

1. To retrieve all the products, make a Get REQUEST to the server.

GET: /api/products

GET ▼ http://localhost:3000/api/products Send ▼

RESPONSE: 200 OK

200 OK 5.55 ms 412 B Just Now ▼

Preview ▼ Headers ⁷ Cookies Timeline

```
1 ▼ [  
2   {  
3     "_id": "65760caa8fd3e37bd00e466f",  
4     "name": "Royal Gala",  
5     "description": "Apple from Argentina",  
6     "price": 8,  
7     "quantity": 2000,  
8     "category": "Fruit"  
9   },  
10  {  
11    "_id": "65760caa8fd3e37bd00e4670",  
12    "name": "Banana Eco",  
13    "description": "Banana from Ecuador",  
14    "price": 6,  
15    "quantity": 5000,  
16    "category": "Fruit"  
17  },  
18  {  
19    "_id": "65760caa8fd3e37bd00e4671",  
20    "name": "Avocado Mogen",  
21    "description": "Avocado from Chile",  
22    "price": 30,  
23    "quantity": 800,  
24    "category": "Fruit"  
25  }  
26 ]
```

2. To retrieve a specific product by ID, make a GET REQUEST to the server.

GET: /api/products/:id

GET ▼ http://localhost:3000/api/products/65760caa8fd3e37bd00e466f Send ▼

RESPONSE: 200 OK

The screenshot shows a REST client interface with a dark theme. At the top, the response status is '200 OK' in a green box, with '5.24 ms' and '136 B' in grey boxes, and 'Just Now' with a dropdown arrow on the right. Below this is a tab bar with 'Preview' selected, 'Headers' (with a '7' badge), 'Cookies', and 'Timeline'. The main area displays a JSON response:

```
1 {  
2   "_id": "65760caa8fd3e37bd00e466f",  
3   "name": "Royal Gala",  
4   "description": "Apple from Argentina",  
5   "price": 8,  
6   "quantity": 2000,  
7   "category": "Fruit"  
8 }
```

If the product cannot be found or the ID is incorrect when retrieving a specific product by ID:

The screenshot shows a REST client interface with a dark theme. The request is a GET to 'http://localhost:3000/api/products/65760caa8fd3e37bd00e466a'. A 'Send' button with a dropdown arrow is on the right.

RESPONSE: 404 Not Found

The screenshot shows a REST client interface with a dark theme. At the top, the response status is '404 Not Found' in an orange box, with '3.84 ms' and '27 B' in grey boxes, and 'Just Now' with a dropdown arrow on the right. Below this is a tab bar with 'Preview' selected, 'Headers' (with a '7' badge), 'Cookies', and 'Timeline'. The main area displays a JSON response:

```
1 {  
2   "err": "Product not found"  
3 }
```

3. To create a new product, make a POST REQUEST to the server.

POST: /api/products

The screenshot shows a REST client interface with a dark theme. The request is a POST to 'http://localhost:3000/api/products'. A 'Send' button with a dropdown arrow is on the right. Below the request bar is a tab bar with 'JSON' selected, 'Auth', 'Query', 'Headers' (with a '2' badge), and 'Docs'. The main area displays a JSON request body:

```
1 {  
2   "name": "Kiwi Yellow",  
3   "description": "Kiwi from New Zealand",  
4   "price": 17,  
5   "quantity": 1400,  
6   "category": "fruit"  
7 }
```

RESPONSE: 201 Created

201 Created5.22 ms185 BJust Now ▾

Preview ▾Headers7CookiesTimeline

```
1 {  
2   "message": "Product created!",  
3   "body": {  
4     "name": "Kiwi Yellow",  
5     "description": "Kiwi from New Zealand",  
6     "price": 17,  
7     "quantity": 1400,  
8     "category": "fruit",  
9     "_id": "65760d30b67e685d1dd0f874",  
10    "__v": 0  
11  }  
12 }
```

Doublecheck with GET all products to make sure the new product is added:

200 OK4.55 ms560 BJust Now ▾

Preview ▾Headers7CookiesTimeline

```
1  [
2    {
3      "_id": "65760caa8fd3e37bd00e466f",
4      "name": "Royal Gala",
5      "description": "Apple from Argentina",
6      "price": 8,
7      "quantity": 2000,
8      "category": "Fruit"
9    },
10   {
11     "_id": "65760caa8fd3e37bd00e4670",
12     "name": "Banana Eco",
13     "description": "Banana from Ecuador",
14     "price": 6,
15     "quantity": 5000,
16     "category": "Fruit"
17   },
18   {
19     "_id": "65760caa8fd3e37bd00e4671",
20     "name": "Avocado Mogen",
21     "description": "Avocado from Chile",
22     "price": 30,
23     "quantity": 800,
24     "category": "Fruit"
25   },
26   {
27     "_id": "65760d30b67e685d1dd0f874",
28     "name": "Kiwi Yellow",
29     "description": "Kiwi from New Zealand",
30     "price": 17,
31     "quantity": 1400,
32     "category": "fruit",
33     "__v": 0
34   }
35 ]
```

If any of the required property is missing when creating a new product with POST REQUEST:

POST ▼ http://localhost:3000/api/products Send ▼

JSON ▼ Auth ▼ Query Headers 2 Docs

```
1 {  
2   "name": "Kiwi Yellow",  
3   "price": 17,  
4   "quantity": 1400,  
5   "category": "fruit"  
6 }
```

RESPONSE: 400 Bad Request

400 Bad Request 2.76 ms 21 B Just Now ▼

Preview ▼ Headers 7 Cookies Timeline

```
1 {  
2   "err": "Bad request"  
3 }
```

4. To update a specific product by ID, make a PUT REQUEST to the server.

PUT: /api/products/:id

PUT ▼ http://localhost:3000/api/products/65760caa8fd3e37bd00e466f Send ▼

JSON ▼ Auth ▼ Query Headers 2 Docs

```
1 {  
2   "price": 10  
3 }
```

RESPONSE: 200 OK

200 OK 17.4 ms 137 B Just Now ▼

Preview ▼ Headers 7 Cookies Timeline

```
1 {  
2   "_id": "65760caa8fd3e37bd00e466f",  
3   "name": "Royal Gala",  
4   "description": "Apple from Argentina",  
5   "price": 10,  
6   "quantity": 2000,  
7   "category": "Fruit"  
8 }
```

If update non-existed property when updating a specific product with PUT REQUEST:

```
JSON ▾ Auth ▾ Query Headers 2 Docs
1 ▾ {
2   "expired_date": "2024/12/05"
3 }
```

RESPONSE: 400 Bad Request

```
400 Bad Request 4.29 ms 21 B Just Now ▾
Preview ▾ Headers 7 Cookies Timeline
1 ▾ {
2   "err": "Bad request"
3 }
```

If update property to the non-existed product when updating a specific product with PUT REQUEST:

```
PUT ▾ http://localhost:3000/api/products/65760caa8fd3e37bd00e466a Send ▾
JSON ▾ Auth ▾ Query Headers 2 Docs
1 ▾ {
2   "price": 10
3 }
```

RESPONSE: 404 Not Found

```
404 Not Found 4.16 ms 27 B Just Now ▾
Preview ▾ Headers 7 Cookies Timeline
1 ▾ {
2   "err": "Product not found"
3 }
```

5. To delete a product by ID, make a DELETE REQUEST to the server.

DELETE: /api/products/:id

```
DELETE ▾ http://localhost:3000/api/products/65760caa8fd3e37bd00e466f Send ▾
```

RESPONSE: 200 OK

```
200 OK 5.11 ms 44 B Just Now ▾
Preview ▾ Headers 7 Cookies Timeline
1 "Document with Royal Gala has been deleted!"
```

Doublecheck with GET all products to make sure the product is deleted:

200 OK 6.89 ms 423 B Just Now ▾

Preview ▾ Headers 7 Cookies Timeline

```
1 [
2   {
3     "_id": "65760caa8fd3e37bd00e4670",
4     "name": "Banana Eco",
5     "description": "Banana from Ecuador",
6     "price": 6,
7     "quantity": 5000,
8     "category": "Fruit"
9   },
10  {
11    "_id": "65760caa8fd3e37bd00e4671",
12    "name": "Avocado Mogen",
13    "description": "Avocado from Chile",
14    "price": 30,
15    "quantity": 800,
16    "category": "Fruit"
17  },
18  {
19    "_id": "65760d30b67e685d1dd0f874",
20    "name": "Kiwi Yellow",
21    "description": "Kiwi from New Zealand",
22    "price": 17,
23    "quantity": 1400,
24    "category": "fruit",
25    "__v": 0
26  }
27 ]
```

If the product cannot be found or the ID is incorrect when deleting a specific product by ID:

DELETE ▾ http://localhost:3000/api/products/65760caa8fd3e37bd00e466a Send ▾

RESPONSE: 404 Not Found

404 Not Found 4.46 ms 27 B Just Now ▾

Preview ▾ Headers 7 Cookies Timeline

```
1 {
2   "err": "Product not found"
3 }
```

6. To delete all the products, make a DELETE REQUEST to the server.

DELETE: /api/products

DELETE ▼ http://localhost:3000/api/products Send ▼

RESPONSE: 200 OK

200 OK 8.36 ms 34 B Just Now ▼

Preview ▼ Headers ⁷ Cookies Timeline

```
1 "All documents have been deleted!"
```

Doublecheck with GET all products to make sure all the products are deleted:

200 OK 5.22 ms 2 B Just Now ▼

Preview ▼ Headers ⁷ Cookies Timeline

```
1 []
```

7. To retrieve a product whose name contains specified keyword, make a GET REQUEST to the server.

GET: /api/product?name=an

GET ▼ http://localhost:3000/api/product?name=an Send ▼

RESPONSE: 200 OK

200 OK 4.95 ms 135 B Just Now ▼

Preview ▼ Headers ⁷ Cookies Timeline

```
1 {
2   "_id": "657616158fd3e37bd00e4674",
3   "name": "Banana Eco",
4   "description": "Banana from Ecuador",
5   "price": 6,
6   "quantity": 5000,
7   "category": "Fruit"
8 }
```

If the product cannot be found or the name is incorrect when retrieving a specific product whose name contains specified keyword:

GET ▼ http://localhost:3000/api/product?name=ab Send ▼

RESPONSE: 404 Not Found

404 Not Found4.53 ms27 BJust Now ▾

Preview ▾Headers7CookiesTimeline

```
1 {
2   "err": "Product not found"
3 }
```

If the search param is incorrect when retrieving a specific product whose name contains specified keyword:

GET ▾ http://localhost:3000/api/product?search=anSend ▾

RESPONSE: 400 Bad Request

400 Bad Request2.18 ms21 BJust Now ▾

Preview ▾Headers7CookiesTimeline

```
1 {
2   "err": "Bad request"
3 }
```

- To send back HTML with all products in a table with all information about the product, make a GET REQUEST to the server.

GET: /

GET ▾ http://localhost:3000/Send ▾

RESPONSE: 200 OK with HTML

200 OK24.5 ms578 BJust Now ▾

Preview ▾Headers7CookiesTimeline

Products

Name	Description	Price	Quantity	Category
Royal Gala	Apple from Argentina	8	2000	Fruit
Banana Eco	Banana from Ecuador	6	5000	Fruit
Avocado Mogen	Avocado from Chile	30	800	Fruit