

# ShoppyGlobe(backend)

## API Testing Screenshots

### ❖ GET /products success

GET

Status: 200 OK Size: 734 Bytes Time: 57 ms

Query Parameters

| parameter | value |
|-----------|-------|
|-----------|-------|

Response

```
1 {
2   {
3     "_id": "6898c4c1ec761760edaea865",
4     "name": "Wireless Mouse",
5     "price": 899,
6     "description": "Ergonomic 2.4G wireless mouse",
7     "stock": 25
8   },
9   {
10    "_id": "6898caecec761760edaea868",
11    "name": "Mechanical Keyboard",
12    "price": 3299,
13    "description": "RGB backlit mechanical keyboard with blue switches",
14  }
```

### MongoDB:

localhost:27017

- Usersdb
  - Users
- admin
- config
- shoppyglobe
  - carts
  - products**
  - users
- test
  - users
- Users

ADD DATA EXPORT DATA UPDATE DELETE

`_id: ObjectId('6898c4c1ec761760edaea865')`  
`name: "Wireless Mouse"`  
`price: 899`  
`description: "Ergonomic 2.4G wireless mouse"`  
`stock: 25`

`_id: ObjectId('6898caecec761760edaea868')`  
`name: "Mechanical Keyboard"`  
`price: "3299"`  
`description: "RGB backlit mechanical keyboard with blue switches"`  
`stock: "15"`

`_id: ObjectId('6898cb97ec761760edaea869')`  
`name: "USB-C Charging Cable"`  
`price: 299`  
`description: "Durable 1.5m USB-C to USB-A charging cable"`  
`stock: 40`

`_id: ObjectId('6898cbbcec761760edaea86a')`  
`name: "Laptop Stand"`  
`price: 699`  
`description: "Adjustable aluminum laptop stand for desk"`  
`stock: 30`

### ❖ GET /products/:id success :

GET

Status: 200 OK Size: 146 Bytes Time: 11 ms

Query Parameters

| parameter | value |
|-----------|-------|
|-----------|-------|

Response

```
1 {
2   "_id": "6898cb97ec761760edaea869",
3   "name": "USB-C Charging Cable",
4   "price": 299,
5   "description": "Durable 1.5m USB-C to USB-A charging cable",
6   "stock": 40
7 }
```

## ❖ GET /products/:id failed :

The screenshot shows a REST client interface. The top bar displays the method 'GET' and the URL 'http://localhost:3000/products/6898cb97ec761760edaea873'. A 'Send' button is visible. Below the URL bar, the 'Query' tab is selected, showing 'Query Parameters' with a table with columns 'parameter' and 'value'. The right panel shows the 'Response' tab with a status of '404 Not Found', size of '31 Bytes', and time of '6 ms'. The response body is a JSON object: 

```
{  "message": "Product not found"}
```

## ❖ POST /register success :

The screenshot shows a REST client interface. The top bar displays the method 'POST' and the URL 'http://localhost:3000/register'. A 'Send' button is visible. Below the URL bar, the 'Body' tab is selected, showing 'JSON Content' with a JSON object: 

```
{  "username": "test2",  "password": "pas321"}
```

. The right panel shows the 'Response' tab with a status of '201 Created', size of '29 Bytes', and time of '107 ms'. The response body is a JSON object: 

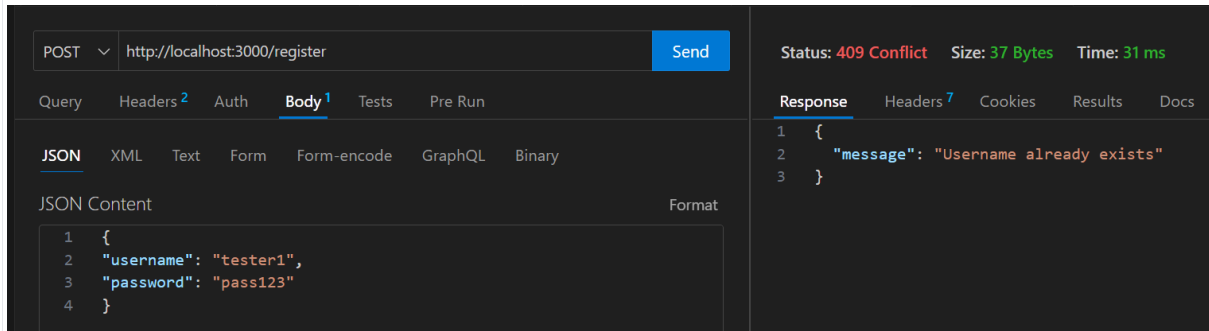
```
{  "message": "User registered"}
```

## MongoDB:

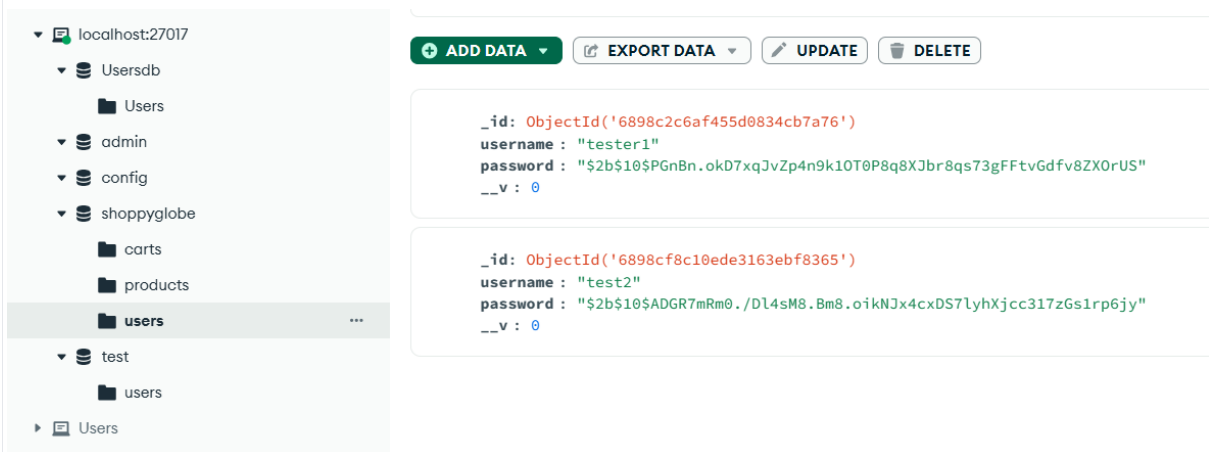
The screenshot shows a MongoDB interface. On the left, a sidebar displays a tree view of the database structure: 'localhost:27017' > 'Usersdb' > 'Users'. The main area shows a list of documents with columns: '\_id', 'username', 'password', and '\_\_v'. There are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The documents are:

| _id                                  | username | password  | __v |
|--------------------------------------|----------|---|-----|
| ObjectId('6898c2c6af455d0834cb7a76') | tester1  | \$2b\$10\$PGnBn.okD7xqJvZp4n9k10T0P8q8XJbr8qs73gFFtvGdfv8ZX0rUS | 0   |
| ObjectId('6898cf8c10ede3163ebf8365') | test2    | \$2b\$10\$ADGR7mRm0./D14sM8.Bm8.oikNJx4cxDS7lyhXjcc317zGs1rp6jy | 0   |

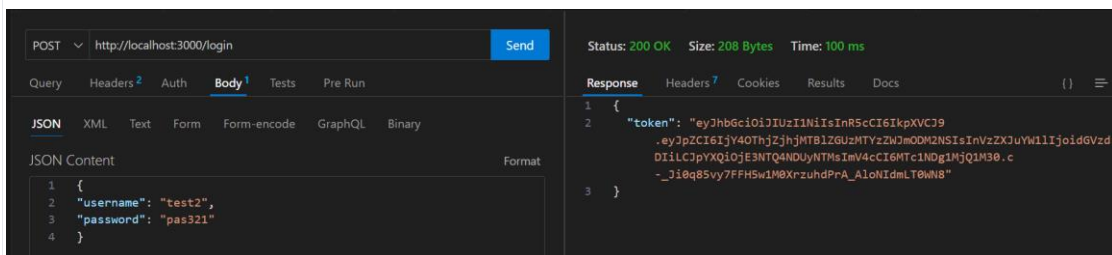
- ❖ POST /register duplicate username conflict :



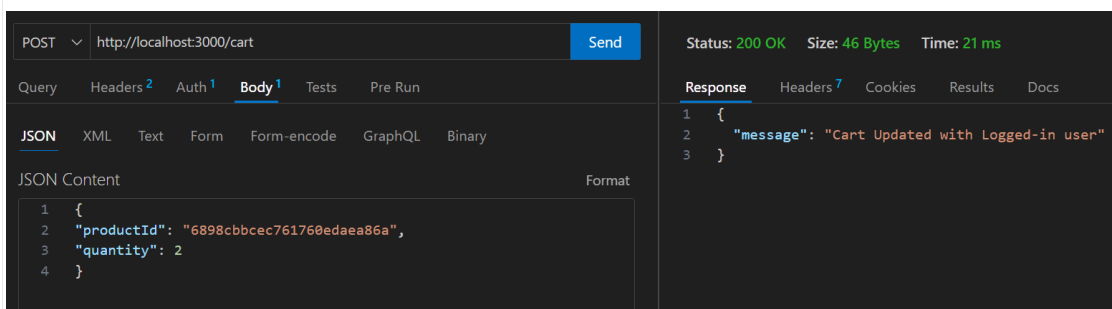
## MongoDB:



- ❖ POST /login success with token :



- ❖ POST /cart success (with token) :



## MongoDB:

localhost:27017

- Usersdb
- admin
- config
- shoppyglobe
  - carts**
  - products
  - users
- test
  - users

ADD DATAEXPORT DATAUPDATEDELETE

```
1  _id: ObjectId('6898d11b10ede3163ebf836b')
2  userId: 6898cf8c10ede3163ebf8365
3  items: Array (1)
4    0: Object
5      productId: 6898cbbcec761760edaea86a
6      quantity: 2
7      _id: 6898d11b10ede3163ebf836d
8    __v: 1
```

❖ PUT /cart/:productId success :

PUT http://localhost:3000/cart/6898cbbcec761760edaea86a Send

Status: 200 OK Size: 55 Bytes Time: 11 ms

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

1 {  
2 "quantity": 3  
3 }

Format

Response Headers Cookies Results Docs

1 {  
2 "message": "the cart with the item's quantity updated"  
3 }

## MongoDB:

localhost:27017

- Usersdb
- admin
- config
- shoppyglobe
  - carts**
  - products
  - users
- test
  - users
- Users

ADD DATAEXPORT DATAUPDATEDELETE

```
_id: ObjectId('6898d11b10ede3163ebf836b')
userId: ObjectId('6898cf8c10ede3163ebf8365')
items: Array (1)
  0: Object
    productId: ObjectId('6898cbbcec761760edaea86a')
    quantity: 3
    _id: ObjectId('6898d50e10ede3163ebf837c')
  __v: 3
```

## ❖ PUT /cart/:productId invalid-quantity error :

PUT

Query Headers <sup>2</sup> Auth <sup>1</sup> **Body <sup>1</sup>** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "quantity": 0
3 }
```

Status: **400 Bad Request** Size: **35 Bytes** Time: **4 ms**

**Response** Headers <sup>7</sup> Cookies Results Docs

```
1 {
2   "message": "Quantity must be >= 1"
3 }
```

## ❖ DELETE /cart/:productId success :

DELETE

Query Headers <sup>2</sup> Auth <sup>1</sup> **Body** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1
```

Status: **200 OK** Size: **89 Bytes** Time: **10 ms**

**Response** Headers <sup>7</sup> Cookies Results Docs

```
1 {
2   "_id": "6898d11b10ede3163ebf836b",
3   "userId": "6898cf8c10ede3163ebf8365",
4   "items": [],
5   "__v": 2
6 }
```

## MongoDB :

localhost:27017

- Usersdb
- admin
- config
- shoppyglobe
  - carts**
  - products
  - users
- test
  - users
- Users

```
{
  _id: ObjectId('6898d11b10ede3163ebf836b')
  userId: ObjectId('6898cf8c10ede3163ebf8365')
  items: Array (empty)
  __v: 2
}
```

❖ 401 unauthorized case for missing/invalid token :

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

- Method:** PUT
- URL:** http://localhost:3000/cart/6898cbbcec761760edaea86a
- Body:** JSON Content: 

```
1 {  
2   "quantity": "3"  
3 }
```
- Status:** 401 Unauthorized
- Size:** 42 Bytes
- Time:** 5 ms
- Response:**

```
1 {  
2   "message": "Authorization token required"  
3 }
```