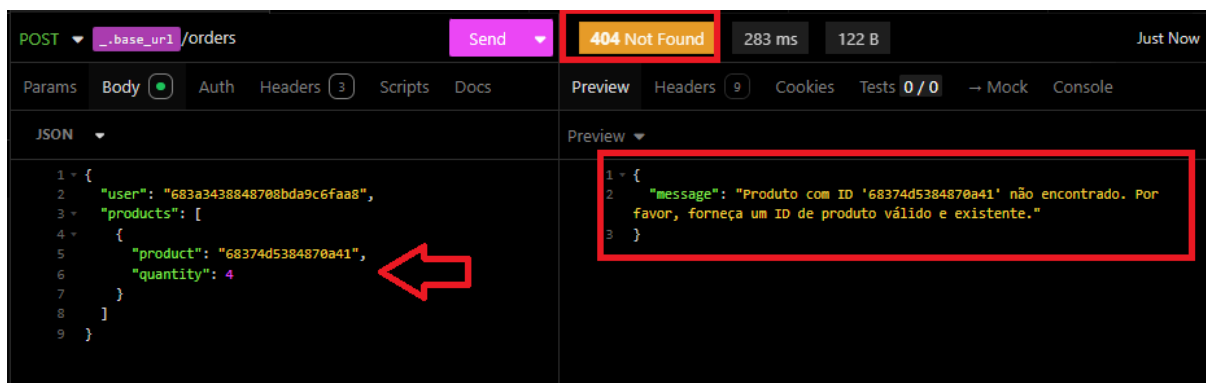
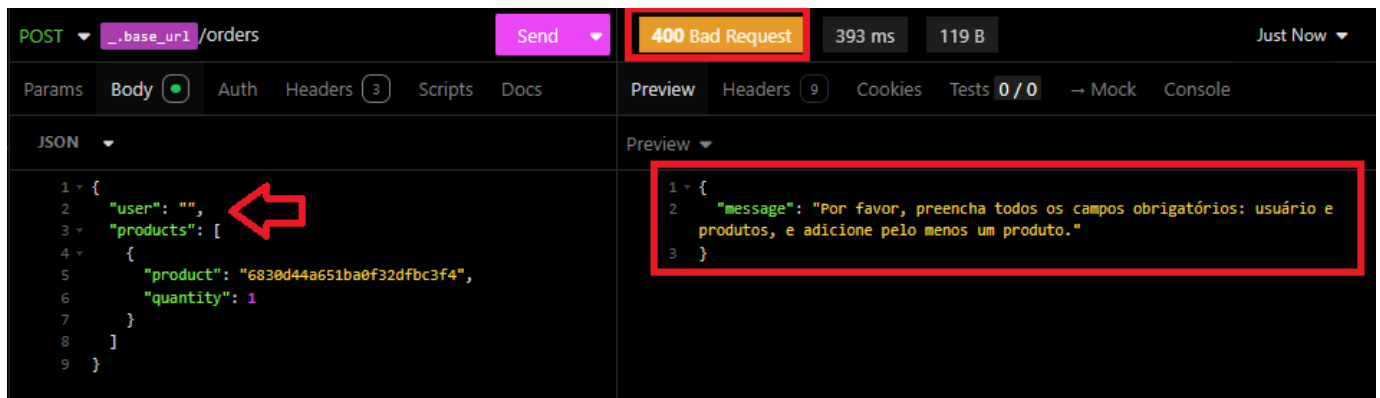
- Request:**
 - Method: **POST**
 - URL: `_.base_url/orders`
 - Body (JSON):

```
1 {
2   "user": "68372e64848708bda9c6fa0e",
3   "products": [
4     {
5       "product": "68374d53848708bda9c6fa41",
6       "quantity": 1
7     }
8   ]
9 }
```
- Response:**
 - Status: **201 Created**
 - Time: **1.17 s**
 - Size: **276 B**
 - Body (JSON):

```
1 {
2   "success": true,
3   "data": {
4     "user": "68372e64848708bda9c6fa0e",
5     "products": [
6       {
7         "product": "68374d53848708bda9c6fa41",
8         "quantity": 1,
9         "_id": "68374fe9848708bda9c6fa57"
10      }
11    ],
12    "totalAmount": 49.9,
13    "status": "Pendente",
14    "_id": "68374fe9848708bda9c6fa56",
15    "orderDate": "2025-05-28T18:03:21.969Z",
16    "__v": 0
17  }
18 }
```

Annotations on the image:

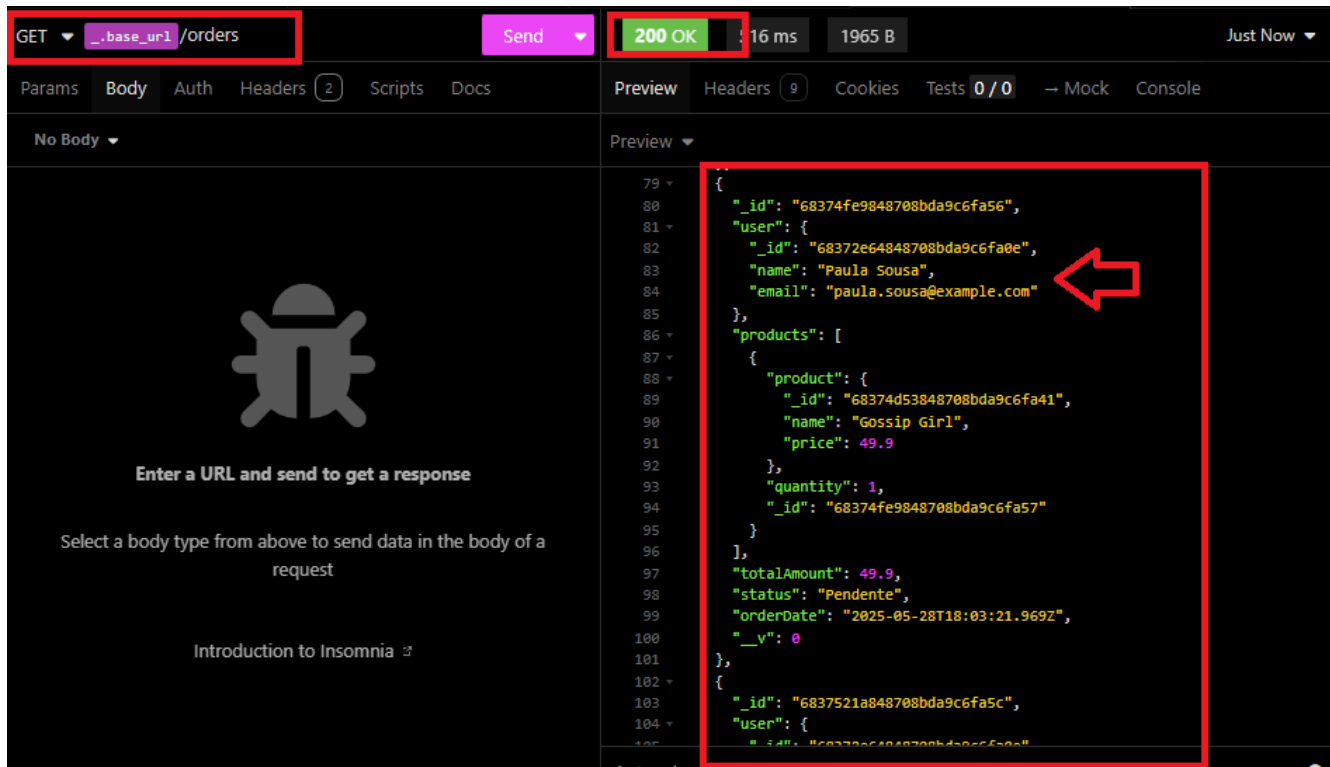
- ID-USUÁRIO:** Points to the `"user"` field in the request body.
- ID PRODUTO:** Points to the `"product"` field in the request body.
- ID PEDIDO:** Points to the `"_id"` field in the response body.



User Story 17: Listagem de todos os pedidos

Como administrador, eu quero listar todos os pedidos realizados para que eu possa gerenciar as vendas.

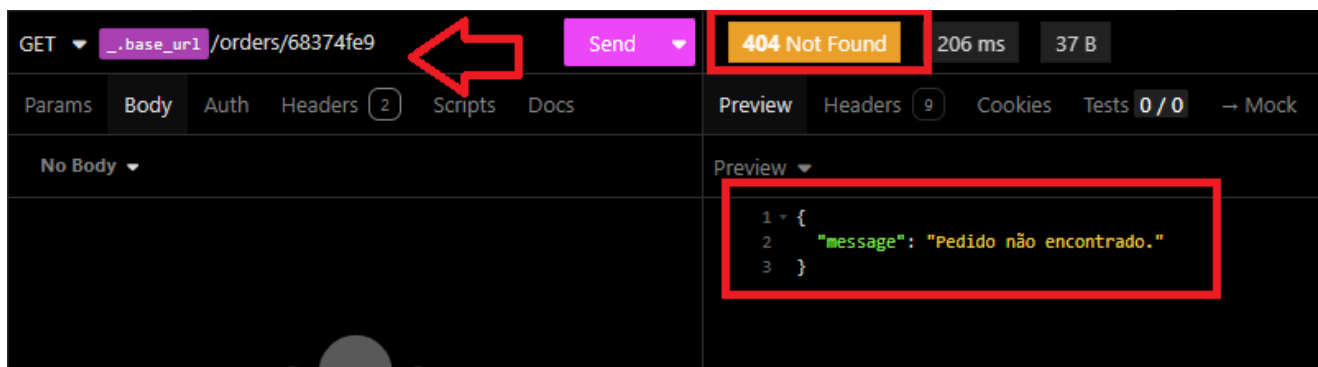
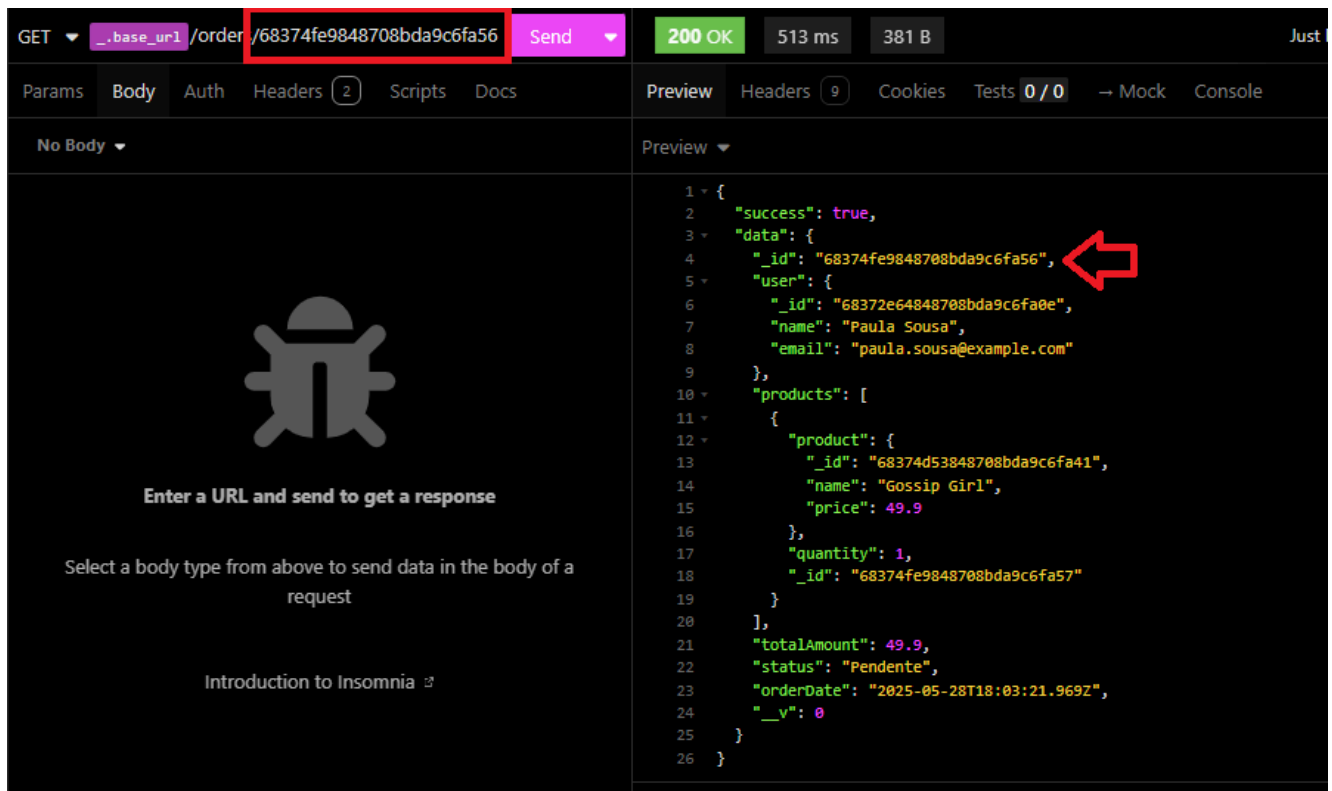
- **Critérios de Aceitação:**
 - Ao realizar uma requisição `GET` para `/pedidos`, a API **deve** retornar o status code `200 OK`.
 - A resposta **deve** conter um array JSON com a lista de todos os pedidos.
 - Cada objeto de pedido no array **deve** conter informações relevantes (a serem definidas ao testar a API, como `id`, `usuarioId`, `itens`, `status`, `data_criacao`, etc.).



User Story 18: Busca de um pedido por ID

Como um cliente ou administrador, eu quero buscar um pedido específico pelo seu ID para que eu possa verificar os detalhes.

- **CrITÉrios de Aceitação:**
 - **Dado** um ID de pedido existente, **ao** realizar uma requisição **GET** para `/pedidos/{id}`, a API **deve** retornar o status code **200 OK**.
 - A resposta **deve** conter um objeto JSON com os detalhes do pedido solicitado.
 - **Dado** um ID de pedido **inexistente**, **ao** realizar uma requisição **GET** para `/pedidos/{id}`, a API **deve** retornar o status code **404 Not Found**.
 - Para um ID inexistente, o corpo da resposta **deve** conter uma mensagem de erro informativa (a ser verificada ao testar, ex: `{"message": "Pedido não encontrado"}`).



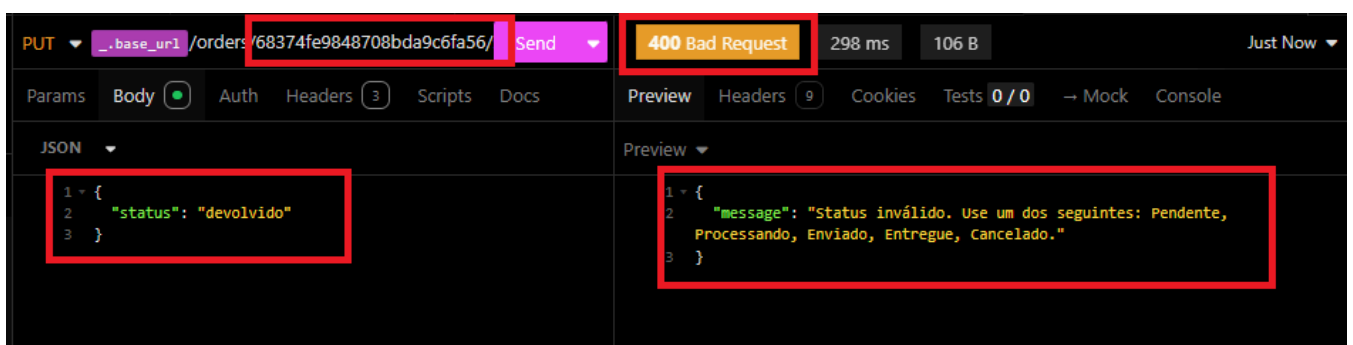
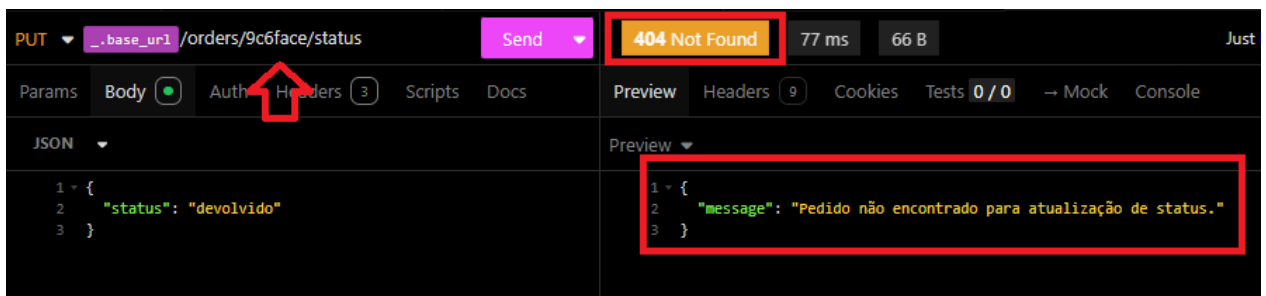
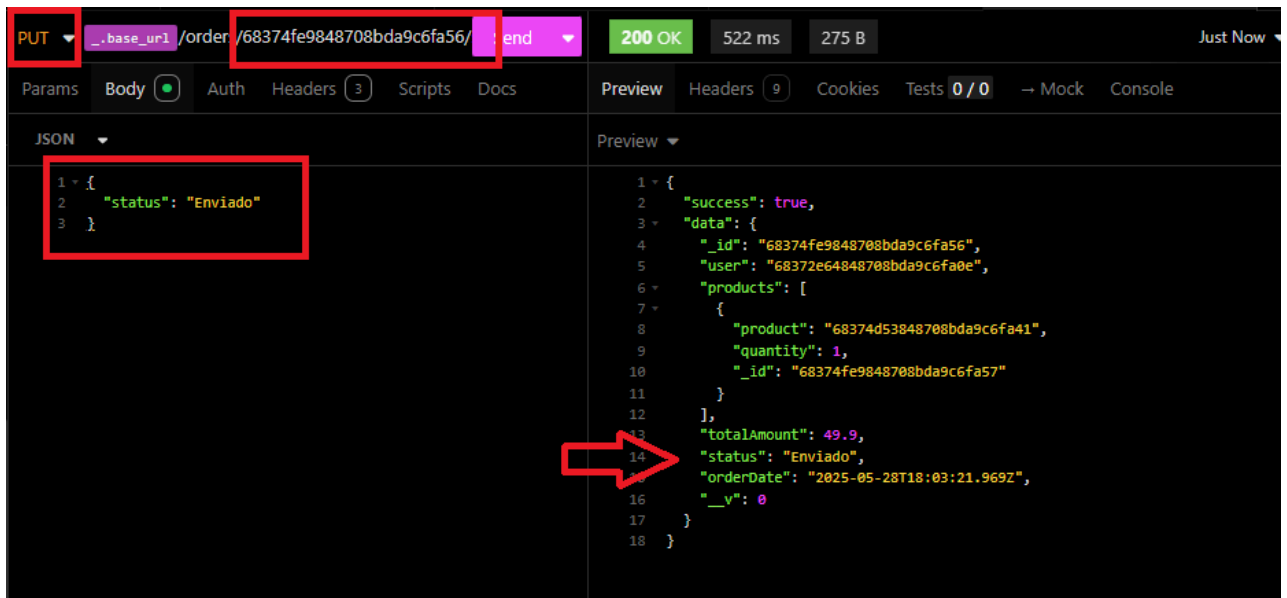
User Story 19: Atualização do status de um pedido por ID

Como administrador, eu quero atualizar o status de um pedido (ex: "pendente", "processando", "enviado", "entregue", "cancelado") para que o cliente seja informado sobre o andamento da sua compra e para controle interno.

- **Critérios de Aceitação:**
 - Dado um ID de pedido existente, ao realizar uma requisição **PUT** para `/pedidos/{id}` com um payload JSON contendo o novo **status**, a API deve retornar o status code **200 OK**.
 - A resposta deve conter o objeto do pedido com o **status** atualizado.
 - Dado um ID de pedido **inexistente**, ao realizar uma requisição **PUT** para `/pedidos/{id}`, a API deve retornar o status code **404 Not Found** e a

mensagem: {"message": "Pedido não encontrado para atualização de status."}

- Ao tentar atualizar o status com um valor inválido, a API **deve** retornar o status code 400 Bad Request e uma mensagem: {"message": "Status inválido. Use um dos seguintes: Pendente, Processado, Enviado, Entregue, Cancelado."}

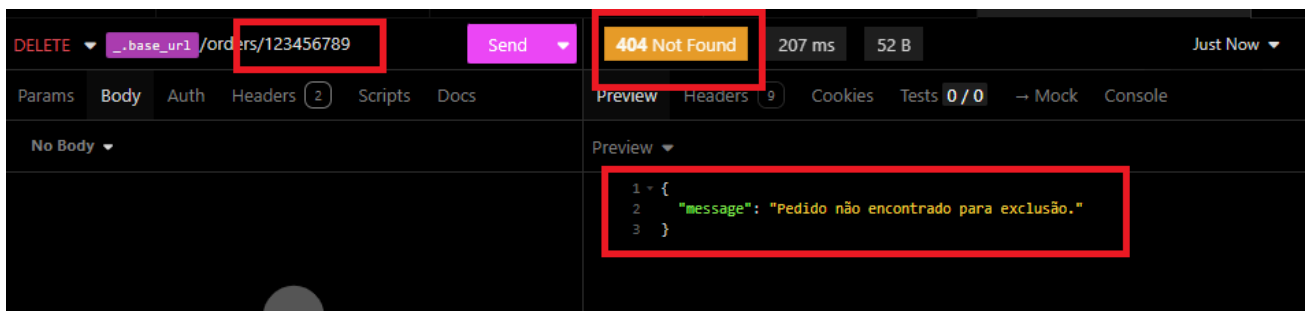
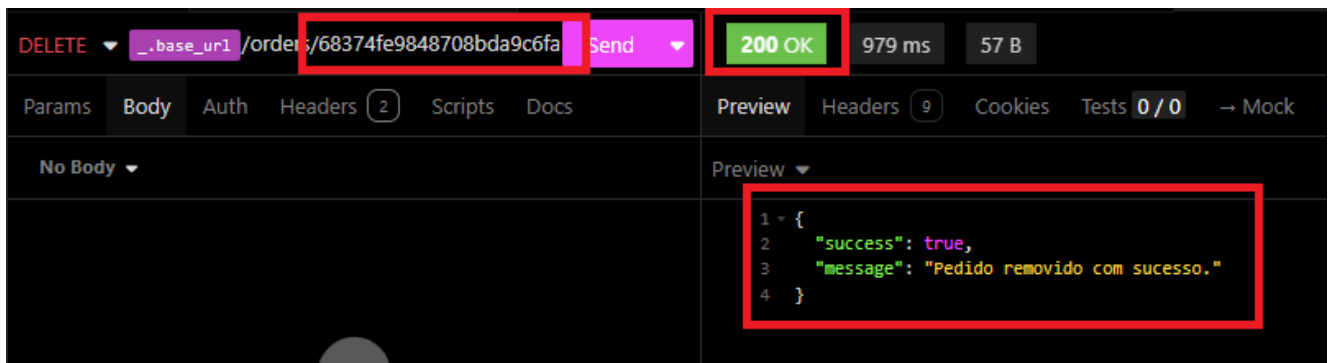


User Story 20: Exclusão de um pedido por ID

Como administrador, eu quero excluir um pedido específico para fins de organização ou cancelamento definitivo.

- **Critérios de Aceitação:**

- **Dado** um ID de pedido existente, **ao** realizar uma requisição **DELETE** para `/pedidos/{id}`, a API **deve** retornar o status code **200 OK**.
- Se a API retorna **200 OK**, o corpo da resposta **deve** conter uma mensagem de sucesso (a ser verificada ao testar, ex: `{"message": "Pedido removido com sucesso"}`).
- **Dado** um ID de pedido **inexistente**, **ao** realizar uma requisição **DELETE** para `/pedidos/{id}`, a API **deve** retornar o status code **404 Not Found** com uma mensagem informativa: `{"message": "Pedido não encontrado para exclusão."}`.



User Story 21: Listagem de todos os pedidos de um usuário específico

Como um cliente, eu quero listar todos os meus pedidos para que eu possa acompanhar meu histórico de compras.

- **Critérios de Aceitação:**

- Ao realizar uma requisição **GET** para `/pedidos/usuario/{usuarioId}` com um `usuarioId` existente que possui pedidos, a API **deve** retornar o status code **200 OK** e uma lista dos pedidos daquele usuário.
- Se o `usuarioId` existir, mas não houver pedidos associados, a API **deve** retornar o status code **200 OK** e uma lista vazia (`[]`) com uma mensagem `{"message": "Nenhum pedido encontrado para este usuário."}`.

- Se o `usuarioId` fornecido for inválido ou inexistente, a API **deve** retornar o status code `404 Not Found` com uma mensagem `{ "message": "Usuário não encontrado." }`.

GET `_.base_url` /orders/user/68372e64848708bda9c6fa0e Send 200 OK 19 s 758 B Just Now

Params Body Auth Headers (2) Scripts Docs Preview Headers (9) Cookies Tests 0/0 → Mock Console

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

```
1 {
2   "success": true,
3   "count": 2,
4   "data": [
5     {
6       "_id": "68374fe9848708bda9c6fa56",
7       "user": {
8         "_id": "68372e64848708bda9c6fa0e",
9         "name": "Paula Sousa",
10        "email": "paula.sousa@example.com"
11      },
12      "products": [
13        {
14          "product": {
15            "_id": "68374d53848708bda9c6fa41",
16            "name": "Gossip Girl",
17            "price": 49.9
18          },
19          "quantity": 1,
20          "_id": "68374fe9848708bda9c6fa57"
21        }
22      ],
23      "totalAmount": 49.9,
24      "status": "Pendente",
25      "orderDate": "2025-05-28T18:03:21.969Z",
26      "_v": 0
27    }
28  ]
29 }
```

GET `_.base_url` /orders/user/683a3458848708bda9c6fa0e Send 200 OK 491 ms 93 B

Params Body Auth Headers (2) Scripts Docs Preview Headers (9) Cookies Tests 0/0 → Mock Console

No Body

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

GET `_.base_url` /orders/user/683a3 Send 404 Not Found 281 ms 39 B

Params Body Auth Headers (2) Scripts Docs Preview Headers (9) Cookies Tests 0/0 → Mock Console

No Body

```
1 {
2   "message": "Usuário não encontrado."
3 }
```