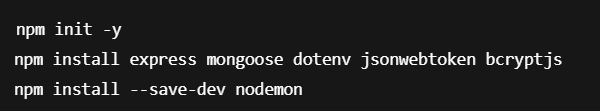
# Project: Build APIs with Node.js and Express.js for Shoppyglobe E-commerce.

1. Initialize a Node.js project:



1. Project Setup:

Technologies Used:

* Node.js
* Express.js
* MongoDB (with Mongoose)
* JSON Web Token (JWT)
* Thunder Client (for API testing)

1. Project Structure:



1. API End Points:

* Product Routes

Method Description EndPoint

GET Get all products /api/products

GET Get product by ID /api/products/:id

* Cart Routes

Method Description EndPoint

Post Add product to cart /api/cart

Put Update product Quantity /api/carts/:products/:id

Delete Remove Product /api/carts/:products/:id

* Auth Routes

Method Description EndPoint

POST Register new user /api/auth/register

POST Login user /api/auth/login

1. Testing:

Auth Testing:

## 1.POST (Register User):

Request: <http://localhost:5000/api/auth/register>

Body: In body we will write details.

{

"name": " Junaidkhan",

"email":”junaid@gmail.com",

"password”:”sunjun0518:

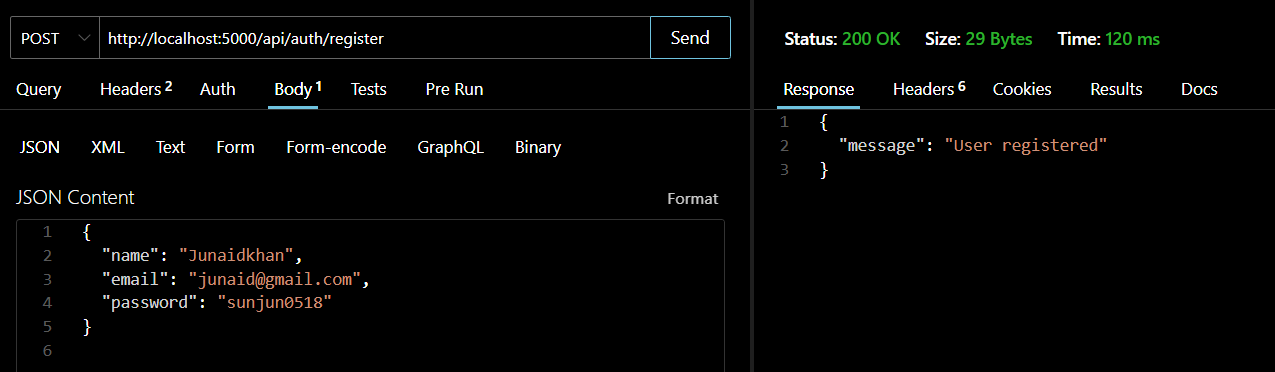
}

Response: It will display a message as

{“message”:”user registered”}

Status code: 200 OK

* **ThunderClient**



* **MongoDB Compass**



## 2.POST (Login User):

Request: <http://localhost:5000/api/auth/login>

Body: In body we will write details.

{

"email":”junaid@gmail.com",

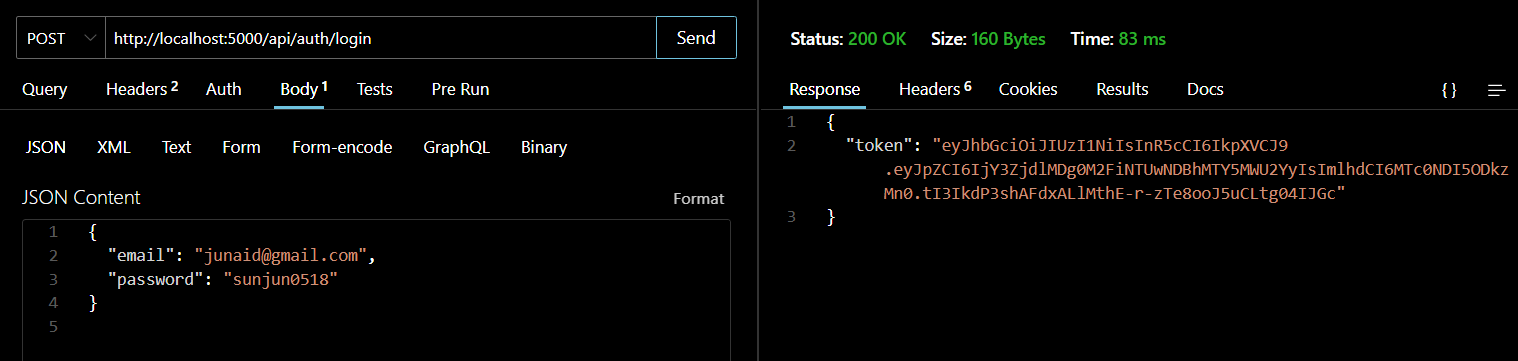
"password”:”sunjun0518:

}

Response: It will display a token

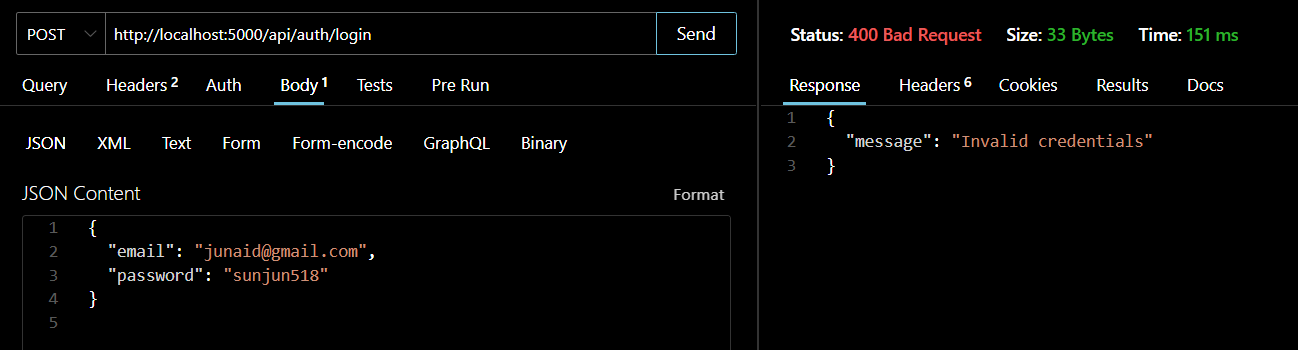
Status code: 200 OK

* **ThunderClient (if user has valid credentials)**



Copy the token for future purpose.

* **ThunderClient (if user has invalid credentials)**



Response: It will display a message as

{“message”:”invalid credentials”}

Status code: 400 Bad request

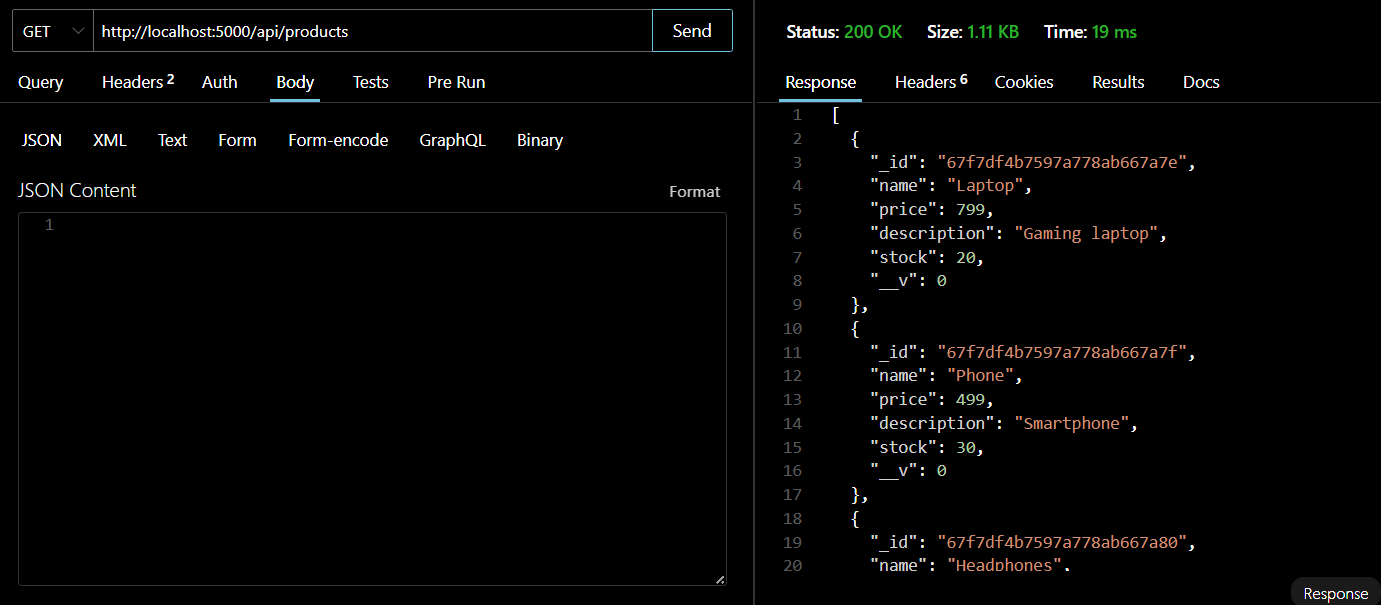
## 3. GET (All products):

Request: <http://localhost:5000/api/products>

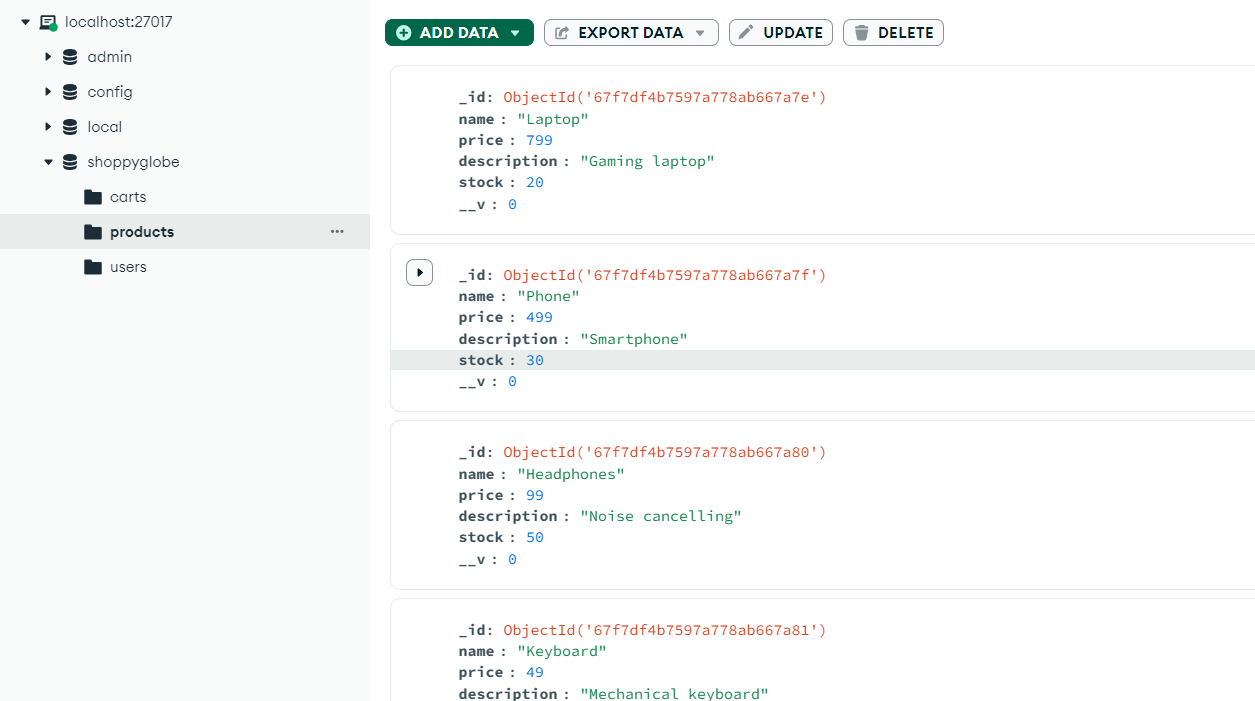
Response: It will display all the products.

Status code: 200 OK

* **ThunderClient**



* **MongoDB Compass**



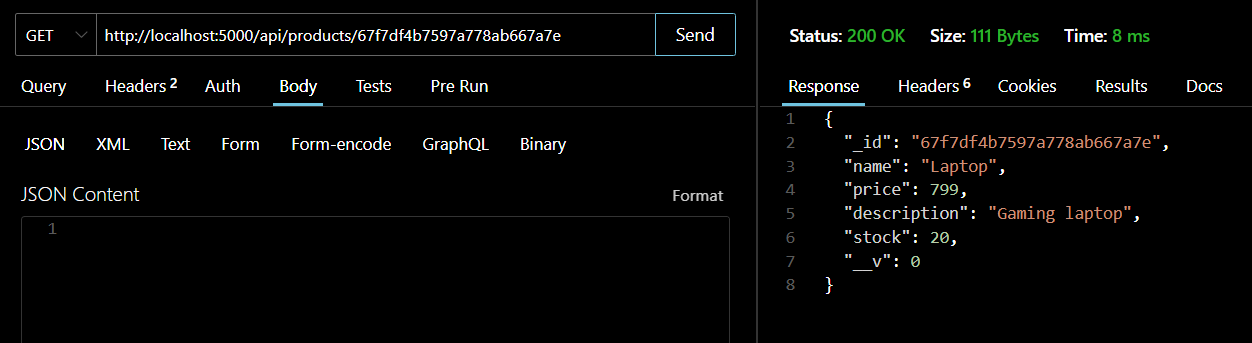
## 4. GET (Products with id):

Request:<http://localhost:5000/api/products/67f7df4b7597a778ab667a7e>

Response: It will display all the products.

Status code: 200 OK

* **ThunderClient**



## 5. POST (Cart Api):

Request: <http://localhost:5000/api/cart>

Response: It will display cart.

Body: In body we will write details.

{

"productId": "PRODUCT\_ID",

"quantity": 2

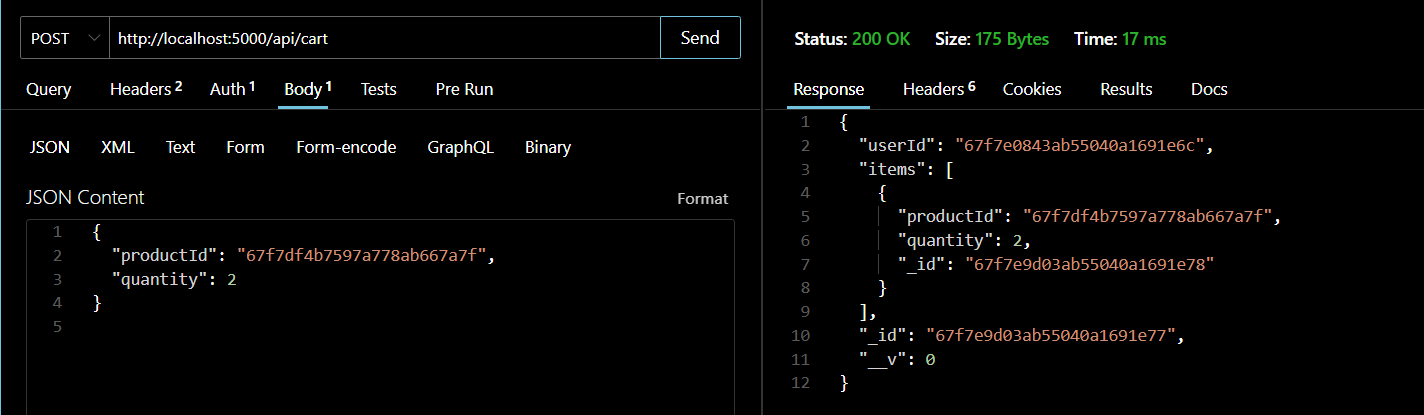
}

**Auth Header**:

* + Click on **"Auth" tab** in Thunder Client
  + Choose Bearer Token and paste the JWT token from login.

Status code: 200 OK

* **ThunderClient**



* **MongoDB Compass**



## 6. PUT (Update Quantity of cart item):

Request: <http://localhost:5000/api/cart>

Response: It will update quantity of cart.

Body: In body we will write details.

{

"quantity": 2

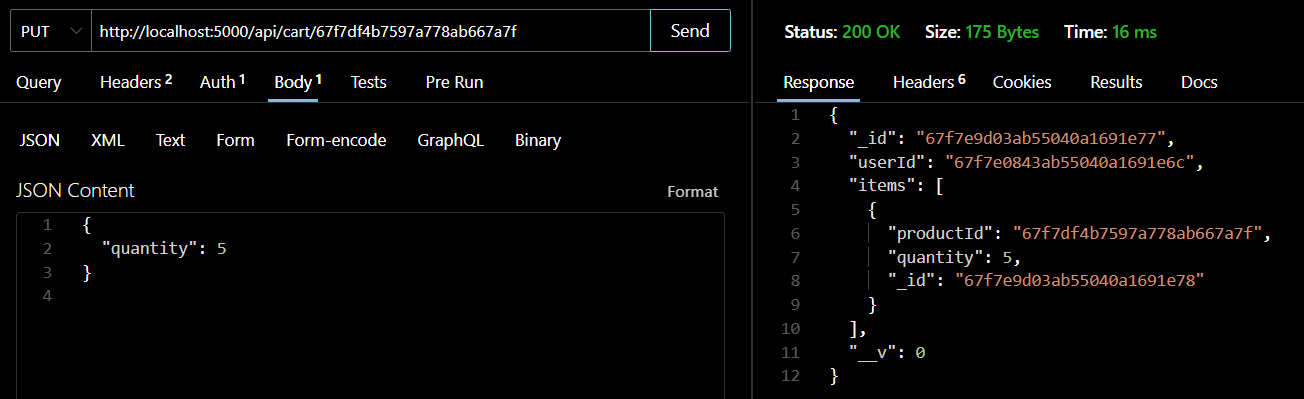
}

**Auth Header**:

* + Click on **"Auth" tab** in Thunder Client
  + Choose Bearer Token and paste the JWT token from login.

Response: It will update the quantity in the cart.

* **ThunderClient**

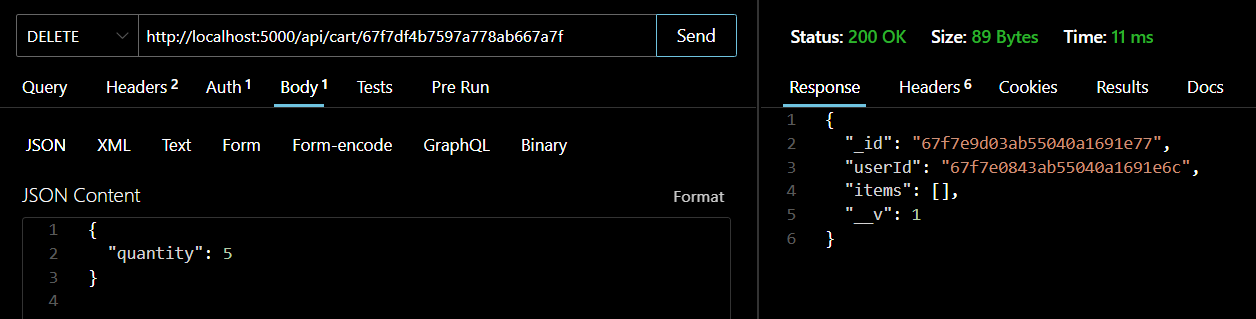


## 6. DELETE (Delete cart item):

Request: <http://localhost:5000/api/cart/67f7df4b7597a778ab667a7e>

Response: It will delete the items in the cart.

* **ThunderClient**



* **MongoDB Compass**

