

shoppyglobe



localhost:27017 > shoppyglobe

> Open MongoDB shell

+ Create collection

Refresh

Sort by

Collection Name



View



products

Storage size:
20.48 kB

Documents:
10

Avg. document size:
154.00 B

Indexes:
1

Total index size:
20.48 kB

users

Storage size:
20.48 kB

Documents:
2

Avg. document size:
109.00 B

Indexes:
1

Total index size:
36.86 kB

Initial collections, Cart is inside User Collection

GET

▼

http://localhost:5555/products

Send

Query

Headers²

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK Size: 1.62 KB Time: 65 ms

Response Headers⁶ Cookies Results Docs

1

[

2

{

3

"_id": "6737ab70a4ce746810d8b2b8",

4

"name": "Smartphone",

5

"price": 699,

6

"description": "A high-quality smartphone with a stunning display and

powerful processor.",

7

"stock": 50

8

},

9

{

10

"_id": "6737ab70a4ce746810d8b2b9",

11

"name": "Laptop",

12

"price": 1200,

13

"description": "A sleek laptop with powerful performance, perfect for

work and entertainment.",

14

"stock": 30

15

},

16

{

17

"_id": "6737ab70a4ce746810d8b2ba",

18

"name": "Wireless Earbuds",

19

"price": 150,

20

"description": "Compact wireless earbuds with noise cancellation and

long battery life.",

21

"stock": 100

22

},

23

[

Get all products

GEThttp://localhost:5555/products/6737ab70a4ce746810d8b2b9

Send

Query

Headers²

Auth

Body

Tests

Pre Run

Query Parameters

parameter

value

Status: 200 OK Size: 168 Bytes Time: 13 ms

Response

Headers⁶

Cookies

Results

Docs

{ }

≡

1

{

2

"_id": "6737ab70a4ce746810d8b2b9",

3

"name": "Laptop",

4

"price": 1200,

5

"description": "A sleek laptop with powerful performance, perfect for work

and entertainment.",

6

"stock": 30

7

}

Get Specific Product by Id

GET ⌵ http://localhost:5555/products/wrong-id Send

Query

Headers ²

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 500 Internal Server Error Size: 24 Bytes Time: 9 ms

Response

Headers ⁶

Cookies

Results

Docs

1

{

2

"error": "Server error"

3

}

Get Specific Product by Wrong Id Validation

Registering 2 users

POST ☐ http://localhost:5555/auth/register Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {  
2   "userName": "diwakar",  
3   "password": "saxena"  
4 }
```

Status: 200 OK Size: 42 Bytes Time: 1.96 s

Response Headers⁶ Cookies Results Docs

```
1 {  
2   "message": "User registered successfully"  
3 }
```

POST ☐ http://localhost:5555/auth/register Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {  
2   "userName": "internshala",  
3   "password": "trainings"  
4 }
```

Status: 200 OK Size: 42 Bytes Time: 14 ms

Response Headers⁶ Cookies Results Docs

```
1 {  
2   "message": "User registered successfully"  
3 }
```

localhost:27017 > shoppyglobe > users

> Open MongoDB shell

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

</>

Options

+ ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 2 of 2

↺

↻

≡

{ }

⌘

```
_id: ObjectId('6738e0f301a75465b739ae53')
userName: "diwakar"
password: "saxena"
userCart: Array (empty)
__v: 0
```

```
_id: ObjectId('6738e1d101a75465b739ae56')
userName: "internshala"
password: "trainings"
userCart: Array (empty)
__v: 0
```

After Register, inside Collections

The screenshot shows a REST client interface with a POST request to `http://localhost:5555/auth/login`. The request body is a JSON object with `"userName": "diwakar"` and an empty `"password"` field. The response is a 400 Bad Request with a message: `"message": "Invalid password"`.

Method	URL	Status	Size	Time
POST	http://localhost:5555/auth/login	400 Bad Request	30 Bytes	12 ms

Request Body (JSON):

```
{  "userName": "diwakar",  "password": ""}
```

Response (JSON):

```
{  "message": "Invalid password"}
```

Trying to login without Password

POST

http://localhost:5555/auth/login

Send

Query

Headers³

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1

{

2

"userName": "diwakar",

3

"password": "saxena"

4

}

Status: 200 OK Size: 197 Bytes Time: 41 ms

Response

Headers⁶

Cookies

Results

Docs

1

{

2

"message": "Login successful",

3

"jwtToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

.eyJ1c2VySWQiOiJkaXdha2FyIiwiaWF0IjoxNzMxNzg5MjAwLCJleHAiOiE3MzE3ODQ4MDB9

.EMQ4Ti9TFiq1GpIqshuAYUe5rhW3b5mI1mvMriHCWMg"

4

}

Trying To login in with correct Password Return JWT token



Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJkaXdha2FyIiwiaWF0IjoxNzE3ODQ4MD00LCJpYXN0IjoiZm9udC9EMQ4Ti9TFiq1GpIqshuAYUe5rhW3b5mI1mvMriHCWMg
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

```
{  "userId": "diwakar",  "iat": 1731781200,  "exp": 1731784800}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) ☐ secret base64 encoded
```

The decode JWT

POST ⌵ http://localhost:5555/cart Send

Query

Headers ³

Auth

Body ¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content Format

```
1 {
2   "productId": "some wrong id",
3   "quantity": 20
4 }
5
6
```

Status: 400 Bad Request Size: 28 Bytes Time: 12 ms

Response

Headers ⁶

Cookies

Results

Docs

```
1 {
2   "message": "Invalid token."
3 }
```

Trying to add in cart without Token has validations

POST ⌵ http://localhost:5555/cart Send

Query

Headers³

Auth

Body¹

Tests

Pre Run

HTTP Headers

☐ Raw

<input checked="" type="checkbox"/>	Accept	*/*	×
<input checked="" type="checkbox"/>	User-Agent	Thunder Client (https://www.thunderclient.com)	
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ	
<input type="checkbox"/>	header	value	

Status: 400 Bad Request Size: 28 Bytes Time: 12 ms

Response

Headers⁶

Cookies

Results

Docs

1

{

2 "message": "Invalid token."

3 }

Passing token we got after login in headers

POST

http://localhost:5555/cart

Send

Query

Headers³

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1 {
2   "productId": "6737ab70a4ce746810d8b2b9",
3   "quantity": 20
4 }
5
6
```

Status: 201 Created

Size: 125 Bytes

Time: 77 ms

Response

Headers⁶

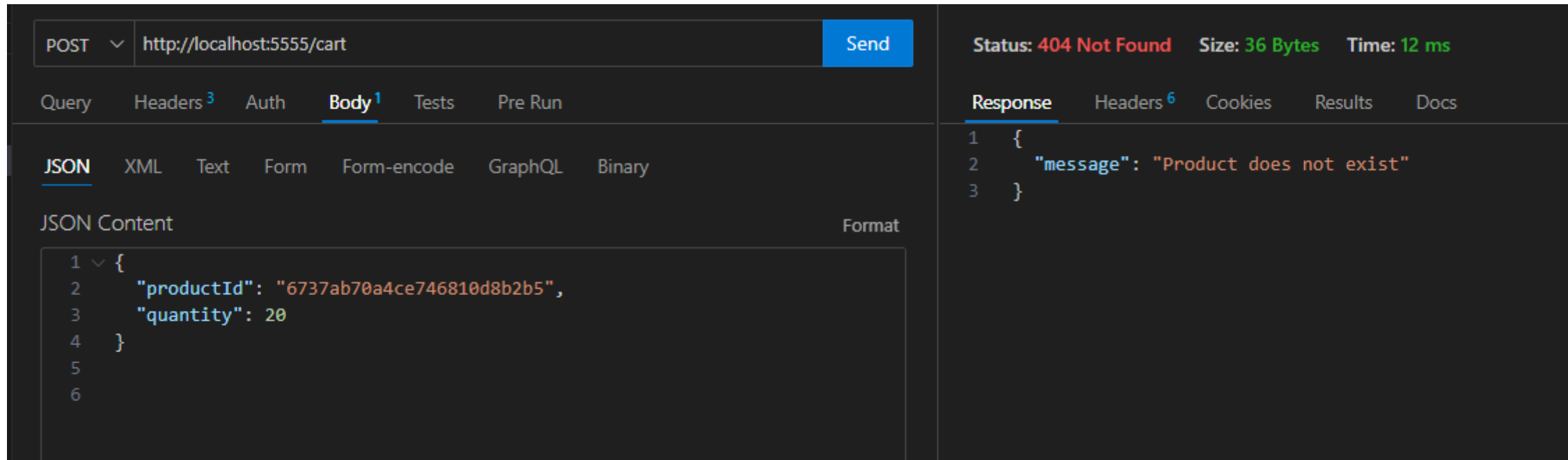
Cookies

Results

Docs

```
1 {
2   "message": "Item added",
3   "userCart": [
4     {
5       "productId": "6737ab70a4ce746810d8b2b9",
6       "quantity": 20,
7       "_id": "6738e3e601a75465b739ae5c"
8     }
9   ]
10 }
```

After adding token , adding to cart is success



Look carefully in ID , I **replaced 9 with 5 at the end**, from previous pic.

It means Wrong Id product cannot be added, it must exist in all products

localhost:27017 > shoppyglobe > users

> Open Mon

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

</>

+ ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 2 of 2

↺ ↻ ↷

< >

☰

```
{
  "_id": ObjectId('6738ef301a75465b739ae53'),
  "userName": "diwakar",
  "password": "saxena",
  "userCart": Array (1)
    0: Object
      "productId": ObjectId('6737ab70a4ce746810d8b2b9')
      "quantity": 20
      "_id": ObjectId('6738e3e601a75465b739ae5c')
  "__v": 1
}
```

```
{
  "_id": ObjectId('6738e1d101a75465b739ae56'),
  "userName": "internshala",
  "password": "trainings",
  "userCart": Array (empty)
  "__v": 0
}
```

1 item with 20 quantity is added.

PUT ⌵ http://localhost:5555/cart/6737ab70a4ce746810d8b2b9 Send

QueryHeaders³AuthBody¹TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "productId": "6737ab70a4ce746810d8b2b9",
3   "quantity": 40
4 }
5
6
```

Status: 200 OKSize: 132 BytesTime: 26 ms

ResponseHeaders⁶CookiesResultsDocs

```
1 {
2   "message": "Cart item updated",
3   "userCart": [
4     {
5       "productId": "6737ab70a4ce746810d8b2b9",
6       "quantity": 40,
7       "_id": "6738e3e601a75465b739ae5c"
8     }
9   ]
10 }
```

Updating Quantity to 40

```
  _id: ObjectId('6738e0f301a75465b739ae53')
  userName : "diwakar"
  password : "saxena"
  ▼ userCart : Array (1)
    ▼ 0: Object
      productId : ObjectId('6737ab70a4ce746810d8b2b9')
      quantity : 40
      _id : ObjectId('6738e3e601a75465b739ae5c')
    __v : 1
```

```
  _id: ObjectId('6738e1d101a75465b739ae56')
  userName : "internshala"
  password : "trainings"
  ▶ userCart : Array (empty)
  __v : 0
```

Quantity is now 40.

DELETE ☐ http://localhost:5555/cart/6737ab70a4ce746810d8b999 Send

Query **Headers³** Auth Body Tests Pre Run

HTTP Headers

☐ Raw

<input checked="" type="checkbox"/>	Accept	*/*
<input checked="" type="checkbox"/>	User-Agent	Thunder Client (https://www.thunderclient.com)
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
<input type="checkbox"/>	header	value

Status: **404 Not Found** Size: **36 Bytes** Time: **12 ms**

Response Headers⁶ Cookies Results Docs

```
1 {
2   "message": "Item not found in cart"
3 }
```

DELETE ☐ http://localhost:5555/cart/6737ab70a4ce746810d8b2b9 Send

Query **Headers³** Auth Body Tests Pre Run

HTTP Headers

☐ Raw

<input checked="" type="checkbox"/>	Accept	*/*
<input checked="" type="checkbox"/>	User-Agent	Thunder Client (https://www.thunderclient.com)
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
<input type="checkbox"/>	header	value

Status: **200 OK** Size: **53 Bytes** Time: **19 ms**

Response Headers⁶ Cookies Results Docs

```
1 {
2   "message": "Item removed successfully",
3   "userCart": []
4 }
```

Trying to delete an item with wrong ID vs with a correct ID, inside cart

localhost:27017 > shoppyglobe > users

[Open MongoDB shell](#)

Documents 0

Aggregations

Schema

Indexes 1

Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain

Reset

Find

</>

Options

+ ADD DATA

EXPORT DATA

UPDATE

DELETE

25

1 - 2 of 2

↺

<

>

≡

{ }

|

⌘

```
_id: ObjectId('6738e0f301a75465b739ae53')
userName: "diwakar"
password: "saxena"
userCart: Array (empty)
__v: 2
```

```
_id: ObjectId('6738e1d101a75465b739ae56')
userName: "internshala"
password: "trainings"
userCart: Array (empty)
__v: 0
```

User Cart after delete is now Empty, Cart Schema is nested inside User as userCart