

a) Para o model Cliente:

Um print demonstrando o resultado da criação de um cliente, com código de retorno 201. Um dos clientes deve ter o nome completo do(a) aluno(a).

POST http://localhost:8080/clientes

Status: 201 Created Size: 0 Bytes Time: 1.21 s

Response

```
1 {
2   "nome": "Wilton Monteiro Resende",
3   "cpf": "12345678900",
4   "data_nascimento": "1988-06-15T00:00:00Z",
5   "email": "wilton.resende@example.com",
6   "logradouro": "Rua das Flores",
7   "num_casa": "123",
8   "complemento": "Apto 101",
9   "bairro": "Centro",
10  "municipio": "Piracicaba",
11  "uf": "SP",
12 }
```

Terminal

```
PARAMS: []
db.Cliente.aggregate([ { $match: { $expr: { $and: [ { $and: [ { $eq: [ "$_id", { $literal: ObjectId("67031a878ab9283ccc65041c") }, ], ], { $ne: [ "$_id", "$REMOVE" ], }, ], }, ], }, }, { $project: { _id: 1, nome: 1, cpf: 1, data_nascimento: 1, email: 1, logradouro: 1, num_casa: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1 }, }, ] )
PARAMS: []
POST /clientes 201 1186.028 ms - -
```

Um print demonstrando, pelo menos, dois clientes cadastrados, com código de retorno 200.

Cliente 1

GET http://localhost:8080/clientes

Status: 200 OK Size: 1.3 KB Time: 52 ms

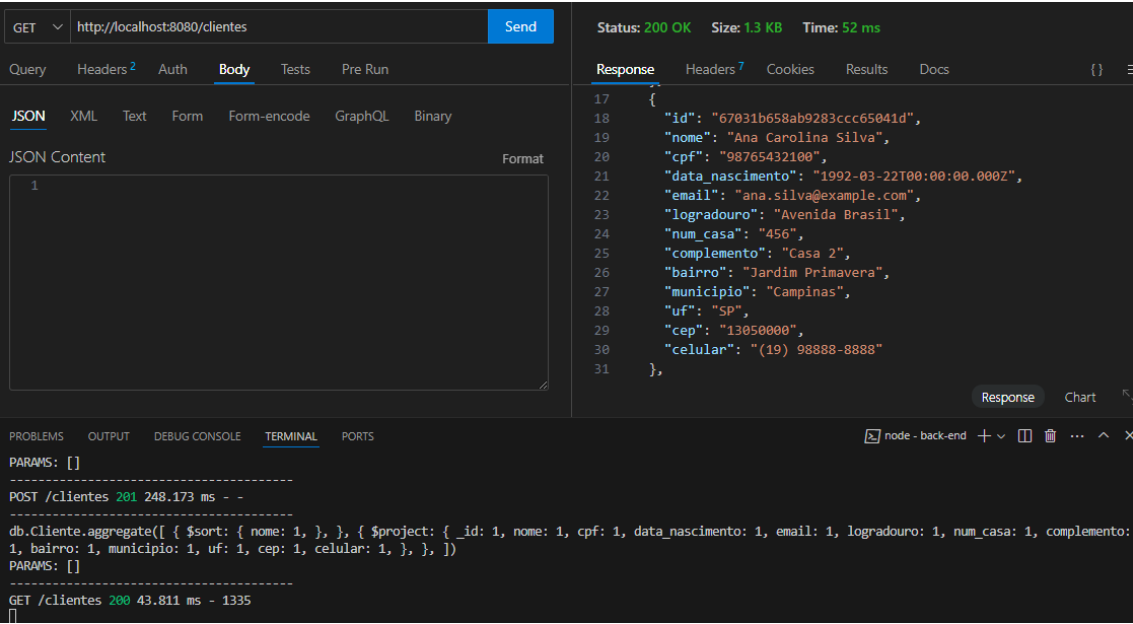
Response

```
1 [
2   {
3     "id": "66df93d55ff365d84f486520",
4     "nome": "Abelina de Abreu",
5     "cpf": "012.345.678-90",
6     "data_nascimento": null,
7     "email": "abelina.abreu@bol.com.br",
8     "logradouro": "Av. Presidente Vargas",
9     "num_casa": "2340",
10    "complemento": "Ap. 206",
11    "bairro": "Cidade Nova",
12    "municipio": "Franca",
13    "uf": "SP",
14    "cep": "14401-060",
15    "celular": "(16) 90123-4567"
16  },
17  {
18    "id": "66df93d55ff365d84f486520",
19    "nome": "Abelina de Abreu",
20    "cpf": "012.345.678-90",
21    "data_nascimento": null,
22    "email": "abelina.abreu@bol.com.br",
23    "logradouro": "Av. Presidente Vargas",
24    "num_casa": "2340",
25    "complemento": "Ap. 206",
26    "bairro": "Cidade Nova",
27    "municipio": "Franca",
28    "uf": "SP",
29    "cep": "14401-060",
30    "celular": "(16) 90123-4567"
31  }
32 ]
```

Terminal

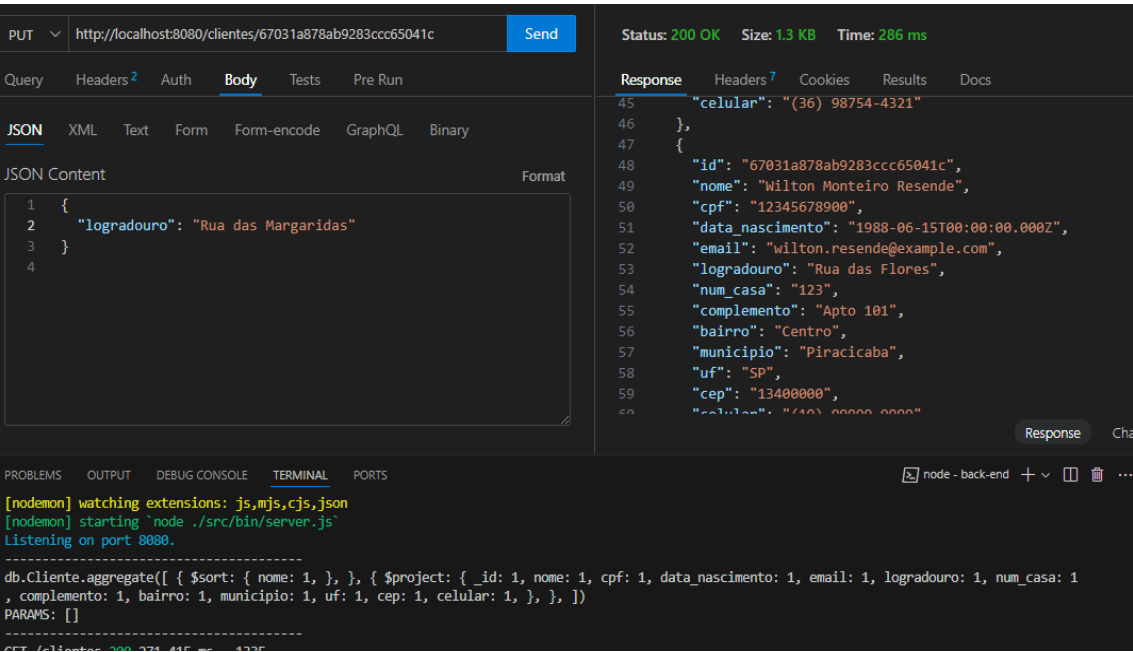
```
PARAMS: []
db.Cliente.aggregate([ { $sort: { nome: 1 }, }, { $project: { _id: 1, nome: 1, cpf: 1, data_nascimento: 1, email: 1, logradouro: 1, num_casa: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1 }, }, ] )
PARAMS: []
GET /clientes 200 43.811 ms - 1335
```

Cliente 2



Um print demonstrando a operação de atualização de um cliente, com código de retorno 204.

Antes do update:



Execução do put alterando o logradouro do cliente selecionado pelo Id:

PUT

http://localhost:8080/clientes/67031a878ab9283ccc65041c

Send

Status: 204 No Content

Size: 0 Bytes

Time: 121 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1 {

2 "logradouro": "Rua das Margaridas"

3 }

4 }

Response

1

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

+

⌵

🗑

PARAMS: []

db.Cliente.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("67031a878ab9283ccc65041c") },],], { \$ne: ["\$_id", "\$REMOVE"], },], },], }, }, { \$project: { _id: 1, nome: 1, cpf: 1, data_nascimento: 1, email: 1, logradouro: 1, num_casa: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, },])

PARAMS: []

PUT /clientes/67031a878ab9283ccc65041c 204 115.946 ms - -

⏏

Depois do update:

GET

http://localhost:8080/clientes/67031a878ab9283ccc65041c

Send

Status: 200 OK

Size: 342 Bytes

Time: 33 ms

Query

Headers 2

Auth

Body 1

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1 {

2 "logradouro": "Rua das Margaridas"

3 }

4 }

Response

1 {

2 "id": "67031a878ab9283ccc65041c",

3 "nome": "Wilton Monteiro Resende",

4 "cpf": "12345678900",

5 "data_nascimento": "1988-06-15T00:00:00.000Z",

6 "email": "wilton.resende@example.com",

7 "logradouro": "Rua das Margaridas",

8 "num_casa": "123",

9 "complemento": "Apto 101",

10 "bairro": "Centro",

11 "municipio": "Piracicaba",

12 "uf": "SP",

13 "cep": "13400000",

14 "celular": "(19) 99999-9999"

15 }

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

+

⌵

🗑

PUT /clientes/67031a878ab9283ccc65041c 204 115.946 ms - -

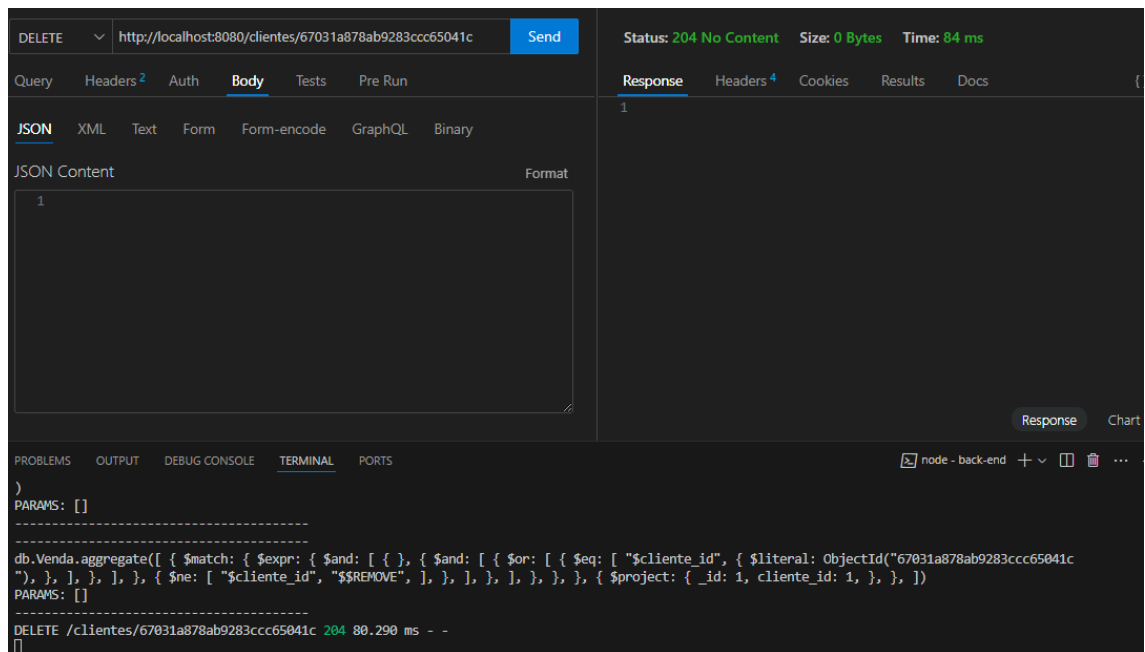
db.Cliente.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("67031a878ab9283ccc65041c") },],], { \$ne: ["\$_id", "\$REMOVE"], },], },], }, }, { \$project: { _id: 1, nome: 1, cpf: 1, data_nascimento: 1, email: 1, logradouro: 1, num_casa: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, },])

PARAMS: []

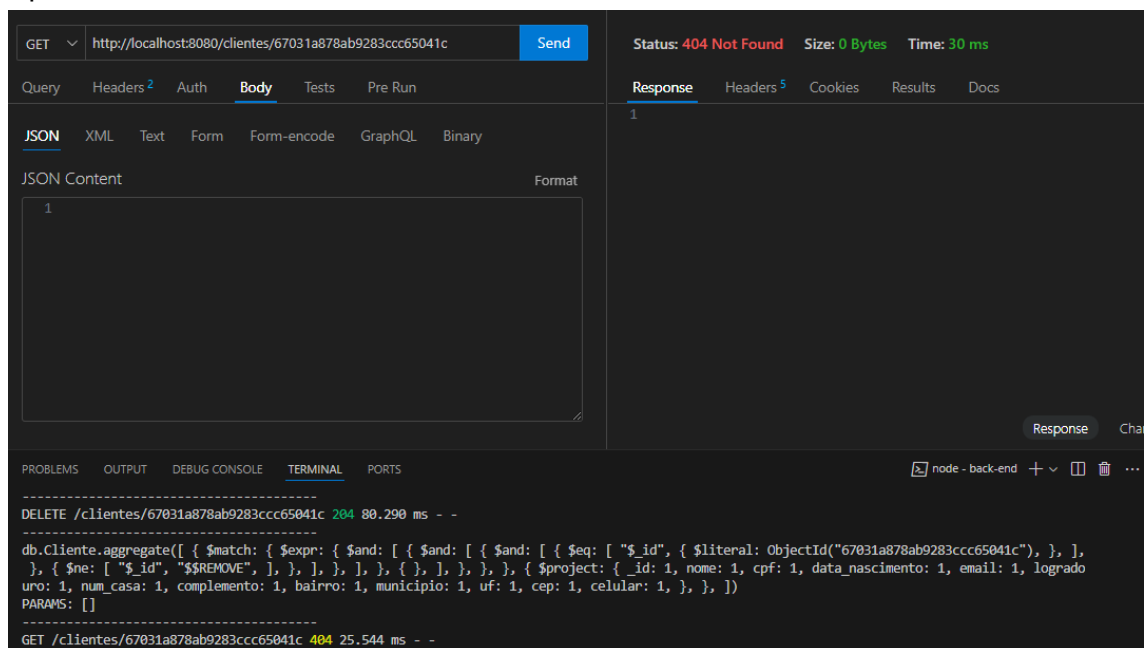
GET /clientes/67031a878ab9283ccc65041c 200 28.590 ms - 342

⏏

Um print demonstrando a operação de exclusão de um cliente, com código de retorno 204.

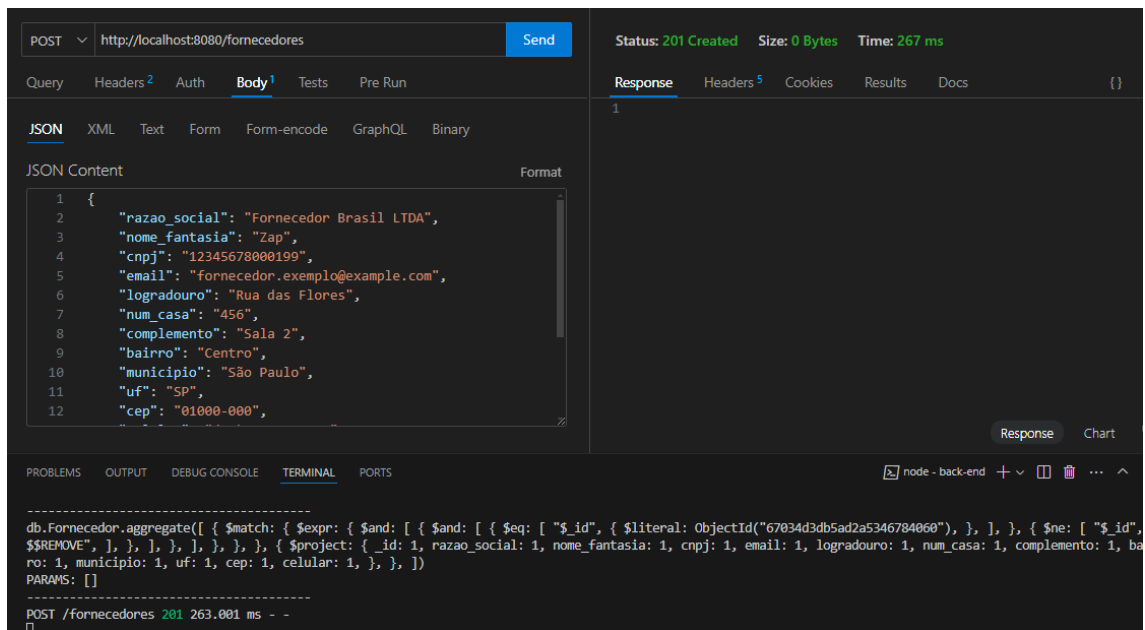


Após a exclusão



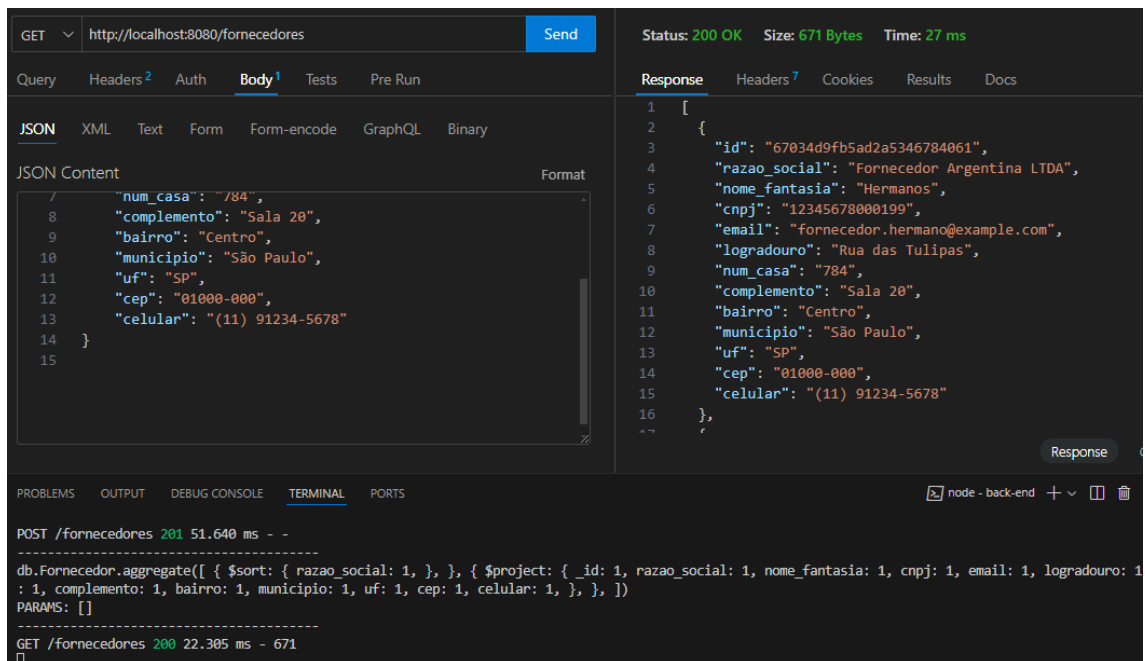
b) Para o model Fornecedor:

Um print demonstrando o resultado da criação de um fornecedor, com código de retorno 201.



Um print demonstrando, pelo menos, dois fornecedores cadastrados, com código de retorno 200.

Fornecedor 1



Fornecedor 2

```
GET http://localhost:8080/fornecedores Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

/
8  "num_casa": "784",
9  "complemento": "Sala 20",
10 "bairro": "Centro",
11 "municipio": "São Paulo",
12 "uf": "SP",
13 "cep": "01000-000",
14 "celular": "(11) 91234-5678"
15 }

Status: 200 OK Size: 671 Bytes Time: 27 ms

Response Headers 7 Cookies Results Docs

17 {
18   "id": "67034d3db5ad2a5346784060",
19   "razao_social": "Fornecedor Brasil LTDA",
20   "nome_fantasia": "Zap",
21   "cnpj": "12345678000199",
22   "email": "fornecedor.exemplo@example.com",
23   "logradouro": "Rua das Flores",
24   "num_casa": "456",
25   "complemento": "Sala 2",
26   "bairro": "Centro",
27   "municipio": "São Paulo",
28   "uf": "SP",
29   "cep": "01000-000",
30   "celular": "(11) 91234-5678"
31 }
32 ]

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node - back-end + v []

POST /fornecedores 201 51.640 ms - -
-----
db.Fornecedor.aggregate([ { $sort: { razao_social: 1, }, }, { $project: { _id: 1, razao_social: 1, nome_fantasia: 1, cnpj: 1, email: 1, logradouro: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, }, ])
PARAMS: []
-----
GET /fornecedores 200 22.305 ms - 671
```

Um print demonstrando a operação de atualização de um fornecedor, com código de retorno 204.

Antes do update:

```
PUT http://localhost:8080/fornecedores Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

1 {
2   "num_casa": "750"
3 }
4

Status: 200 OK Size: 671 Bytes Time: 27 ms

Response Headers 7 Cookies Results Docs

17 {
18   "id": "67034d3db5ad2a5346784060",
19   "razao_social": "Fornecedor Brasil LTDA",
20   "nome_fantasia": "Zap",
21   "cnpj": "12345678000199",
22   "email": "fornecedor.exemplo@example.com",
23   "logradouro": "Rua das Flores",
24   "num_casa": "456",
25   "complemento": "Sala 2",
26   "bairro": "Centro",
27   "municipio": "São Paulo",
28   "uf": "SP",
29   "cep": "01000-000",
30   "celular": "(11) 91234-5678"
31 }
32 ]

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node - back-end + v []

POST /fornecedores 201 51.640 ms - -
-----
db.Fornecedor.aggregate([ { $sort: { razao_social: 1, }, }, { $project: { _id: 1, razao_social: 1, nome_fantasia: 1, cnpj: 1, email: 1, logradouro: 1, complemento: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, }, ])
PARAMS: []
-----
GET /fornecedores 200 22.305 ms - 671
```

Execução do put alterando o num_casa de fornecedor selecionado pelo Id:

PUT

http://localhost:8080/fornecedores/67034d9fb5ad2a5346784061

Send

Query

Headers²

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1 {

2 "num_casa": "750"

3 }

4

Status: 204 No Content

Size: 0 Bytes

Time: 95 ms

Response

Headers⁴

Cookies

Results

Docs

1

Response

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

PARAMS: []

db.Fornecedor.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("67034d9fb5ad2a5346784061") },], }, { \$ne: ["\$_id", "\$\$REMOVE"], },], },], }, { \$project: { _id: 1, razao_social: 1, nome_fantasia: 1, cnpj: 1, email: 1, logradouro: 1, num_casa: 1, complemento: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, },])

PARAMS: []

PUT /fornecedores/67034d9fb5ad2a5346784061 204 91.662 ms - -

Depois do update:

GET

http://localhost:8080/fornecedores/67034d9fb5ad2a5346784061

Send

Query

Headers²

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1 {

2 "num_casa": "750"

3 }

4

Status: 200 OK

Size: 339 Bytes

Time: 33 ms

Response

Headers⁷

Cookies

Results

Docs

1 {

2 "id": "67034d9fb5ad2a5346784061",

3 "razao_social": "Fornecedor Argentina LTDA",

4 "nome_fantasia": "Hermanos",

5 "cnpj": "12345678000199",

6 "email": "fornecedor.hermano@example.com",

7 "logradouro": "Rua das Tulipas",

8 "num_casa": "750",

9 "complemento": "Sala 20",

10 "bairro": "Centro",

11 "municipio": "São Paulo",

12 "uf": "SP",

13 "cep": "01000-000",

14 "celular": "(11) 91234-5678"

15 }

Response

Chart

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

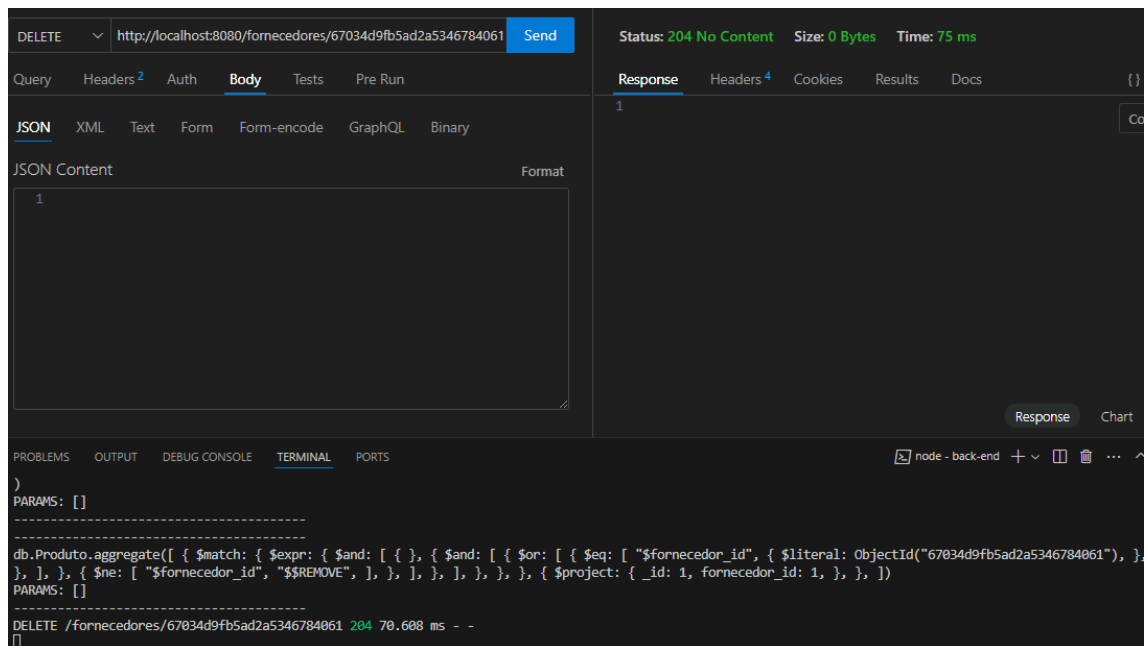
PUT /fornecedores/67034d9fb5ad2a5346784061 204 91.662 ms - -

db.Fornecedor.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("67034d9fb5ad2a5346784061") },], }, { \$ne: ["\$_id", "\$\$REMOVE"], },], },], }, { \$project: { _id: 1, razao_social: 1, nome_fantasia: 1, cnpj: 1, email: 1, logradouro: 1, num_casa: 1, bairro: 1, municipio: 1, uf: 1, cep: 1, celular: 1, }, },])

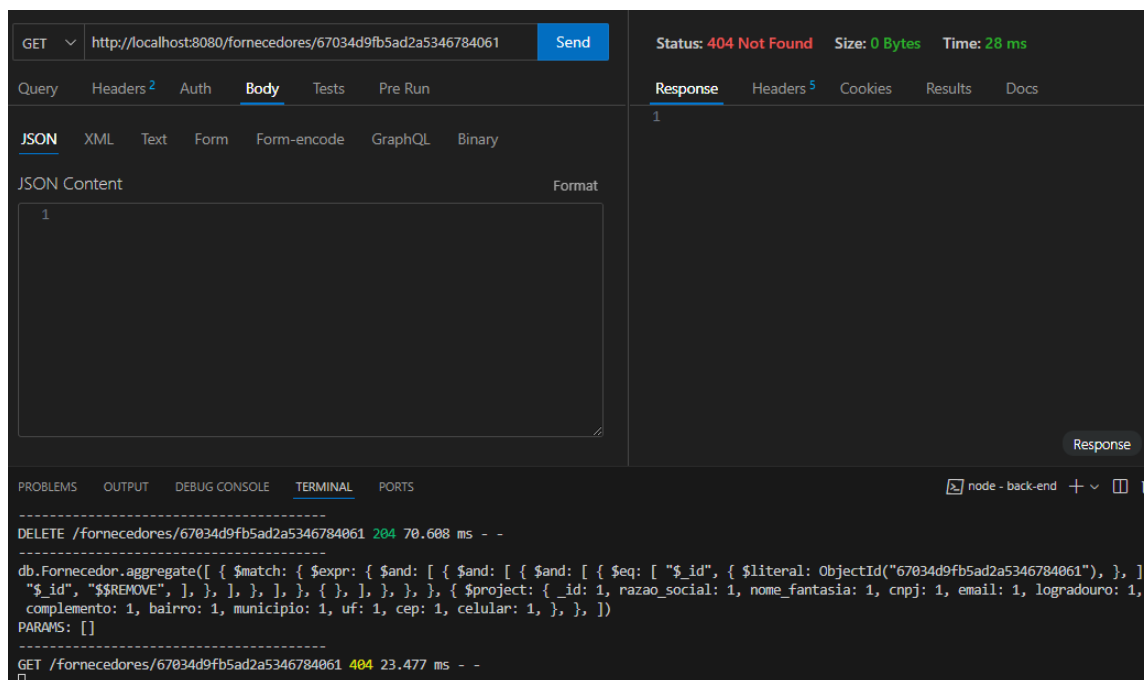
PARAMS: []

GET /fornecedores/67034d9fb5ad2a5346784061 200 28.743 ms - 339

Um print demonstrando a operação de exclusão de um fornecedor, com código de retorno 204.

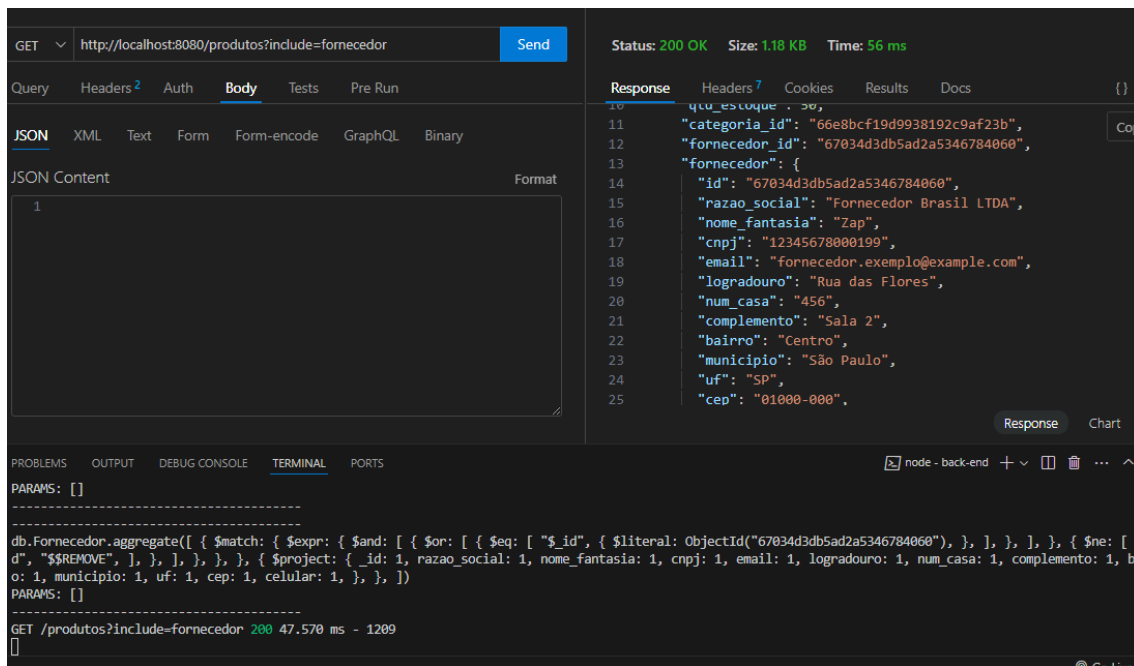


Após a exclusão



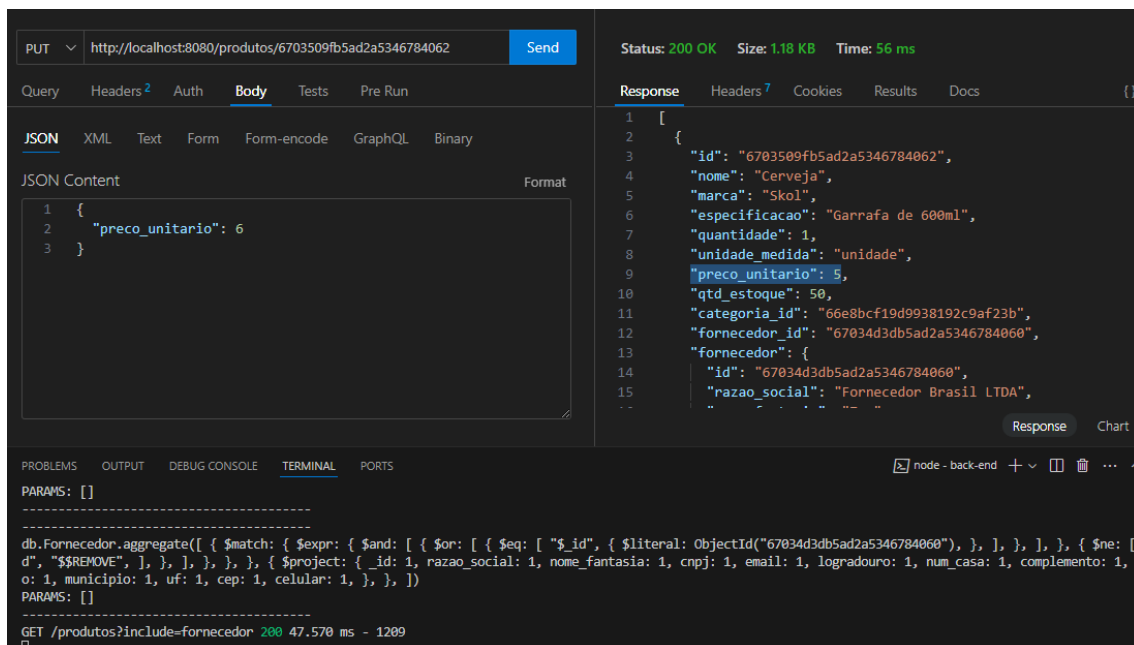
c) Para o model Produto:

Um print demonstrando o resultado da criação de um produto, com código de retorno 201.



Um print demonstrando a operação de atualização de um produto,
com código de retorno 204

Antes do update:



Execução do put alterando o preco-unitario de produto selecionado pelo Id:

PUT `http://localhost:8080/produtos/6703509fb5ad2a5346784062` **Send**

Query Headers² Auth **Body¹** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "preco_unitario": 6
3 }
```

Status: **204 No Content** Size: **0 Bytes** Time: **113 ms**

Response Headers⁴ Cookies Results Docs

1

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

node - back-end + v [] ..

PARAMS: []

db.Produto.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("6703509fb5ad2a5346784062") },], }, { \$ne: ["\$_id", "\$\$REMOVE"], },], }, }, }, { \$project: { _id: 1, nome: 1, marca: 1, especificacao: 1, quantidade: 1, unidade_medida: 1, preco_unitario: 1, qtd_estoque: 1, categoria_id: 1, fornecedor_id: 1, }, },]

PARAMS: []

PUT /produtos/6703509fb5ad2a5346784062 **204** 109.794 ms - -

Depois do update:

GET `http://localhost:8080/produtos/6703509fb5ad2a5346784062` **Send**

Query Headers² Auth **Body¹** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "preco_unitario": 6
3 }
```

Status: **200 OK** Size: **263 Bytes** Time: **27 ms**

Response Headers⁷ Cookies Results Docs

```
1 {
2   "id": "6703509fb5ad2a5346784062",
3   "nome": "Cerveja",
4   "marca": "Skol",
5   "especificacao": "Garrafa de 600ml",
6   "quantidade": 1,
7   "unidade_medida": "unidade",
8   "preco_unitario": 6,
9   "qtd_estoque": 50,
10  "categoria_id": "66e8bcf19d9938192c9af23b",
11  "fornecedor_id": "67034d3db5ad2a5346784060"
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

node - back-end + v []

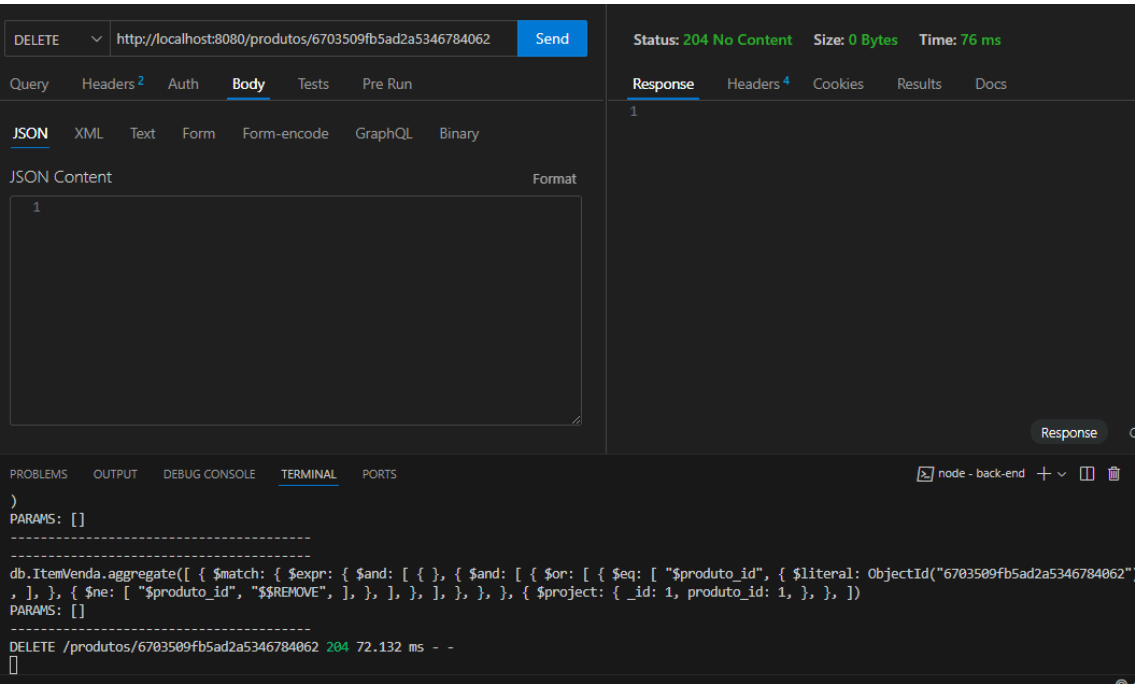
PUT /produtos/6703509fb5ad2a5346784062 **204** 109.794 ms - -

db.Produto.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("6703509fb5ad2a5346784062") },], }, { \$ne: ["\$_id", "\$\$REMOVE"], },], }, }, }, { \$project: { _id: 1, nome: 1, marca: 1, especificacao: 1, quantidade: 1, unidade_medida: 1, qtd_estoque: 1, categoria_id: 1, fornecedor_id: 1, }, },]

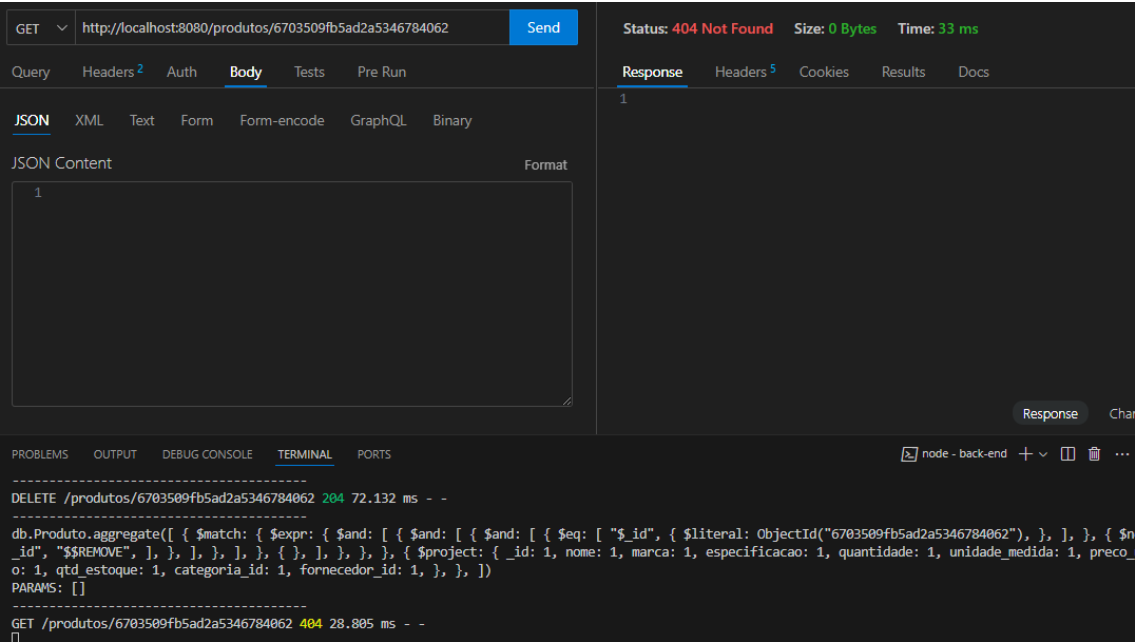
PARAMS: []

GET /produtos/6703509fb5ad2a5346784062 **200** 23.895 ms - 263

Um print demonstrando a operação de exclusão de um produto, com código de retorno 204

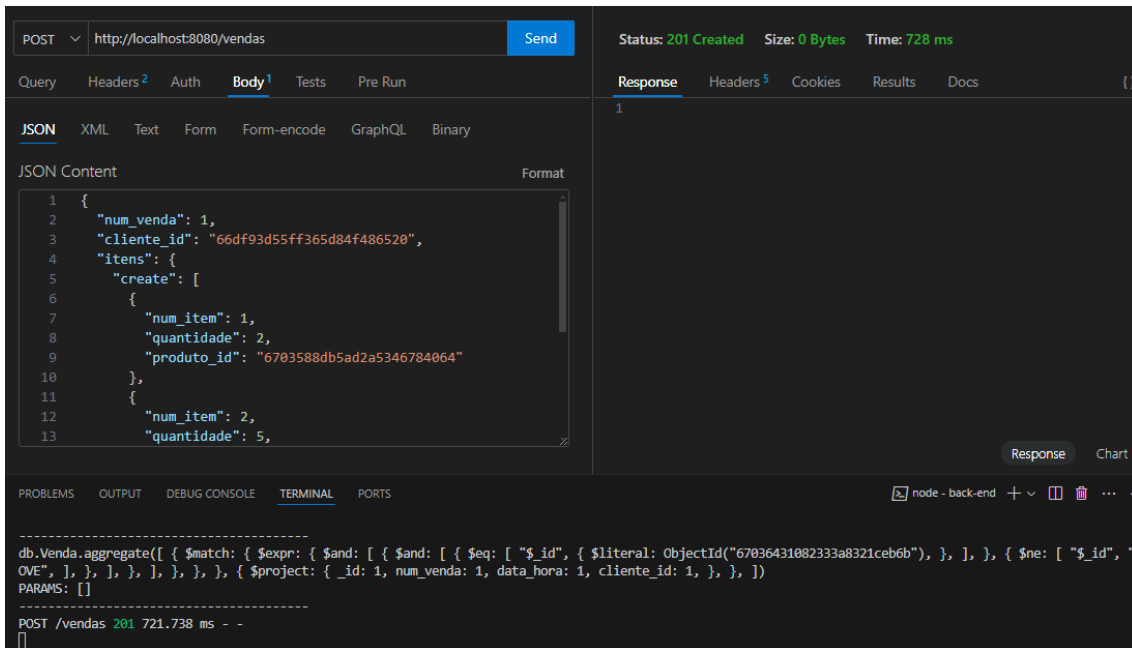


Após a exclusão:



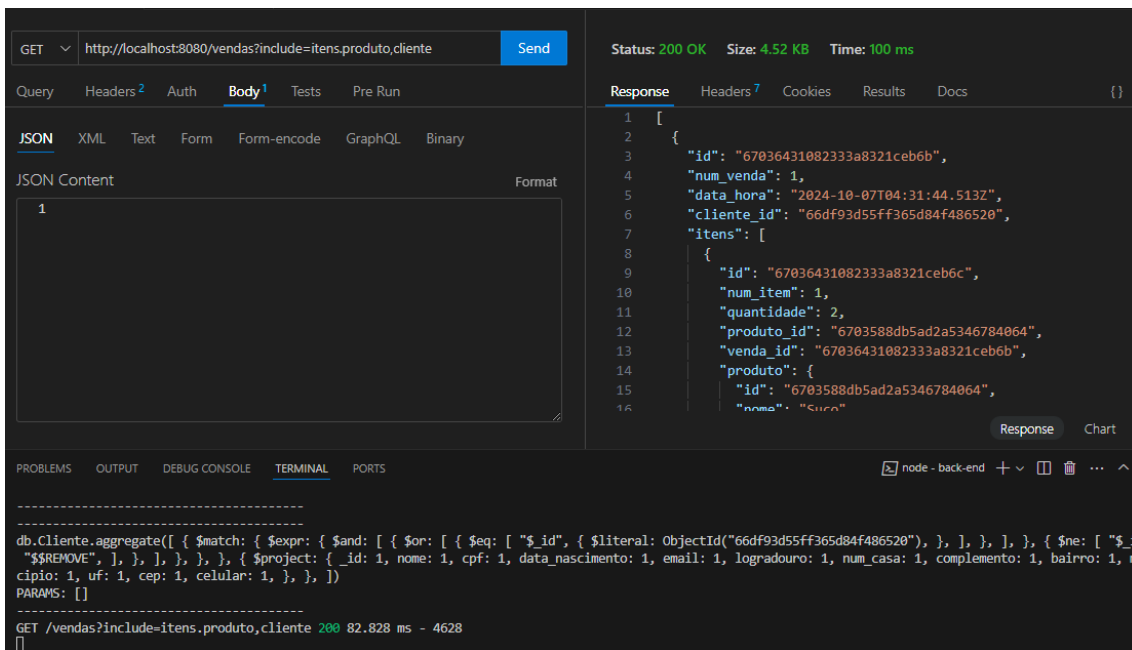
d) Para o model Venda:

Um print demonstrando o resultado da criação de uma venda, com pelo menos dois itens cada uma, com código de retorno 201.



Um print demonstrando, pelo menos, duas vendas cadastradas, com cliente e produto populados, e código de retorno 200

Venda 1



Antes do update:

GET

http://localhost:8080/vendas/66f1ec3c2346e99998276d0e

Send

Query

Headers²

Auth

Body

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1

{

2

"data_hora": "2024-10-06T04:31:44.513Z"

3

}

Response

Headers⁷

Cookies

Results

Docs

2

{

3

"id": "66f1ec3c2346e99998276d0e",

4

"num_venda": 2,

5

"data_hora": "2024-09-23T22:31:24.548Z",

6

"cliente_id": "66df93d55ff365d84f486520"

7

},

8

{

9

"id": "67036431082333a8321ceb6b",

10

"num_venda": 1,

11

"data_hora": "2024-10-07T04:31:44.513Z",

12

"cliente_id": "66df93d55ff365d84f486520"

13

},

14

{

15

"id": "67036464082333a8321ceb6e",

16

"num_venda": 2,

17

"data_hora": "2024-10-07T04:32:36.875Z",

18

"cliente_id": "66df93d55ff365d84f486520"

19

}

Response

Chart

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

+

⌵

🗑

⋮

PUT /vendas 404 1.098 ms - 145

db.Venda.aggregate([{ \$sort: { data_hora: 1, }, }, { \$project: { _id: 1, num_venda: 1, data_hora: 1, cliente_id: 1, }, },])

PARAMS: []

GET /vendas 200 22.724 ms - 382

⌵

Execução do put alterando a data_hora de vendas selecionado pelo Id:

PUT

http://localhost:8080/vendas/66f1ec3c2346e99998276d0e

Send

Query

Headers²

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1

{

2

"data_hora": "2024-10-06T04:31:44.513Z"

3

}

Response

Headers⁴

Cookies

Results

Docs

1

Response

Cha

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

node - back-end

+

⌵

🗑

⋮

db.Venda.aggregate([{ \$match: { \$expr: { \$and: [{ \$and: [{ \$eq: ["\$_id", { \$literal: ObjectId("66f1ec3c2346e99998276d0e") },], }, { \$ne: ["\$_id", "OVE",], },], },], }, { \$project: { _id: 1, num_venda: 1, data_hora: 1, cliente_id: 1, }, },])

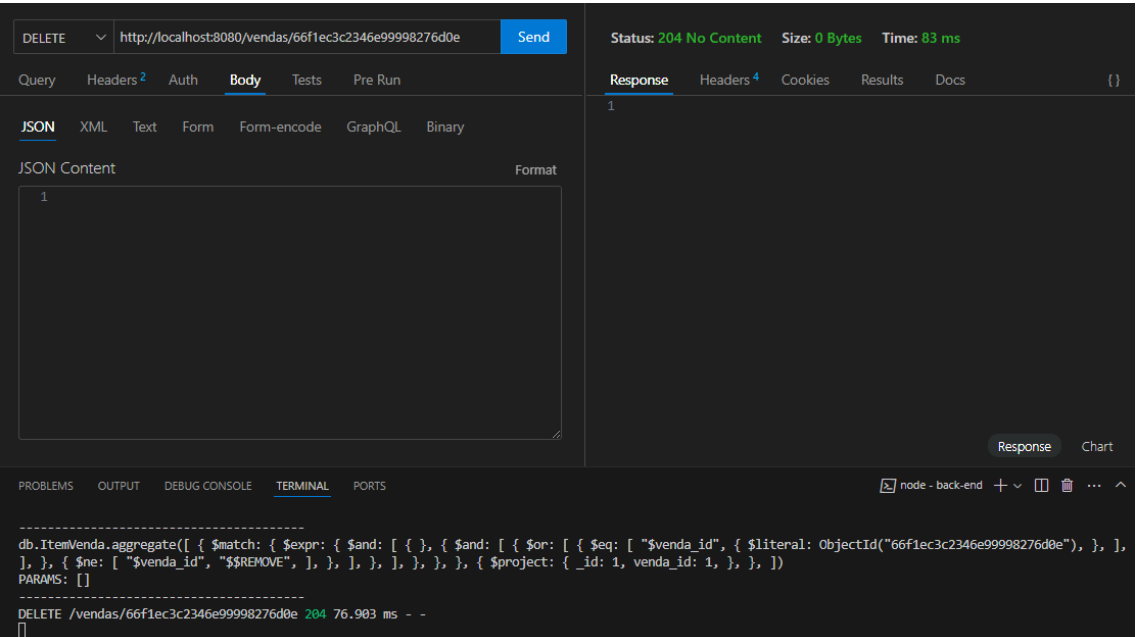
PARAMS: []

PUT /vendas/66f1ec3c2346e99998276d0e 204 360.729 ms - -

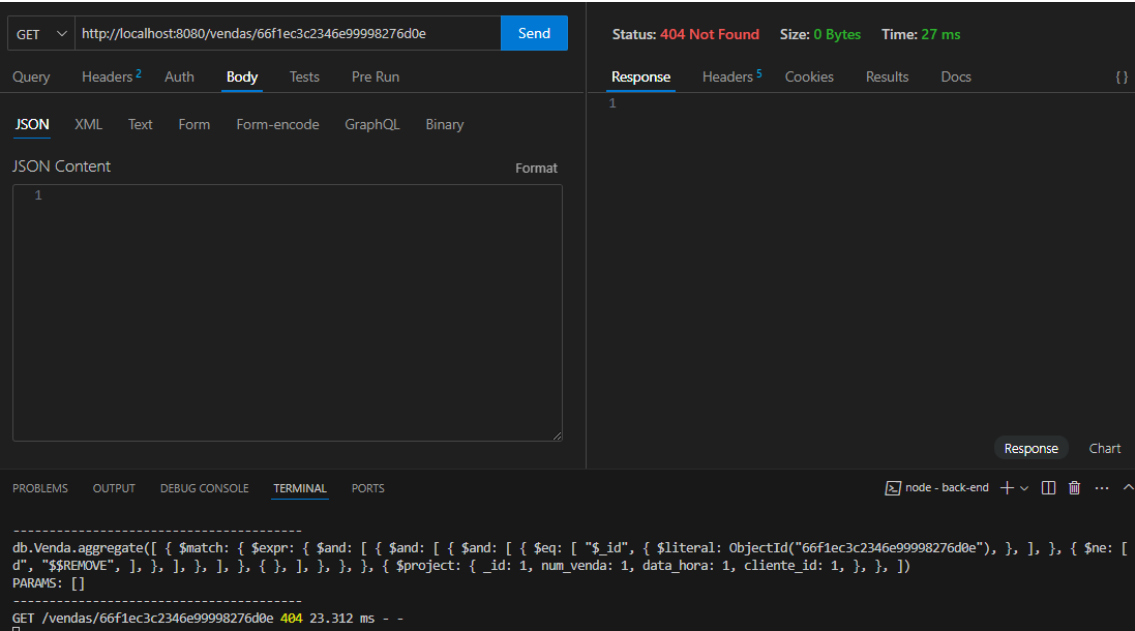
⌵

Depois do update:

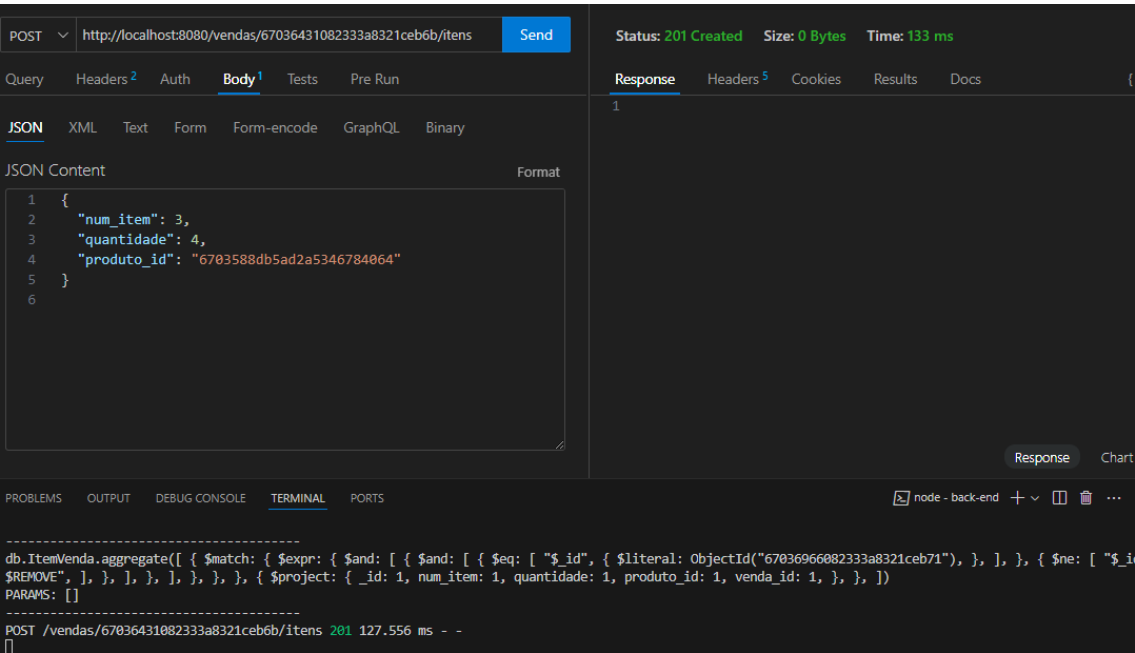
Um print demonstrando a operação de exclusão de uma venda, com código de retorno 204.



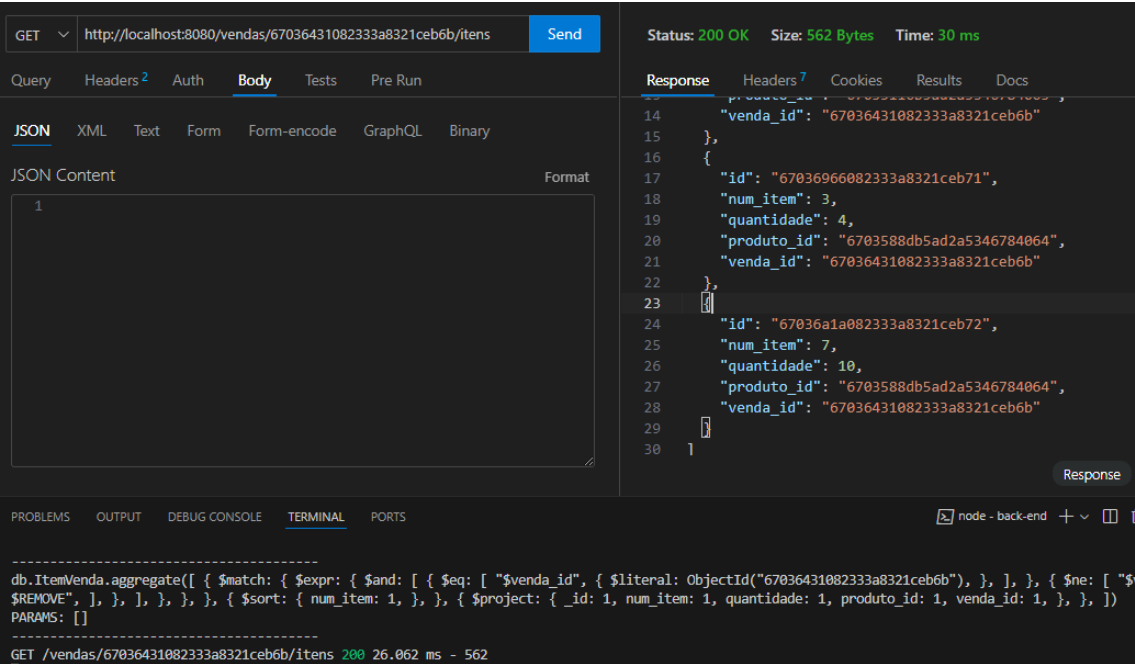
Após a exclusão:



Um print demonstrando a operação de inserção de um novo item em uma venda já existente, com código de retorno 201.

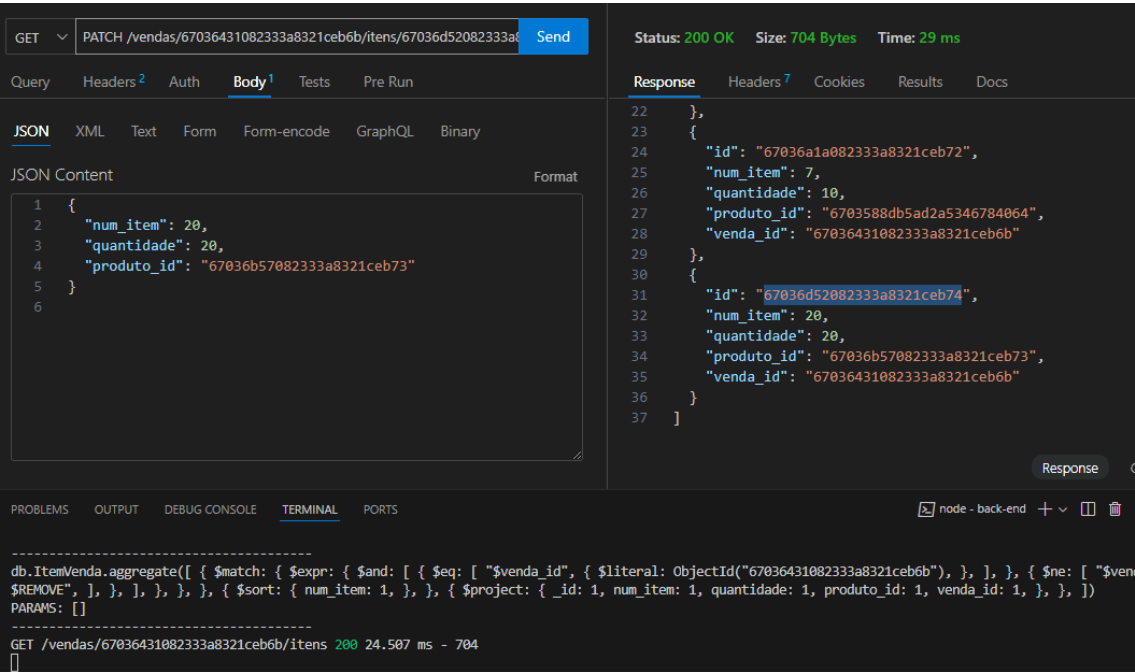


Após a inserção

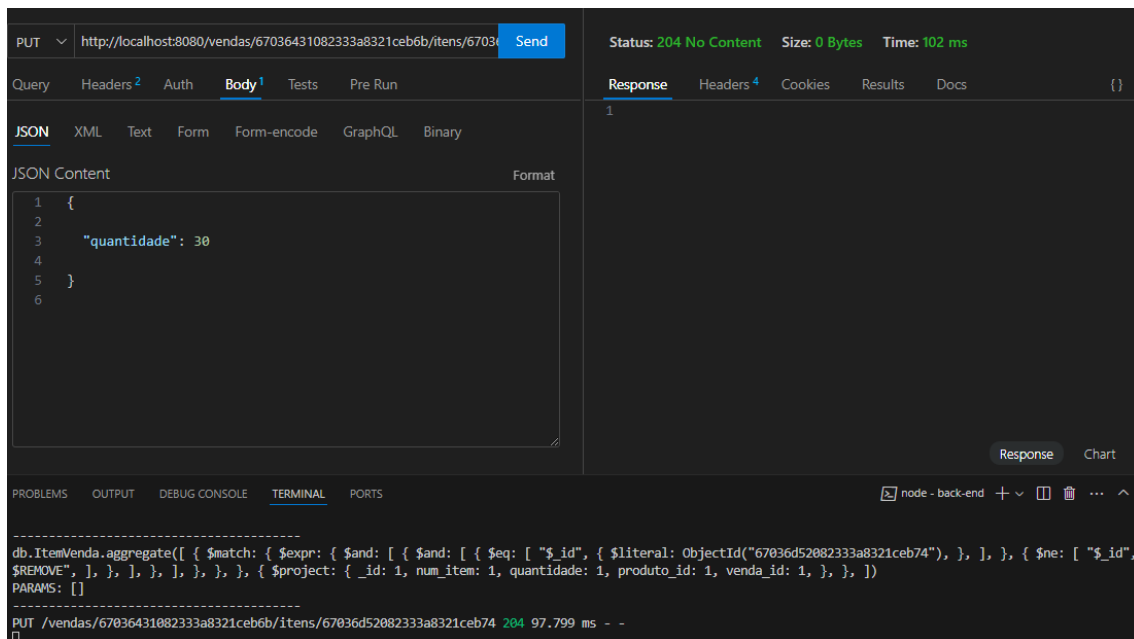


Um print demonstrando a operação de atualização de um item da venda, com código de retorno 204.

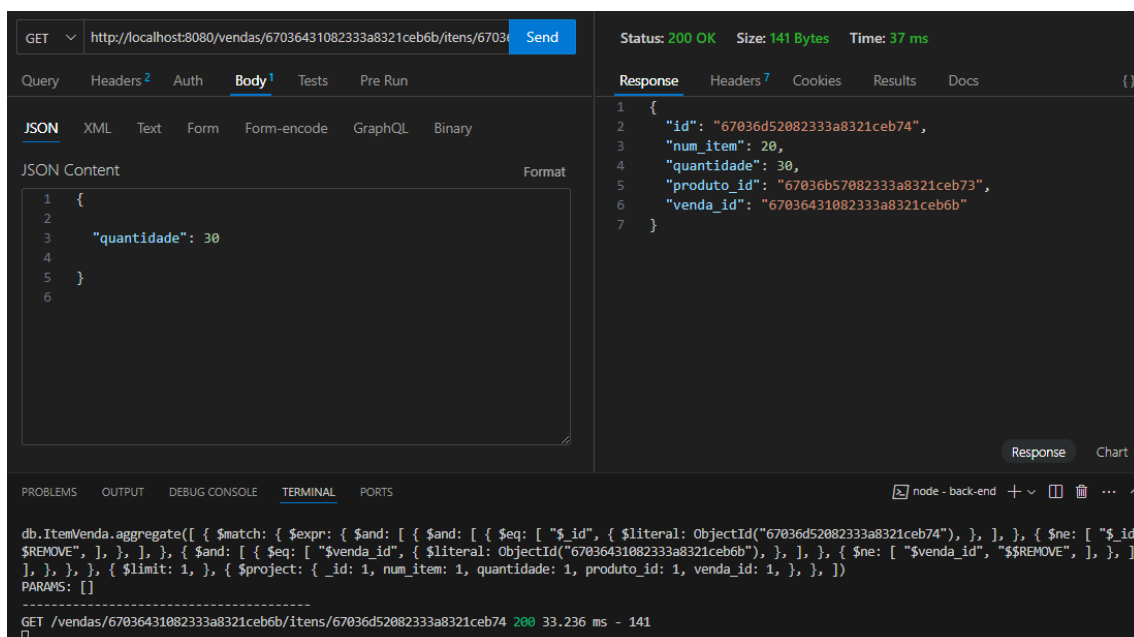
Antes do update:



Execução do put alterando a quantidade de um item de vendas selecionado pelo Id:



Depois do update:



Um print demonstrando a operação de inserção de um novo item em uma venda já existente, com código de retorno 201.

Antes da inserção:

POST http://localhost:8080/vendas/67036431082333a8321ceb6b/itens/ Send

Status: 200 OK Size: 844 Bytes Time: 32 ms

Query Headers Auth **Body** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "num_item": 7,
3   "quantidade": 38,
4   "produto_id": "67036b57082333a8321ceb73"
5 }
```

Response Headers Cookies Results Docs

```
34   "produto_id": "6703588db5ad2a5346784064",
35   "venda_id": "67036431082333a8321ceb6b"
36 },
37 {
38   "id": "67036d52082333a8321ceb74",
39   "num_item": 20,
40   "quantidade": 30,
41   "produto_id": "67036b57082333a8321ceb73",
42   "venda_id": "67036431082333a8321ceb6b"
43 }
44 ]
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
GET /vendas 200 26.279 ms - 255
db.ItemVenda.aggregate([ { $match: { $expr: { $and: [ { $eq: [ "$venda_id", { $literal: ObjectId("67036431082333a8321ceb6b") }, ], }, { $ne: [ "$venda_id", { $literal: ObjectId("67036431082333a8321ceb6b") }, ], }, ], }, }, { $sort: { num_item: 1, }, }, { $project: { _id: 1, num_item: 1, quantidade: 1, produto_id: 1, venda_id: 1, }, }, ])
PARAMS: []
GET /vendas/67036431082333a8321ceb6b/itens/ 200 27.552 ms - 844
```

Execução da inserção:

POST http://localhost:8080/vendas/67036431082333a8321ceb6b/itens/ Send

Status: 201 Created Size: 0 Bytes Time: 72 ms

Query Headers Auth **Body** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "num_item": 7,
3   "quantidade": 38,
4   "produto_id": "67036b57082333a8321ceb73"
5 }
```

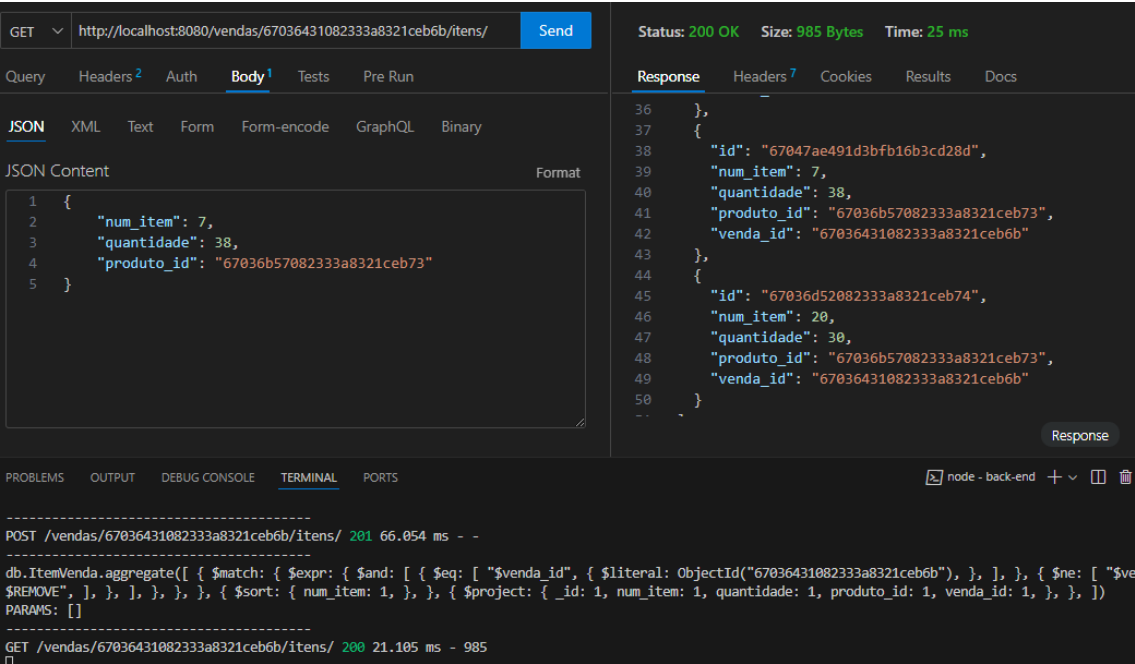
Response Headers Cookies Results Docs

```
1
```

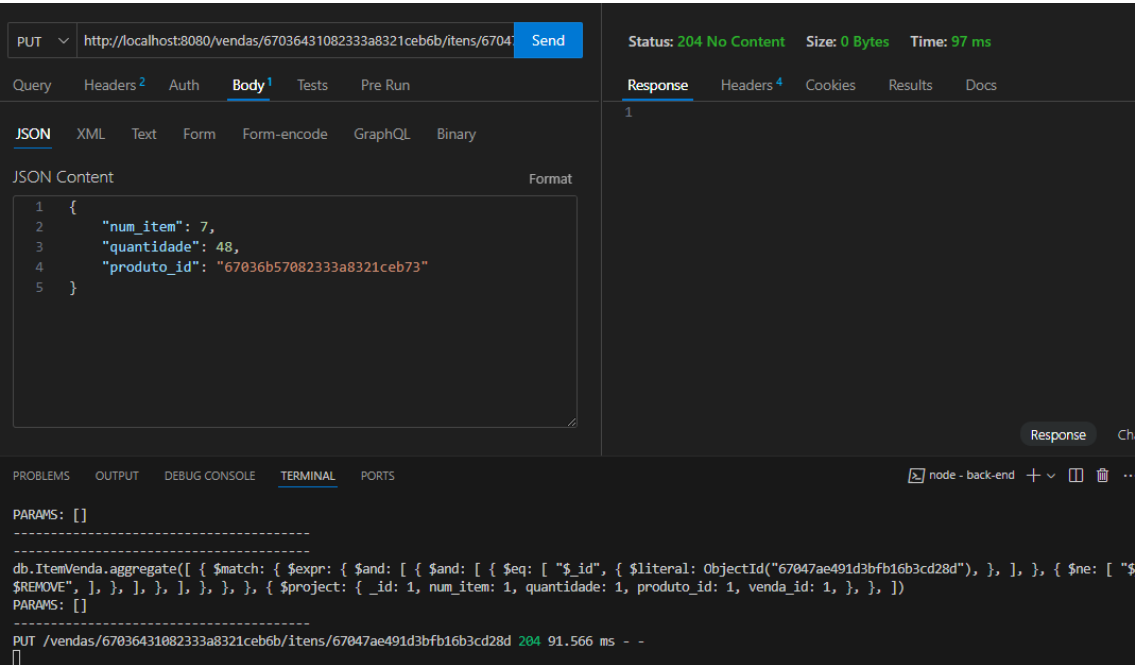
PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PARAMS: []
db.ItemVenda.aggregate([ { $match: { $expr: { $and: [ { $and: [ { $eq: [ "$_id", { $literal: ObjectId("67047ae491d3bfb16b3cd28d") }, ], }, { $ne: [ "$_id", { $literal: ObjectId("67047ae491d3bfb16b3cd28d") }, ], }, ], }, }, { $sort: { _id: 1, num_item: 1, quantidade: 1, produto_id: 1, venda_id: 1, }, }, { $project: { _id: 1, num_item: 1, quantidade: 1, produto_id: 1, venda_id: 1, }, }, ])
PARAMS: []
POST /vendas/67036431082333a8321ceb6b/itens/ 201 66.054 ms - -
```

Após a inserção:



Um print demonstrando a operação de atualização de um item da venda, com código de retorno 204.



Após a atualização

TC New Request X JS vendas.js M

GET http://localhost:8080/vendas/67036431082333a8321ceb6b/itens/67047ae491d3bfb16b3cd28d Send

Status: 200 OK Size: 140 Bytes Time: 33 ms

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "num_item": 7,
3   "quantidade": 48,
4   "produto_id": "67036b57082333a8321ceb73"
5 }
```

Response Headers 7 Cookies Results Docs

Response Chart

```
1 {
2   "id": "67047ae491d3bfb16b3cd28d",
3   "num_item": 7,
4   "quantidade": 48,
5   "produto_id": "67036b57082333a8321ceb73",
6   "venda_id": "67036431082333a8321ceb6b"
7 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node - back-end + -

```
PUT /vendas/67036431082333a8321ceb6b/itens/67047ae491d3bfb16b3cd28d 204 91.566 ms - -
-----
db.ItemVenda.aggregate([ { $match: { $expr: { $and: [ { $and: [ { $eq: [ "$_id", { $literal: ObjectId("67047ae491d3bfb16b3cd28d") }, ], }, { $ne: [ "$_id", "$REMOVE" ], }, ], }, }, { $and: [ { $eq: [ "$venda_id", { $literal: ObjectId("67036431082333a8321ceb6b") }, ], }, { $ne: [ "$venda_id", "$REMOVE" ], }, ], }, ], }, { $limit: 1 }, { $project: { _id: 1, num_item: 1, quantidade: 1, produto_id: 1, venda_id: 1, }, }, ])
PARAMS: []
-----
GET /vendas/67036431082333a8321ceb6b/itens/67047ae491d3bfb16b3cd28d 200 26.976 ms - 140
```