

Post request for register user

The screenshot shows the Thunder Client interface with a new request configured for a POST method to the endpoint `http://localhost:3000/api/auth/register`. The request body is a JSON object containing user registration details. The response is a 201 Created status with a JSON body that includes the user's information and a JWT token.

Request:

```
POST http://localhost:3000/api/auth/register
```

JSON Content:

```
{  "name": "mohit",  "email": "moohit65@gmail.com",  "password": "1234567"}
```

Response: Status: 201 Created, Size: 411 Bytes, Time: 140 ms

```
{  "user": {    "email": "moohit65@gmail.com",    "password": "$2a$08$6ZhjRmsDLcU1mc81hbJA02TV528zgXTqAb3mzG7GzXV.0ZP03QwM",    "name": "mohit",    "id": "67a36f23c30fe38946f795cb",    "createdAt": "2025-02-05T14:01:07.635Z",    "updatedAt": "2025-02-05T14:01:07.635Z",    "__v": 0  },  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmQ1O1I2Z2ZmYyM2ZmZGZlMzg5NDZmNzY1IjE1LjE3YXQ1E3Mzg3NjQwNjQ5.3kth6fFFqHITDAjyzSupjxcG606uJ1fAR2HDDZ8oZ7Q"}
```

Post request for login user

The screenshot shows the Thunder Client interface with a new request configured for a POST method to the endpoint `http://localhost:3000/api/auth/login`. The request body is a JSON object containing login credentials. The response is a 200 OK status with a JSON body that includes the user's information and a JWT token.

Request:

```
POST http://localhost:3000/api/auth/login
```

JSON Content:

```
{  "email": "sandesh165@gmail.com",  "password": "1234567"}
```

Response: Status: 200 OK, Size: 415 Bytes, Time: 122 ms

```
{  "user": {    "id": "67a361c4f4286a54b050bdfe",    "email": "sandesh165@gmail.com",    "password": "$2a$08$LW3p8.k12AZ8v9IokMnD0.lc/6Ljj4k46EGUDfe0s1LlxXBV/G3cy",    "name": "sandesh",    "createdAt": "2025-02-05T13:04:04.769Z",    "updatedAt": "2025-02-05T13:04:04.769Z",    "__v": 0  },  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmQ1O1I2Z2ZmYyM2ZmZGZlMzg5NDZmNzY1IjE1LjE3YXQ1E3Mzg3NjQwNjQ5.GD7EH-mRAac0Wf6Xicu0Qe7J67SrnI6AU6FqMJ51Fo"}
```

Getting list of products

The screenshot shows the Thunder Client interface with a GET request to `http://localhost:3000/api/products` successfully executed. The status is 200 OK, and the response is a JSON array of two product objects.

Request: GET `http://localhost:3000/api/products`

Response: Status: 200 OK, Size: 813 Bytes, Time: 43 ms

```
1 {
2   {
3     "_id": "67a4873b58cfb389fc84d9ac",
4     "name": "iPhone",
5     "price": 5500,
6     "description": "The iPhone is a smartphone made by Apple that combines a computer, iPod, digital camera and cellular phone into one device with a touchscreen interface.",
7     "stockQuantity": 50,
8     "createdAt": "2025-02-06T09:56:11.730Z",
9     "updatedAt": "2025-02-06T09:56:11.730Z",
10    "__v": 0
11  },
12  {
13    "_id": "67a4883e58cfb389fc84d9ae",
14    "name": "HP Victus",
15    "price": 40000,
16    "description": "Next-level gamers need next-level processors like the 12th Gen Intel Core i5-12450H processor, which distributes performance where you need it the most. - Performance hybrid architecture - Intel Thread Director. Performance hybrid
```

Creating new product

The screenshot shows the Thunder Client interface with a POST request to `http://localhost:3000/api/products` successfully executed. The status is 201 Created, and the response is a JSON object indicating success.

Request: POST `http://localhost:3000/api/products`

Request Body (JSON Content):

```
1 {
2   "name": "iPhone",
3   "price": 5500,
4   "description": "The iPhone is a smartphone made by Apple that combines a computer, iPod, digital camera and cellular phone into one device with a touchscreen interface.",
5   "stockQuantity": 50
6 }
```

Response: Status: 201 Created, Size: 365 Bytes, Time: 59 ms

```
1 {
2   "status": "success",
3   "data": {
4     "name": "iPhone",
5     "price": 5500,
6     "description": "The iPhone is a smartphone made by Apple that combines a computer, iPod, digital camera and cellular phone into one device with a touchscreen interface.",
7     "stockQuantity": 50,
8     "_id": "67a4873b58cfb389fc84d9ac",
9     "createdAt": "2025-02-06T09:56:11.730Z",
10    "updatedAt": "2025-02-06T09:56:11.730Z",
11    "__v": 0
12  }
13 }
```

Getting product by id

The screenshot shows the Thunder Client interface with a GET request to `http://localhost:3000/api/products/67a4883e58cfb389fc84d9ae`. The response is a JSON object representing a product.

```
GET http://localhost:3000/api/products/67a4883e58cfb389fc84d9ae
```

Status: 200 OK Size: 473 Bytes Time: 40 ms

```
{
  "_id": "67a4883e58cfb389fc84d9ae",
  "name": "HP Victus",
  "price": 40000,
  "description": "Next-level gamers need next-level processors like the 12th Gen Intel Core i5-12450H processor, which distributes performance where you need it the most. - Performance hybrid architecture - Intel Thread Director. Performance hybrid architecture combines two new core microarchitectures",
  "stockQuantity": 25,
  "createdAt": "2025-02-06T10:00:30.987Z",
  "updatedAt": "2025-02-06T10:00:30.987Z",
  "__v": 0
}
```

Adding item to cart

The screenshot shows the Thunder Client interface with a POST request to `http://localhost:3000/api/cart`. The request body is a JSON object, and the response is a JSON object representing the updated cart state.

```
POST http://localhost:3000/api/cart
```

Status: 200 OK Size: 249 Bytes Time: 154 ms

```
{
  "user": "67a4853658cfb389fc84d9a9",
  "items": [
    {
      "product": "67a4883e58cfb389fc84d9ae",
      "quantity": 2,
      "_id": "67a48d6c58cfb389fc84d9bc"
    }
  ],
  "_id": "67a48d6c58cfb389fc84d9bb",
  "createdAt": "2025-02-06T10:22:36.994Z",
  "updatedAt": "2025-02-06T10:22:36.994Z",
  "__v": 0
}
```

Updating quantity in cart

The screenshot shows the Thunder Client interface with a PUT request to `http://localhost:3000/api/cart/67a4883e58cfb389fc84d9ae`. The request body is a JSON object: `{ "quantity": 7 }`. The response is a 200 OK status with a size of 334 Bytes and a time of 114 ms. The response body is a JSON object: `{ "_id": "67a48d6c58cfb389fc84d9bb", "user": "67a4853658cfb389fc84d9a9", "items": [{ "product": "67a4883e58cfb389fc84d9ae", "quantity": 7, "_id": "67a48d6c58cfb389fc84d9bc" }, { "product": "67a4873b58cfb389fc84d9ac", "quantity": 5, "_id": "67a48db158cfb389fc84d9c1" }], createdAt: "2025-02-06T10:22:36.994Z", updatedAt: "2025-02-06T10:31:40.343Z", __v: 1 }`. The terminal shows the server running on port 3000 and connected to MongoDB.

Deleting product from cart (product with 7 quantity deleted)

The screenshot shows the Thunder Client interface with a DELETE request to `http://localhost:3000/api/cart/67a4883e58cfb389fc84d9ae`. The response is a 200 OK status with a size of 249 Bytes and a time of 92 ms. The response body is a JSON object: `{ "_id": "67a48d6c58cfb389fc84d9bb", "user": "67a4853658cfb389fc84d9a9", "items": [{ "product": "67a4873b58cfb389fc84d9ac", "quantity": 5, "_id": "67a48db158cfb389fc84d9c1" }], createdAt: "2025-02-06T10:22:36.994Z", updatedAt: "2025-02-06T10:34:37.442Z", __v: 2 }`. The terminal shows the server running on port 3000 and connected to MongoDB.