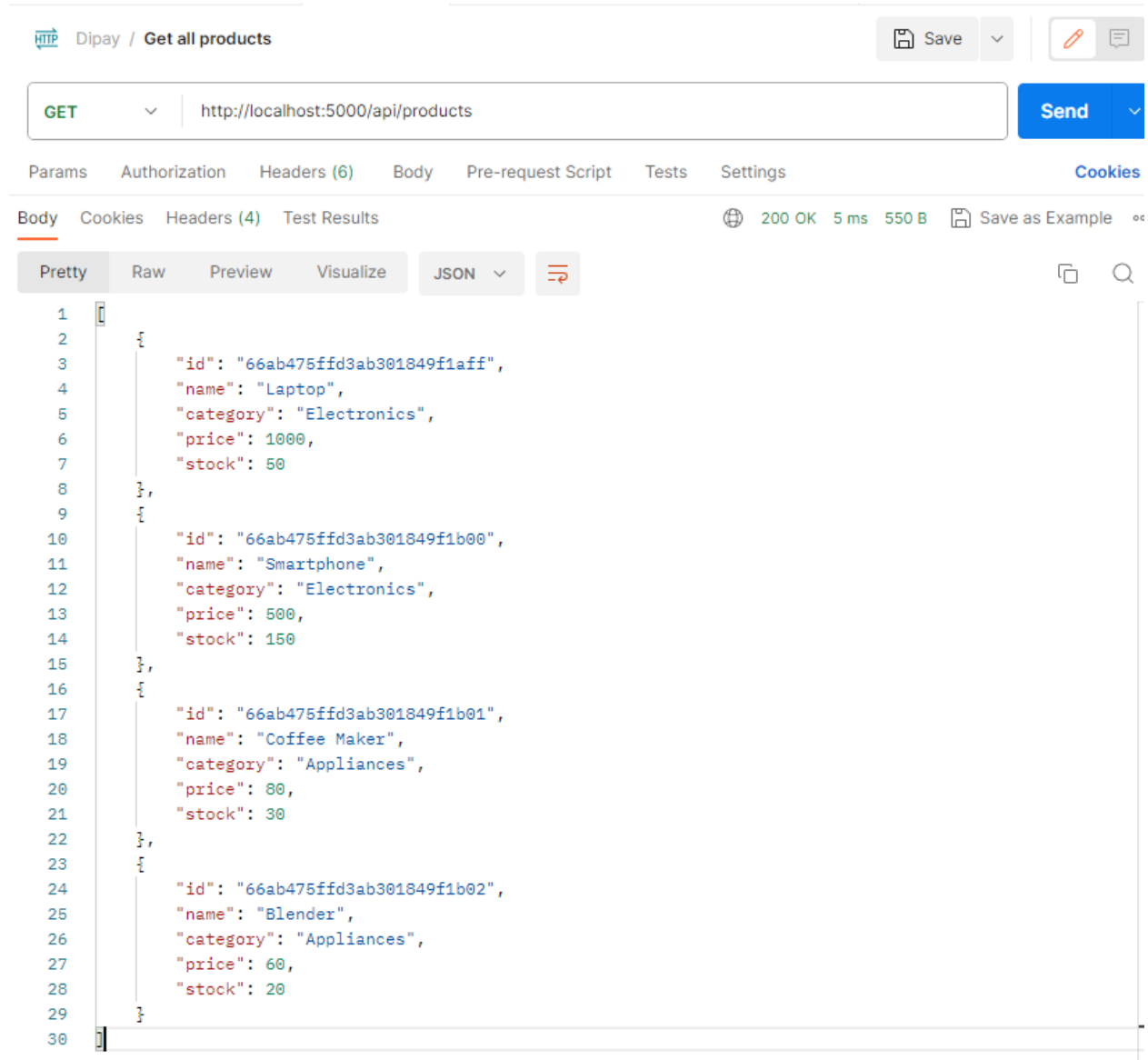


Note that the *products* collection has been filled with four data samples, as shown in figure 1

1. GET all products



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

- URL:** `http://localhost:5000/api/products`
- Method:** `GET`
- Status:** `200 OK`, `5 ms`, `550 B`
- Response Body (JSON):**

```
{
  "id": "66ab475ffd3ab301849f1aff",
  "name": "Laptop",
  "category": "Electronics",
  "price": 1000,
  "stock": 50
},
{
  "id": "66ab475ffd3ab301849f1b00",
  "name": "Smartphone",
  "category": "Electronics",
  "price": 500,
  "stock": 150
},
{
  "id": "66ab475ffd3ab301849f1b01",
  "name": "Coffee Maker",
  "category": "Appliances",
  "price": 80,
  "stock": 30
},
{
  "id": "66ab475ffd3ab301849f1b02",
  "name": "Blender",
  "category": "Appliances",
  "price": 60,
  "stock": 20
}
```

2. GET a product by id

HTTP Dipay / Get a product by id Save

GET ⌵ http://localhost:5000/api/products/66ab475ffd3ab301849f1b01 Send ⌵

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results 200 OK 6 ms 249 B Save as Example ⋮

Pretty Raw Preview Visualize JSON ⌵

```
1 {
2   "id": "66ab475ffd3ab301849f1b01",
3   "name": "Coffee Maker",
4   "category": "Appliances",
5   "price": 80,
6   "stock": 30
7 }
```

3. CREATE a product

- Postman

Postman interface showing a POST request to `http://localhost:5000/api/products`. The request body is a JSON object:

```
{  "name": "Microwave",  "category": "Books",  "price": 32.99}
```

The response is a JSON object:

```
{  "id": "66abb4e7244471ce2dd1878a",  "name": "Microwave",  "category": "Books",  "price": 32.99,  "stock": 0}
```

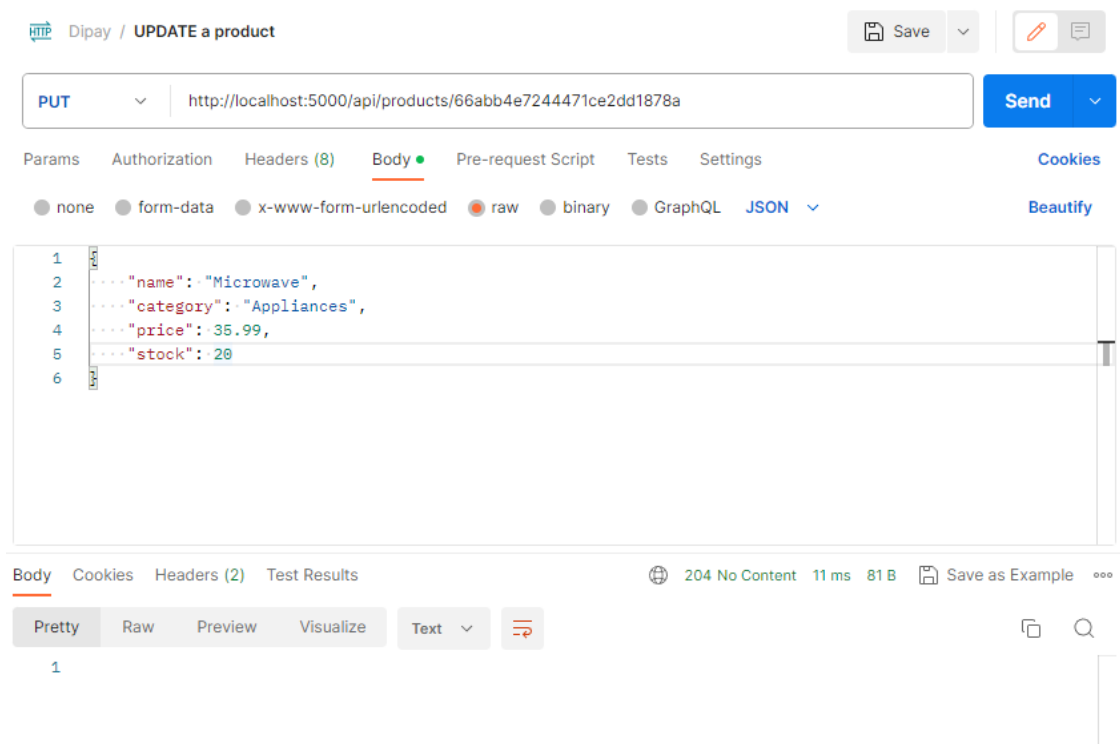
- Database

DipayDB interface showing the `products` collection. The interface displays a list of products with their details:

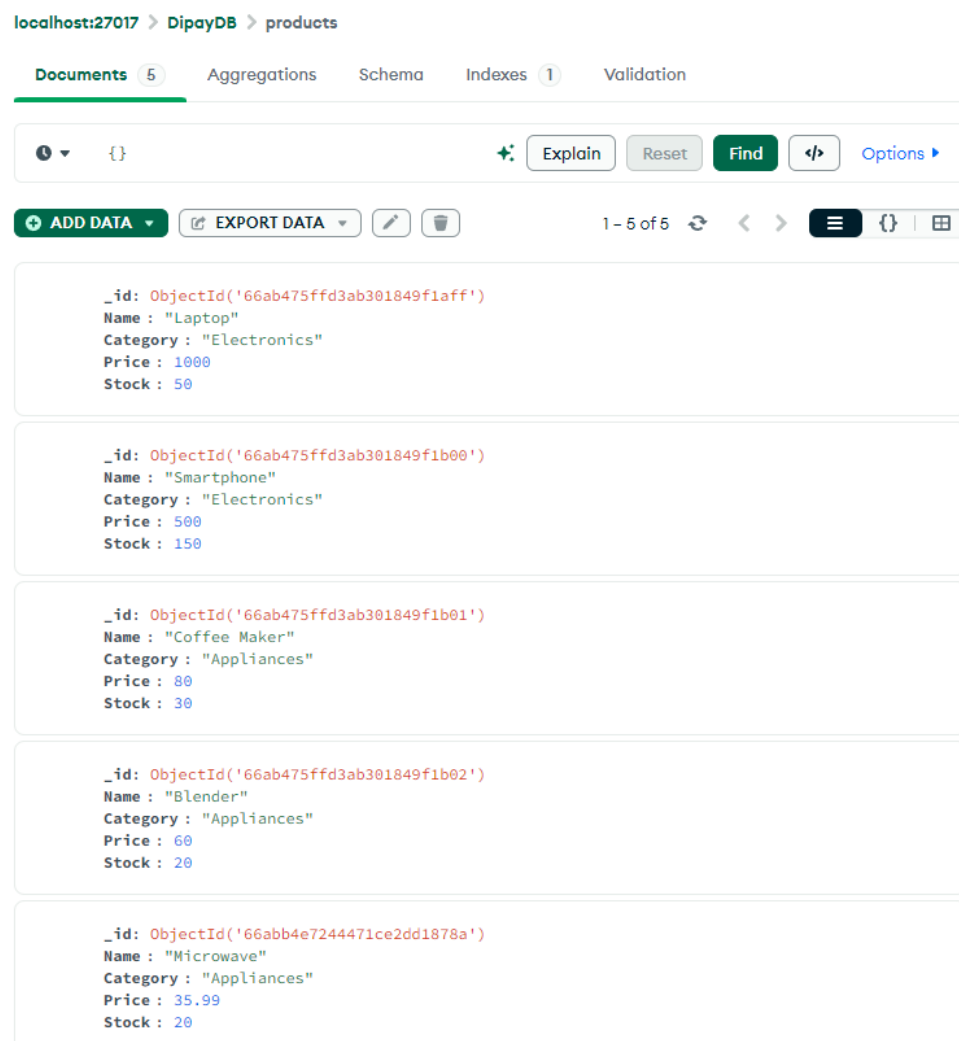
_id	Name	Category	Price	Stock
ObjectId('66ab475ffd3ab301849f1aff')	Laptop	Electronics	1000	50
ObjectId('66ab475ffd3ab301849f1b00')	Smartphone	Electronics	500	150
ObjectId('66ab475ffd3ab301849f1b01')	Coffee Maker	Appliances	80	30
ObjectId('66ab475ffd3ab301849f1b02')	Blender	Appliances	60	20
ObjectId('66abb4e7244471ce2dd1878a')	Microwave	Books	32.99	0

4. UPDATE a product

- Postman

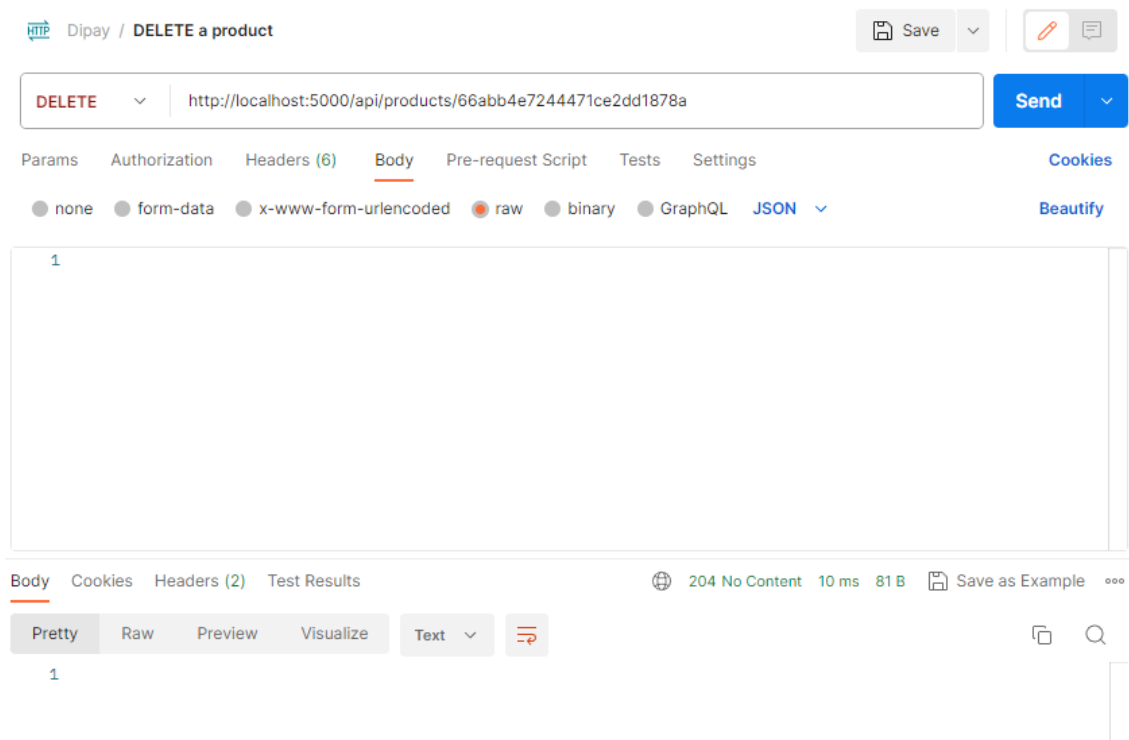


- Database

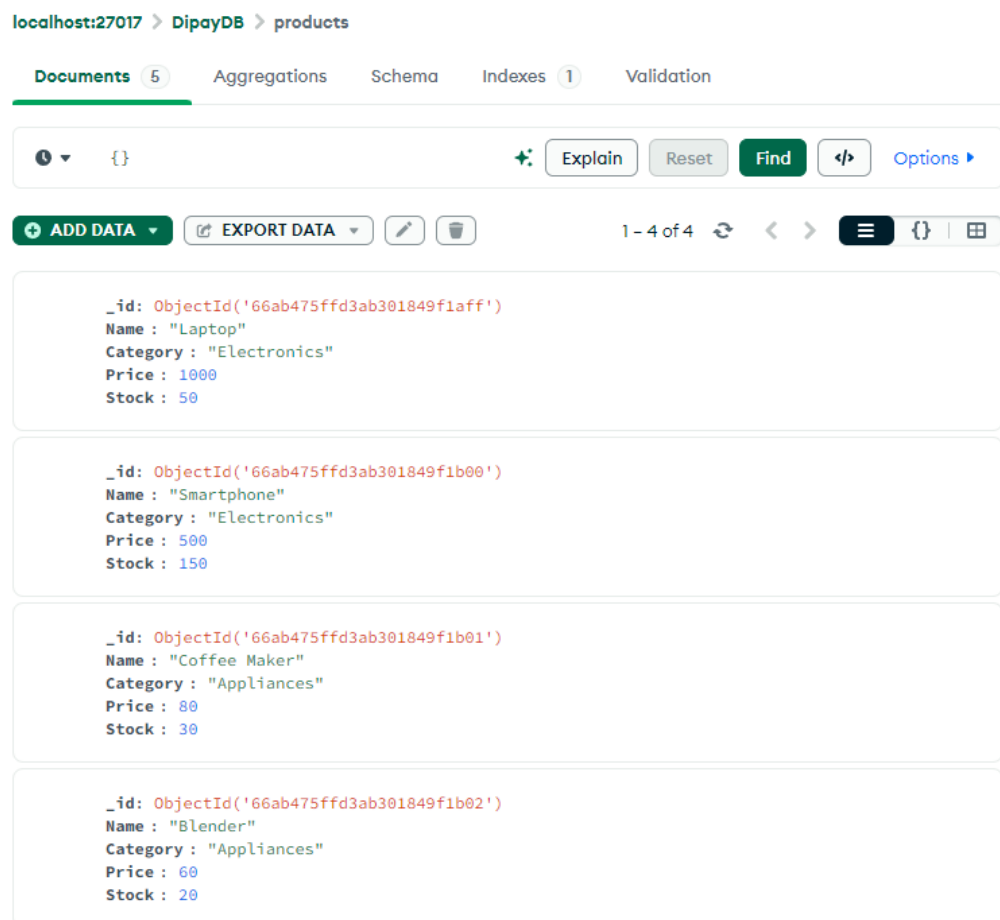


5. DELETE a product

- Postman



- Database



6. GET count by category

HTTP Dipay / GET count by category

GET http://localhost:5000/api/products/count-by-category

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results 200 OK 8 ms 236 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "category": "Electronics",
4     "productCount": 2
5   },
6   {
7     "category": "Appliances",
8     "productCount": 2
9   }
10 }
```

7. GET average price by category

HTTP Dipay / GET average price by category

GET http://localhost:5000/api/products/average-price-by-category

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results 200 OK 4 ms 239 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "category": "Electronics",
4     "averagePrice": 750
5   },
6   {
7     "category": "Appliances",
8     "averagePrice": 70
9   }
10 }
```

8. GET total value by category

HTTP Dipay / GET total value by category Save

GET ⌵ http://localhost:5000/api/products/total-value-by-category Send ⌵

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results 200 OK 3 ms 240 B Save as Example ⋮

Pretty Raw Preview Visualize JSON ⌵

```
1  [
2    {
3      "category": "Electronics",
4      "totalValue": 125000
5    },
6    {
7      "category": "Appliances",
8      "totalValue": 3600
9    }
10 ]
```