

ALL API ROUTES TEST REPORTS FROM THUNDERCLIENT & MONGODB

1.Get all products

Method : GET,

URL: /products

GET

http://localhost:7070/products

Send

Query

Headers 3

Auth

Body 1

Tests

Pre Run

Query Parameters

parameter

value

Status: 200 OK

Size: 18.53 KB

Time: 29 ms

Response

Headers 7

Cookies

Results

Docs

```
1 {
2   "success": true,
3   "count": 30,
4   "data": [
5     {
6       "_id": "68e60288579541242a69e165",
7       "title": "Essence Mascara Lash Princess",
8       "description": "The Essence Mascara Lash Princess is a
9         popular mascara known for its volumizing and lengthening
10        effects. Achieve dramatic lashes with this long-lasting
11        and cruelty-free formula.",
12       "price": 9.99,
13       "discountPercentage": 10.48,
14       "rating": 2.56,
15       "stock": 99,
16       "brand": "Essence",
17       "category": "beauty",
18       "thumbnail": "https://cdn.dummyjson.com/product-images
19         /beauty/essence-mascara-lash-princess/thumbnail.webp",
20       "images": [
21         "https://cdn.dummyjson.com/product-images/beauty/essence
22         -mascara-lash-princess/1.webp"
23       ]
24     }
25   ]
26 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS

node - ShopyGlobe-backend

PS D:\FSD\Training\NodePractice\NodeAssignment> cd D:\FSD\Training\NodePractice\NodeAssignment\ShopyGlobe-backend
> shopyglobe-backend@1.0.0 start
> node app.js

Server running on port 7070
Connected to MongoDB

MongoDB Compass - ShopyGlobe/test.products

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (3)

sample

ShopyGlobe

admin

config

local

test

carts

products

Users

products

test

+

ShopyGlobe > test > products

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 - 25 of 30

```
{
  "_id": ObjectId("68e60288579541242a69e166"),
  "title": "Eyeshadow Palette with Mirror",
  "description": "The Eyeshadow Palette with Mirror offers a versatile range of eyeshado...",
  "price": 19.99,
  "discountPercentage": 18.19,
  "rating": 2.86,
  "stock": 34,
  "brand": "Glamour Beauty",
  "category": "beauty",
  "thumbnail": "https://cdn.dummyjson.com/product-images/beauty/eyeshadow-palette-with-m...",
  "images": Array (1)
  __v: 0
  createdAt: 2025-10-08T06:19:52.818+00:00
  updatedAt: 2025-10-08T06:19:52.818+00:00
}
```

```
{
  "_id": ObjectId("68e60288579541242a69e167"),
  "title": "Powder Canister",
  "description": "The Powder Canister is a finely milled setting powder designed to set ...",
  "price": 14.99,
  "discountPercentage": 9.84,
  "rating": 4.64,
  "stock": 89,
  "brand": "Velvet Touch",
  "category": "beauty",
  "thumbnail": "https://cdn.dummyjson.com/product-images/beauty/powder-canister/thumbn...",
  "images": Array (1)
  __v: 0
  createdAt: 2025-10-08T06:19:52.818+00:00
}
```

2. Get Single Product

Method : GET

URL: /products/:id

The screenshot shows a REST client interface with a GET request to `http://localhost:7070/products/68e60288579541242a69e166`. The response is a JSON object with the following structure:

```
2  "success": true,
3  "data": {
4    "id": "68e60288579541242a69e166",
5    "title": "Eyeshadow Palette with Mirror",
6    "description": "The Eyeshadow Palette with Mirror offers a
7      versatile range of eyeshadow shades for creating stunning
8      eye looks. With a built-in mirror, it's convenient for on
9      -the-go makeup application.",
10   "price": 19.99,
11   "discountPercentage": 18.19,
12   "rating": 2.86,
13   "stock": 34,
14   "brand": "Glamour Beauty",
15   "category": "beauty",
16   "thumbnail": "https://cdn.dummyjson.com/product-images/beauty/eyeshadow-palette-with-mirror/thumbnail.webp",
17   "images": [
18     "https://cdn.dummyjson.com/product-images/beauty/eyeshadow-palette-with-mirror/1.webp"
19   ],
20   "_v": 0,
21   "createdAt": "2025-10-08T06:19:52.818Z",
22   "updatedAt": "2025-10-08T06:19:52.818Z"
23 }
```

Below the REST client, a terminal window shows the command `node - ShoppyGlobe-backend` and the output `Server running on port 7070` and `Connected to MongoDB`.

The screenshot shows the MongoDB Compass interface. The 'products' collection is selected, and two documents are displayed:

```
{
  "_id": ObjectId('68e60288579541242a69e166'),
  "title": "Eyeshadow Palette with Mirror",
  "description": "The Eyeshadow Palette with Mirror offers a versatile range of eyeshado...",
  "price": 19.99,
  "discountPercentage": 18.19,
  "rating": 2.86,
  "stock": 34,
  "brand": "Glamour Beauty",
  "category": "beauty",
  "thumbnail": "https://cdn.dummyjson.com/product-images/beauty/eyeshadow-palette-with...",
  "images": Array (1)
  ...v: 0
  createdAt: 2025-10-08T06:19:52.818+00:00
  updatedAt: 2025-10-08T06:19:52.818+00:00
}
```

```
{
  "_id": ObjectId('68e60288579541242a69e167'),
  "title": "Powder Canister",
  "description": "The Powder Canister is a finely milled setting powder designed to set ...",
  "price": 14.99,
  "discountPercentage": 9.84,
  "rating": 4.64,
  "stock": 89,
  "brand": "Velvet Touch",
  "category": "beauty",
  "thumbnail": "https://cdn.dummyjson.com/product-images/beauty/powder-canister/thumbn...",
  "images": Array (1)
  ...v: 0
  createdAt: 2025-10-08T06:19:52.818+00:00
}
```

3.POST PRODUCT

Method: post

Url: /products

The screenshot shows a VS Code editor with a REST client request and response, and a terminal window. The REST client is configured with the following details:

- Method: POST
- URL: http://localhost:7070/products
- Status: 201 Created
- Size: 271 Bytes
- Time: 11 ms

The request body is a JSON object:

```
{  "title": "Wings of fire",  "description": "Autobiography of APJ Abdul Kalam",  "price": "200",  "stock": "50",  "category": "books"}
```

The response is a JSON object:

```
{  "success": true,  "data": {    "title": "Wings of fire",    "description": "Autobiography of APJ Abdul Kalam",    "price": 200,    "stock": 50,    "category": "books",    "images": [],    "_id": "68e603d3dfb305710b0c7b74",    "createdAt": "2025-10-08T06:25:23.111Z",    "updatedAt": "2025-10-08T06:25:23.111Z",    "__v": 0  }  }}
```

The terminal window shows the following commands and output:

```
PS D:\FSD\Training\NodePractice\NodeAssignment> cd D:\FSD\Training\NodePractice\NodeAssignment\ShopyGlobe-backend
> shopyglobe-backend@1.0.0 start
> node app.js

Server running on port 7070
Connected to MongoDB
```

The screenshot shows the MongoDB Compass interface with the following details:

- Database: ShoppyGlobe
- Collection: products
- Documents: 0
- Aggregations: 0
- Schema: 1
- Indexes: 1
- Validation: 0

The query bar shows the following query:

```
{ field: 'value' }
```

The results show two documents:

```
{  "_id": "68e60288579541242a69e182",  "title": "Kiwi",  "description": "Nutrient-rich kiwi, perfect for snacking or adding a tropical twist to-",  "price": 2.49,  "discountPercentage": 15.22,  "rating": 4.93,  "stock": 99,  "category": "groceries",  "thumbnail": "https://cdn.dummyjson.com/product-images/groceries/kiwi/thumbnail.webp",  "images": [  ],  "createdAt": "2025-10-08T06:19:52.820+00:00",  "updatedAt": "2025-10-08T06:19:52.820+00:00"}
```

```
{  "_id": "68e603d3dfb305710b0c7b74",  "title": "Wings of fire",  "description": "Autobiography of APJ Abdul Kalam",  "price": 200,  "stock": 50,  "category": "books",  "images": [  ],  "createdAt": "2025-10-08T06:25:23.111+00:00",  "updatedAt": "2025-10-08T06:25:23.111+00:00",  "__v": 0}
```

4. User Registration

Method: POST

URL: /auth/register

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

- Method:** POST
- URL:** http://localhost:7070/auth/register
- Body (JSON):**

```
{  "name": "test123",  "email": "test123@gmail.com",  "password": "test123"}
```
- Status:** 201 Created
- Size:** 261 Bytes
- Time:** 83 ms
- Response (JSON):**

```
{  "_id": "68e604d4dfb305710b0c7b77",  "name": "test123",  "email": "test123@gmail.com",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3ZCI6IjY4ZTYwNGQ0ZGZiMzA1NzEwYjBjN2I3NyIsIm1hdCI6MTc1OTkwNDk4MCwiZXhwIjoxNzYyNDk2OTgwZGw0LWVubG5me_1XrWQ1y32t3nxH1hoxJ6hMOHEV4zkjQ8c"}
```

Below the REST client, a terminal window shows the command to start the application:

```
PS D:\FSD\Training\NodePractice\NodeAssignment> cd D:\FSD\Training\NodePractice\NodeAssignment\ShoppYGlobe-backend
> shoppyglobe-backend@1.0.0 start
> node app.js

Server running on port 7070
Connected to MongoDB
```

The screenshot shows the MongoDB Compass interface with the following details:

- Database:** ShoppYGlobe
- Collection:** test > users
- Documents:** 1
- Query:** Type a query: { field: 'value' } or [Generate query](#)
- Buttons:** Explain, Reset, Find, </>, Options
- Document:**

```
{  "_id": ObjectId('68e604d4dfb305710b0c7b77'),  "name": "test123",  "email": "test123@gmail.com",  "password": "$2b$10$qyrK0T70G7jKDFMTh4ixZuN1J50oabA880W3RqMkDU3oN1a5QpQ1i",  "createdAt": 2025-10-08T06:29:40.565+00:00,  "updatedAt": 2025-10-08T06:29:40.565+00:00,  "__v": 0}
```

5. User Login

Method: POST

URL: /auth/login

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

- Method:** POST
- URL:** http://localhost:7070/auth/login
- Status:** 200 OK
- Size:** 261 Bytes
- Time:** 96 ms
- Body (JSON Content):**

```
1 {
2   "email": "test123@gmail.com",
3   "password": "test123"
4 }
```
- Response:**

```
1 {
2   "_id": "68e604d4dfb305710b0c7b77",
3   "name": "test123",
4   "email": "test123@gmail.com",
5   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4ZTYwNGQ0ZGZiMzA1NzEwYjBjN2I3NyIsIm1hdCI6MTc1OTkwNTA5MywiZXhwIjoxNzYyNDkzMDkzfkQ.BvojyJRJqp7QLFkHGFx1Qb4d1BQrJ6gfkJpiFv0nylE"
6 }
```
- Terminal:**

```
> shoppyglobe-backend@1.0.0 start
> node app.js

Server running on port 7070
Connected to MongoDB
```

The screenshot shows a MongoDB collection view with the following document:

```
{
  "_id": {
    "$oid": "68e604d4dfb305710b0c7b77"
  },
  "name": "test123",
  "email": "test123@gmail.com",
  "password": "$2b$10$qyrK0T70G7jKDFMTh4ixZuNiJ50oabA880W3RqMkDU3oN1a5QpQii",
  "createdAt": {
    "$date": "2025-10-08T06:29:40.565Z"
  },
  "updatedAt": {
    "$date": "2025-10-08T06:29:40.565Z"
  },
  "__v": 0
}
```

6.Add to Cart

Method: POST

URL: /cart

The screenshot shows a REST client interface with a POST request to `http://localhost:7070/cart`. The request body is a JSON object: `{ "productId": "68e60288579541242a69e166", "quantity": "2" }`. The response status is 201 Created, with a size of 249 Bytes and a time of 19 ms. The response body is a JSON object:

```
1 {
2   "user": "68e604d4dfb305710b0c7b77",
3   "items": [
4     {
5       "product": "68e60288579541242a69e166",
6       "quantity": 2,
7       "_id": "68e606af40c18cc8df71e897"
8     }
9   ],
10  "_id": "68e606af40c18cc8df71e896",
11  "createdAt": "2025-10-08T06:37:35.508Z",
12  "updatedAt": "2025-10-08T06:37:35.508Z",
13  "__v": 0
14 }
```

The screenshot shows the MongoDB Compass interface. The left sidebar shows the database structure with 'ShoppYGlobe' selected, containing 'admin', 'config', 'local', 'test', 'products', 'users', and 'Users'. The 'test' database is selected, and the 'carts' collection is highlighted. The main panel shows the 'Documents' tab with 0 documents. A query is entered: `{ field: 'value' }` or `Generate query`. The document content is displayed as follows:

```
_id: ObjectId('68e606af40c18cc8df71e896')
user: ObjectId('68e604d4dfb305710b0c7b77')
items: Array (3)
createdAt: 2025-10-08T06:37:35.508+00:00
updatedAt: 2025-10-08T06:38:33.622+00:00
__v: 2
```

URL: /cart

```

    _id: ObjectId('68e606af40c18cc8df71e896')
    user: ObjectId('68e604d4dfb305710b0c7b77')
  },
  'v': Array (3)
    0: Object
      product: ObjectId('68e60288579541242a69e166')
      quantity: 2
      _id: ObjectId('68e606af40c18cc8df71e897')
    1: Object
      product: ObjectId('68e60288579541242a69e16c')
      quantity: 2
      _id: ObjectId('68e606d740c18cc8df71e89d')
    2: Object
      product: ObjectId('68e60288579541242a69e171')
      quantity: 10
      _id: ObjectId('68e606e940c18cc8df71e8a4')
  },
  createdAt: 2025-10-08T06:37:35.598+00:00
  updatedAt: 2025-10-08T06:38:33.622+00:00
  _v: 2
}

```


8.Update Cart Item(Protected)

Method:PUT

URL: /cart/:id

The screenshot shows a REST client interface in VS Code. The request is a PUT to `http://localhost:7070/cart/68e606d740c18cc8df71e89d`. The request body is a JSON object: `{ "quantity": "20" }`. The response status is 200 OK, with a size of 421 Bytes and a time of 9 ms. The response body is a JSON object: `{ "_id": "68e606af40c18cc8df71e896", "user": "68e604d4dfb305710b0c7b77", "items": [{ "product": "68e60288579541242a69e166", "quantity": 2, "_id": "68e606af40c18cc8df71e897" }, { "product": "68e60288579541242a69e16c", "quantity": 20, "_id": "68e606d740c18cc8df71e89d" }, { "product": "68e60288579541242a69e171", "quantity": 10, "_id": "68e606e940c18cc8df71e8a4" }], "createdAt": "2025-10-08T06:37:35.508Z", "updatedAt": "2025-10-08T06:41:19.689Z", "v": 2 }`. The terminal shows the following stack trace:

```
at handle (D:\FSD\Training\NodePractice\NodeAssignment\ShoppyGlobe-backend\node_modules\router\index.js:435:11)
at Layer.handleRequest (D:\FSD\Training\NodePractice\NodeAssignment\ShoppyGlobe-backend\node_modules\router\lib\layer.js:152:17)
at D:\FSD\Training\NodePractice\NodeAssignment\ShoppyGlobe-backend\node_modules\router\index.js:295:15
at processParams (D:\FSD\Training\NodePractice\NodeAssignment\ShoppyGlobe-backend\node_modules\router\index.js:582:12)
at next (D:\FSD\Training\NodePractice\NodeAssignment\ShoppyGlobe-backend\node_modules\router\index.js:291:5)
at file:///D:/FSD/Training/NodePractice/NodeAssignment/ShoppyGlobe-backend/middlewares/auth.js:36:7
```

```
    "product": {
      "$oid": "68e60288579541242a69e16c"
    },
    "quantity": 20,
    "_id": {
      "$oid": "68e606d740c18cc8df71e89d"
    }
  },
  {
```

9.Remove from cart(Protected)

Method: DELETE

URL: /cart/:id

The screenshot shows a REST client interface with a DELETE request to `http://localhost:7070/cart/68e606d740c18cc8df71e89d`. The response is a JSON object with the following structure:

```
1 {
2   "_id": "68e606af40c18cc8df71e896",
3   "user": "68e604d4dfb305710b0c7b77",
4   "items": [
5     {
6       "product": "68e60288579541242a69e166",
7       "quantity": 2,
8       "_id": "68e606af40c18cc8df71e897"
9     },
10    {
11      "product": "68e60288579541242a69e171",
12      "quantity": 10,
13      "_id": "68e606e940c18cc8df71e8a4"
14    }
15  ],
16  "createdAt": "2025-10-08T06:37:35.508Z",
17  "updatedAt": "2025-10-08T06:42:29.032Z",
18  "__v": 3
19 }
```

The JSON content on the left is formatted as follows:

```
1
```

The bottom status bar shows the terminal output: `node - ShoppyGlobe-backend`.

```
_id: ObjectId('68e606af40c18cc8df71e896')
user : ObjectId('68e604d4dfb305710b0c7b77')
▼ items: Array (2)
  ▼ 0: Object
    product : ObjectId('68e60288579541242a69e166')
    quantity : 2
    _id : ObjectId('68e606af40c18cc8df71e897')
  ▼ 1: Object
    product : ObjectId('68e60288579541242a69e171')
    quantity : 10
    _id : ObjectId('68e606e940c18cc8df71e8a4')
createdAt : 2025-10-08T06:37:35.508+00:00
updatedAt : 2025-10-08T06:42:29.032+00:00
__v : 3
```