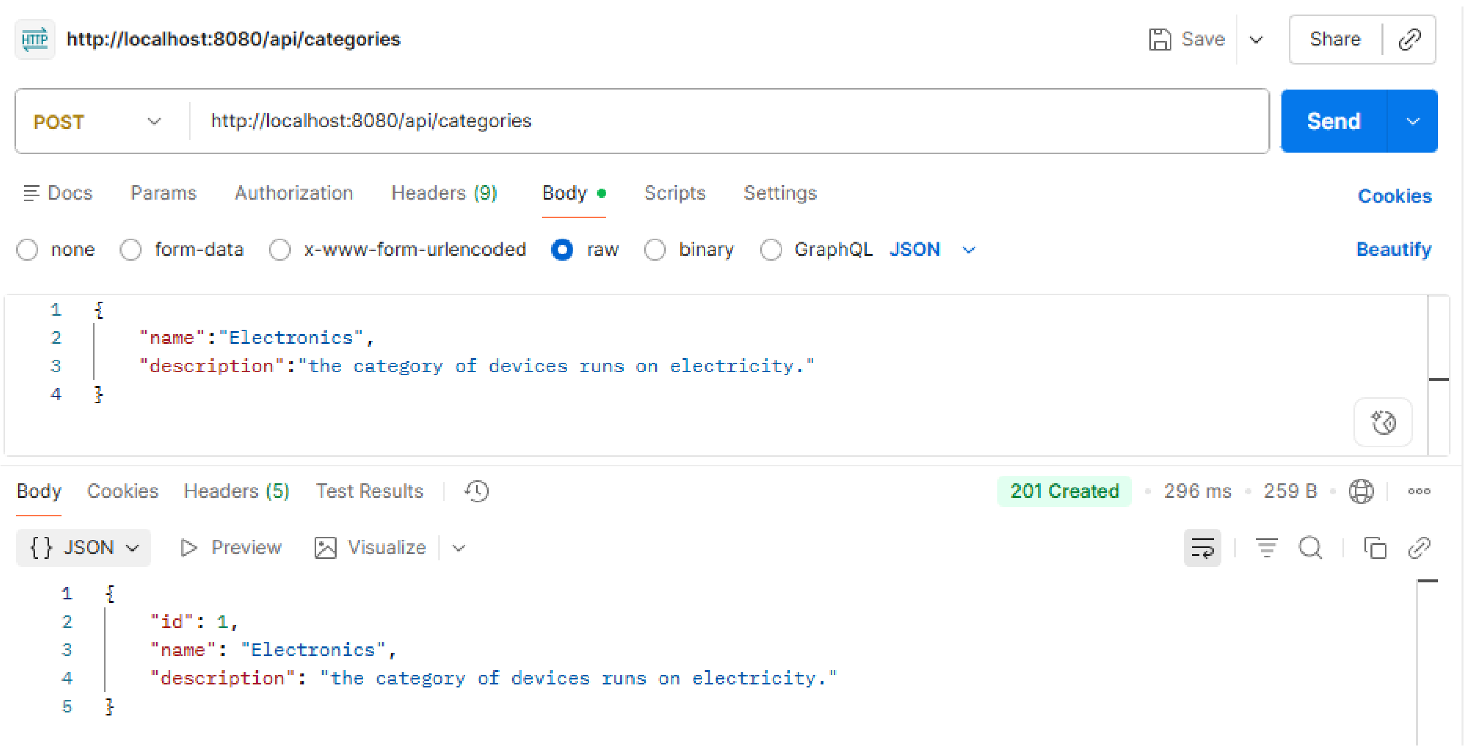
Postman Screenshots

1) Category endpoint(/api/categories)

a) Creating a category



POST http://localhost:8080/api/categories

Body: [✓ raw] JSON

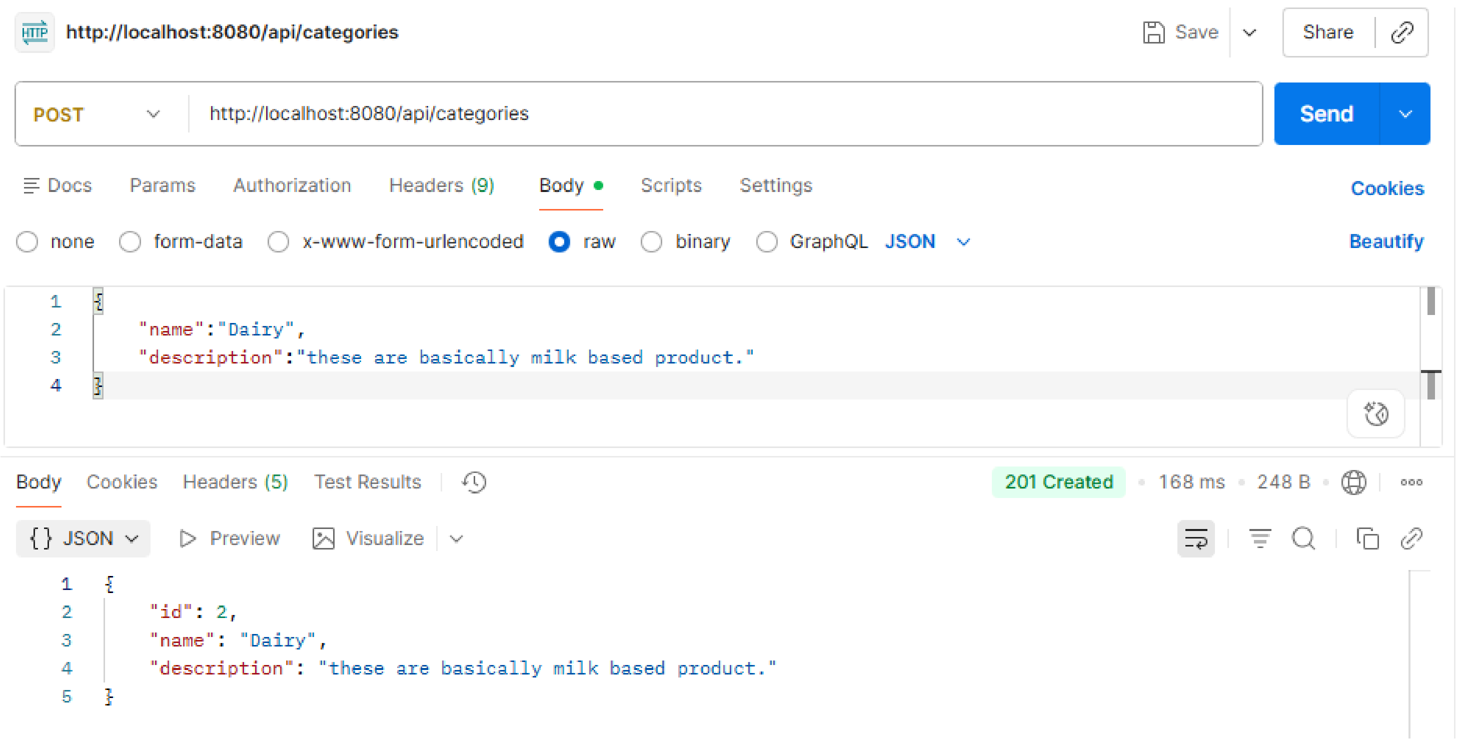
```
{
    "name":"Electronics",
    "description":"the category of devices runs on electricity."
}
```

201 Created • 296 ms • 259 B

JSON

```
{
    "id": 1,
    "name": "Electronics",
    "description": "the category of devices runs on electricity."
}
```

b) Creating second category



POST http://localhost:8080/api/categories

Body: [✓ raw] JSON

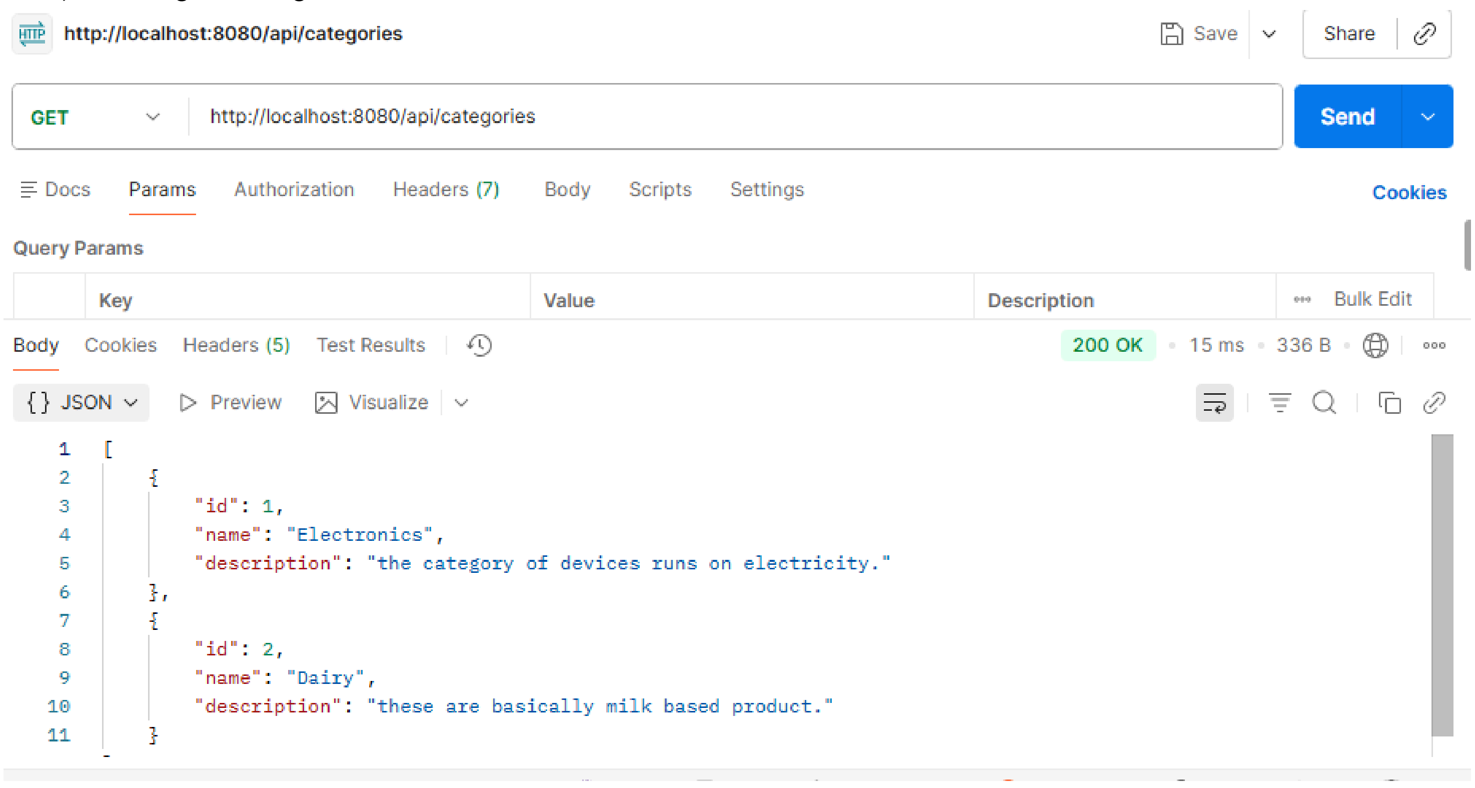
```
{
    "name":"Dairy",
    "description":"these are basically milk based product."
}
```

201 Created • 168 ms • 248 B

JSON

```
{
    "id": 2,
    "name": "Dairy",
    "description": "these are basically milk based product."
}
```

c) Getting all categories



http://localhost:8080/api/categories

GET http://localhost:8080/api/categories

Query Params

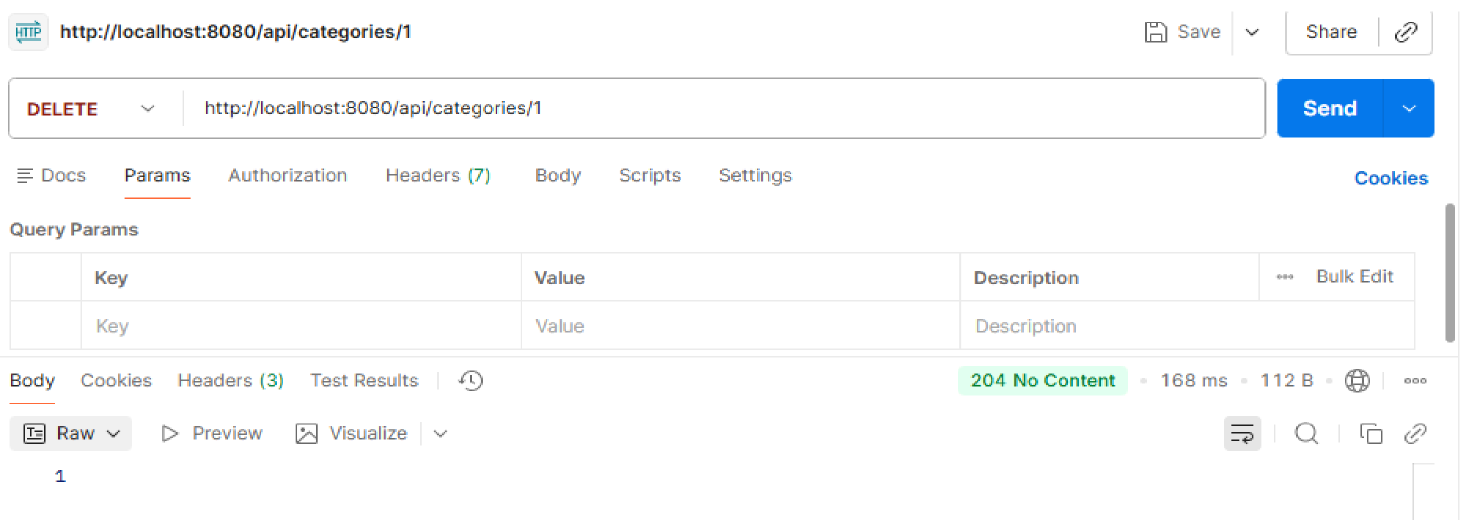
Key	Value	Description

Body | Cookies | Headers (5) | Test Results — 200 OK • 15 ms • 336 B

JSON

```
[
    {
        "id": 1,
        "name": "Electronics",
        "description": "the category of devices runs on electricity."
    },
    {
        "id": 2,
        "name": "Dairy",
        "description": "these are basically milk based product."
    }
```

d) Deleting a category



http://localhost:8080/api/categories/1

DELETE http://localhost:8080/api/categories/1

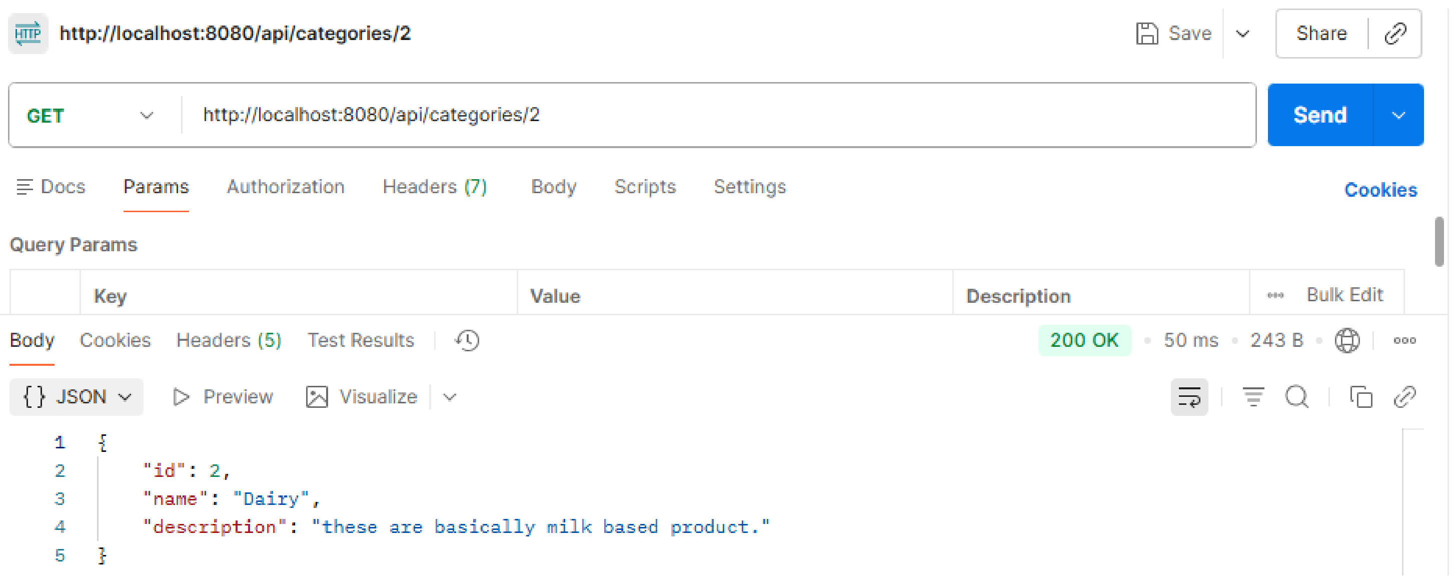
Query Params

Key	Value	Description
Key	Value	Description

Body | Cookies | Headers (3) | Test Results — 204 No Content • 168 ms • 112 B

Raw

e) Fetching the category



http://localhost:8080/api/categories/2

GET http://localhost:8080/api/categories/2

Query Params

Key	Value	Description

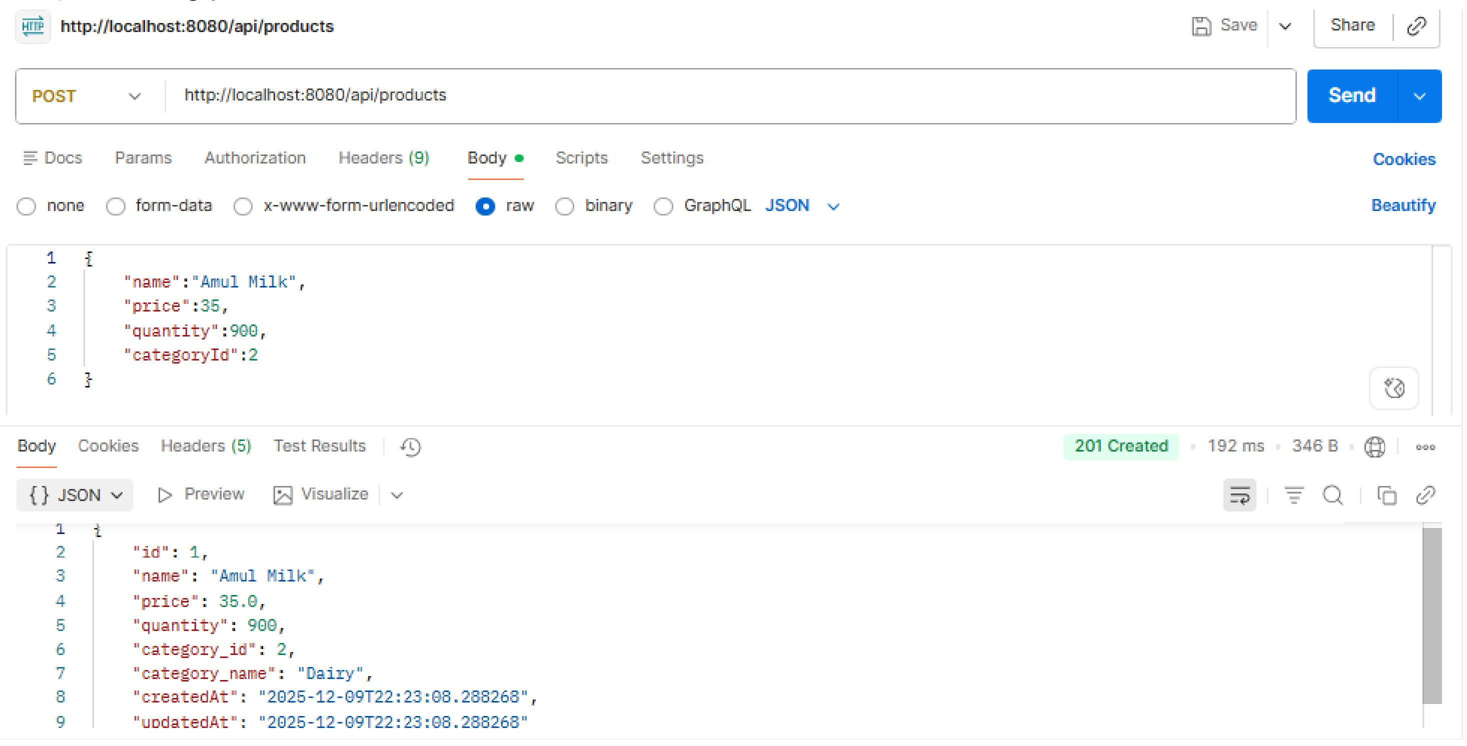
Body | Cookies | Headers (5) | Test Results — 200 OK • 50 ms • 243 B

JSON

```
{
    "id": 2,
    "name": "Dairy",
    "description": "these are basically milk based product."
}
```

2) Product Endpoints(/api/products)

a) Creating products



POST http://localhost:8080/api/products

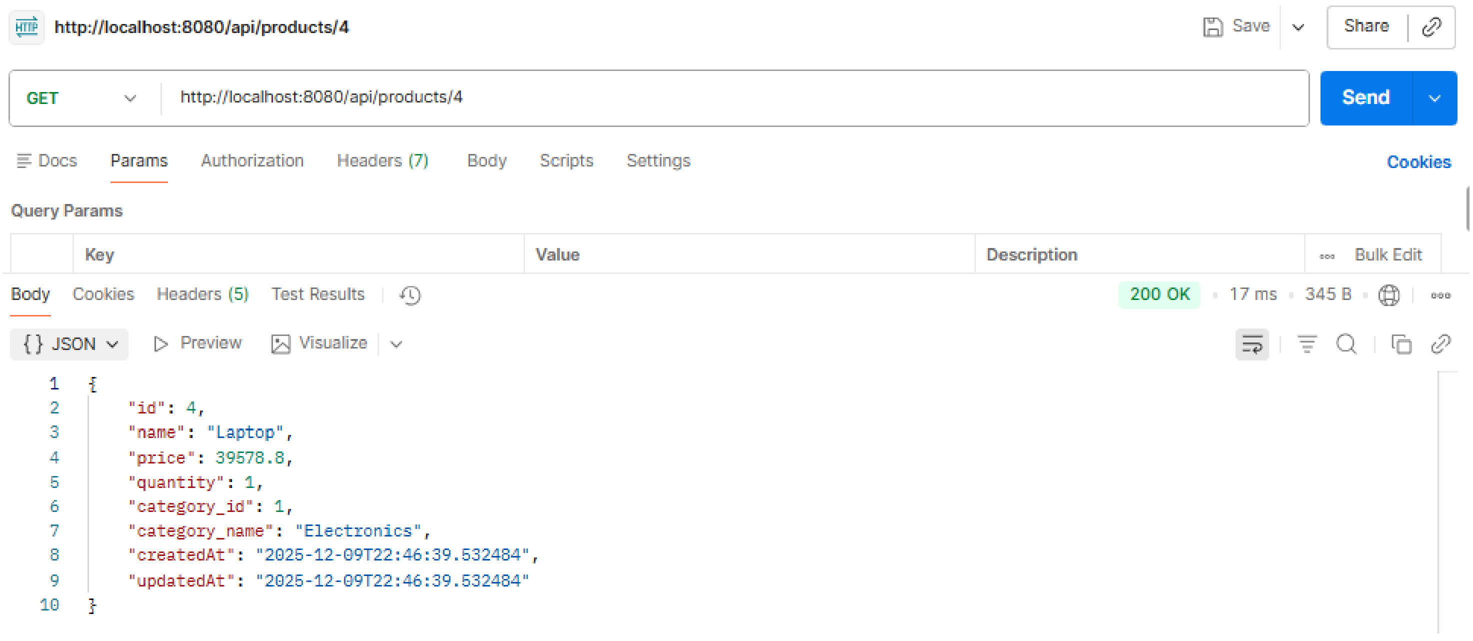
```
{
    "name":"Amul Milk",
    "price":35,
    "quantity":900,
    "categoryId":2
}
```

201 Created · 192 ms · 346 B

JSON

```
{
    "id": 1,
    "name": "Amul Milk",
    "price": 35.0,
    "quantity": 900,
    "category_id": 2,
    "category_name": "Dairy",
    "createdAt": "2025-12-09T22:23:08.288268",
    "updatedAt": "2025-12-09T22:23:08.288268"
```

b) Fetching product



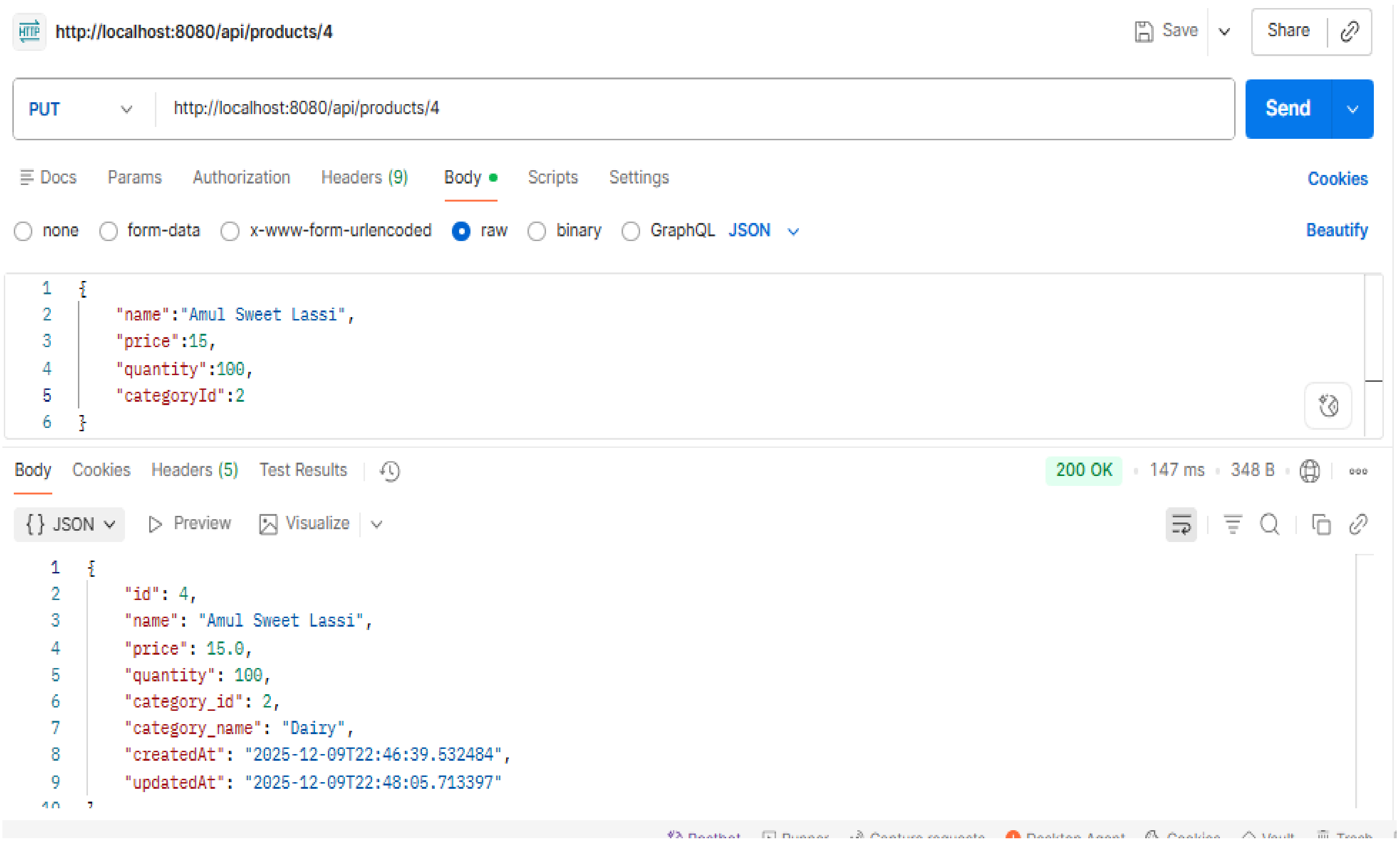
GET http://localhost:8080/api/products/4

200 OK · 17 ms · 345 B

JSON

```
{
    "id": 4,
    "name": "Laptop",
    "price": 39578.8,
    "quantity": 1,
    "category_id": 1,
    "category_name": "Electronics",
    "createdAt": "2025-12-09T22:46:39.532484",
    "updatedAt": "2025-12-09T22:46:39.532484"
}
```

c) <u>Updating a product</u>



PUT http://localhost:8080/api/products/4

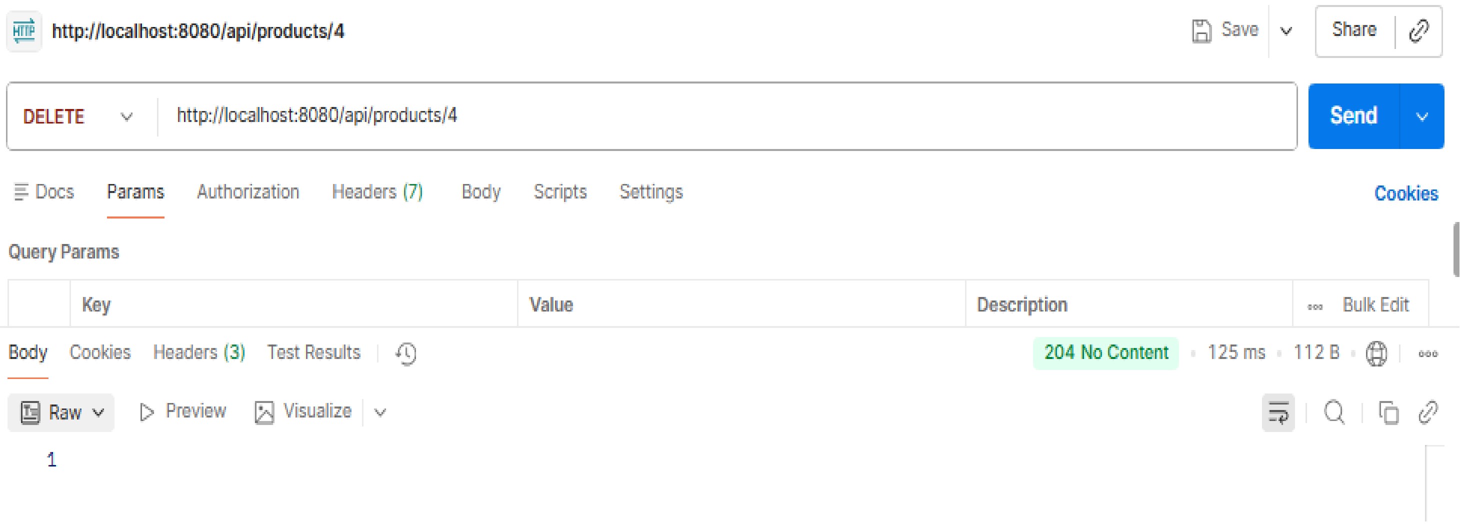
Body: none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL — JSON

```
{
    "name":"Amul Sweet Lassi",
    "price":15,
    "quantity":100,
    "categoryId":2
}
```

200 OK · 147 ms · 348 B

```
{
    "id": 4,
    "name": "Amul Sweet Lassi",
    "price": 15.0,
    "quantity": 100,
    "category_id": 2,
    "category_name": "Dairy",
    "createdAt": "2025-12-09T22:46:39.532484",
    "updatedAt": "2025-12-09T22:48:05.713397"
}
```

d) <u>Deleting a product</u>

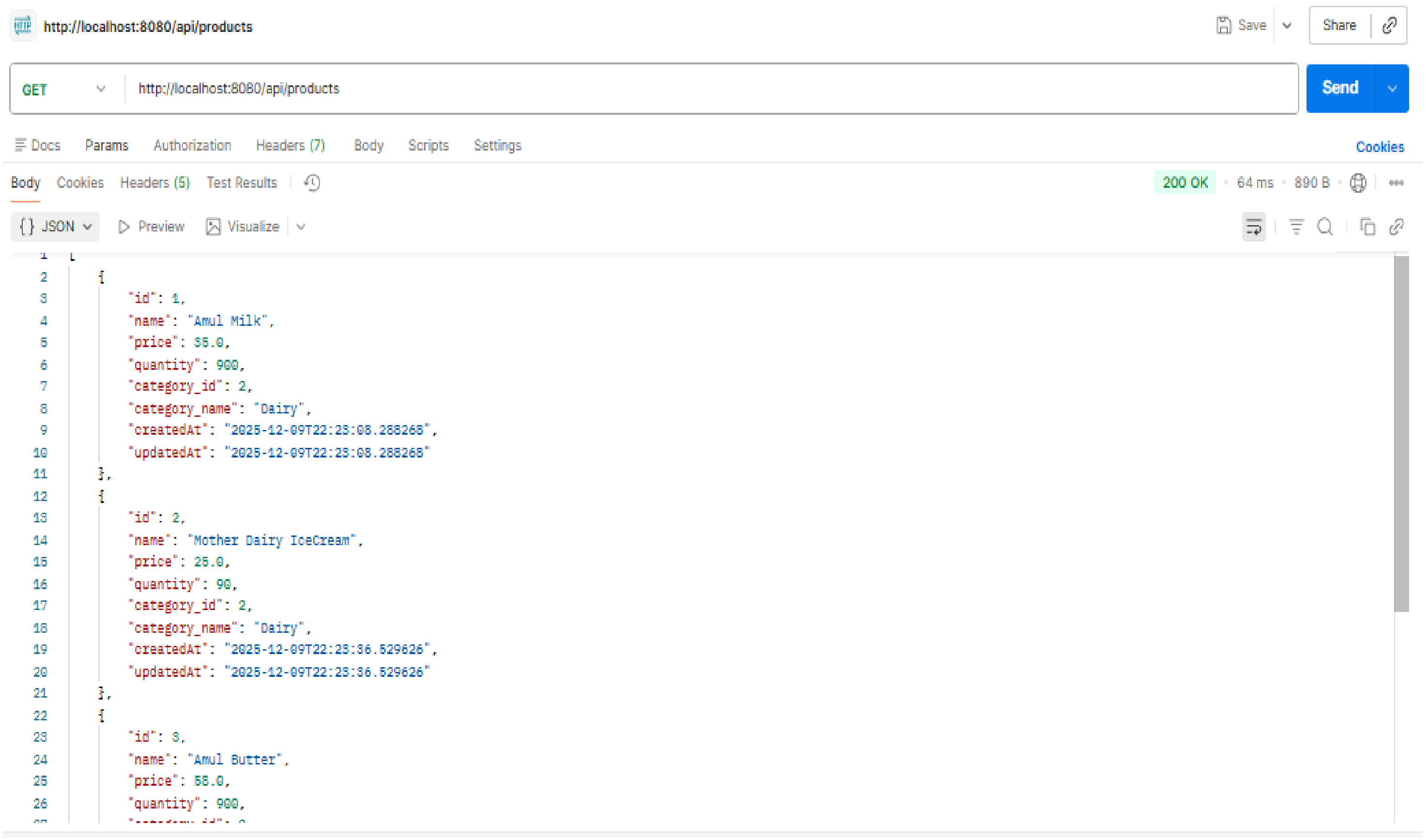


DELETE http://localhost:8080/api/products/4

204 No Content · 125 ms · 112 B

e) Fetching using filters

1) Fetching all the products(all filters null)

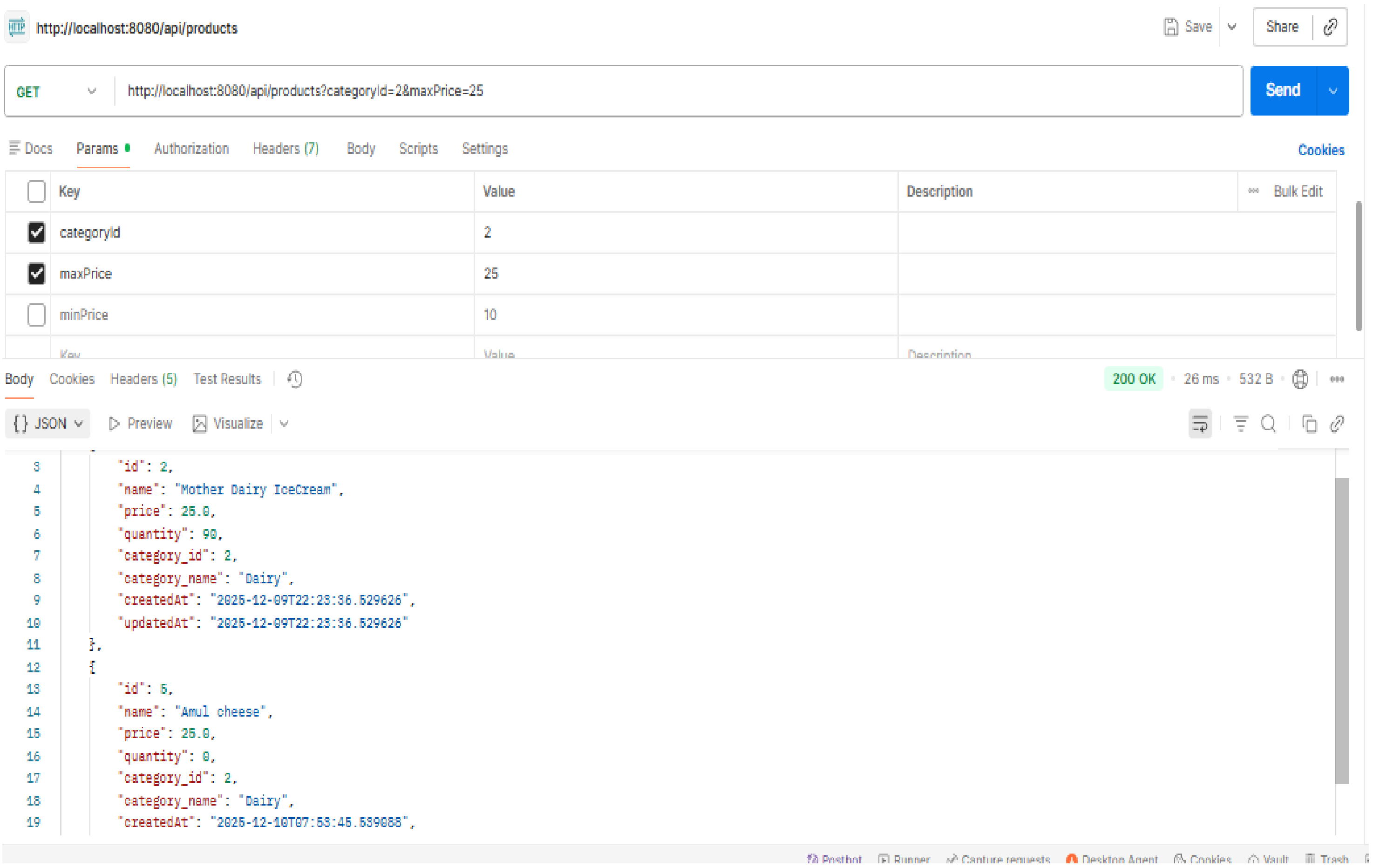


http://localhost:8080/api/products

GET http://localhost:8080/api/products

```
[
    {
        "id": 1,
        "name": "Amul Milk",
        "price": 35.0,
        "quantity": 900,
        "category_id": 2,
        "category_name": "Dairy",
        "createdAt": "2025-12-09T22:23:08.288268",
        "updatedAt": "2025-12-09T22:23:08.288268"
    },
    {
        "id": 2,
        "name": "Mother Dairy IceCream",
        "price": 25.0,
        "quantity": 90,
        "category_id": 2,
        "category_name": "Dairy",
        "createdAt": "2025-12-09T22:23:36.529626",
        "updatedAt": "2025-12-09T22:23:36.529626"
    },
    {
        "id": 3,
        "name": "Amul Butter",
        "price": 58.0,
        "quantity": 900,
```

2) Filter(category:2 , maxprice :25)



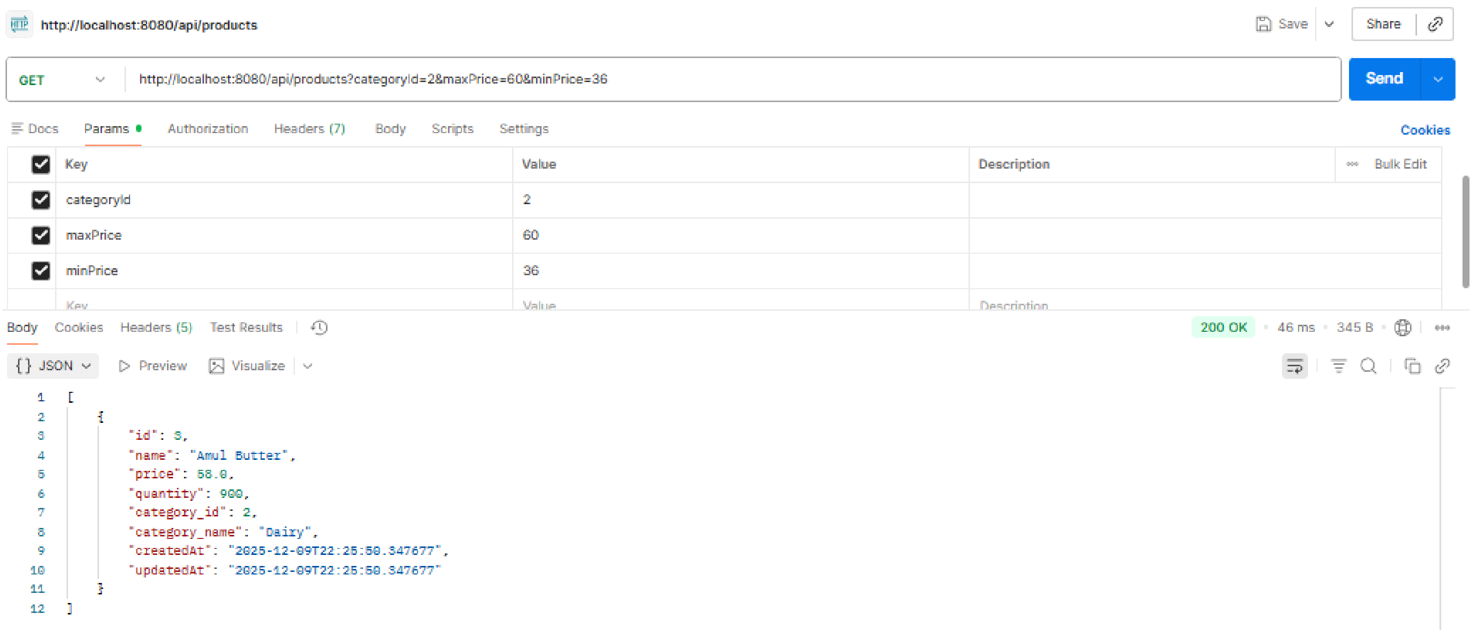
http://localhost:8080/api/products

GET http://localhost:8080/api/products?categoryId=2&maxPrice=25

	Key	Value	Description
✓	categoryId	2	
✓	maxPrice	25	
☐	minPrice	10	
	Key	Value	Description

```
        "id": 2,
        "name": "Mother Dairy IceCream",
        "price": 25.0,
        "quantity": 90,
        "category_id": 2,
        "category_name": "Dairy",
        "createdAt": "2025-12-09T22:23:36.529626",
        "updatedAt": "2025-12-09T22:23:36.529626"
    },
    {
        "id": 5,
        "name": "Amul cheese",
        "price": 25.0,
        "quantity": 0,
        "category_id": 2,
        "category_name": "Dairy",
        "createdAt": "2025-12-10T07:53:45.539088",
```

3) <u>With All Three filters</u>



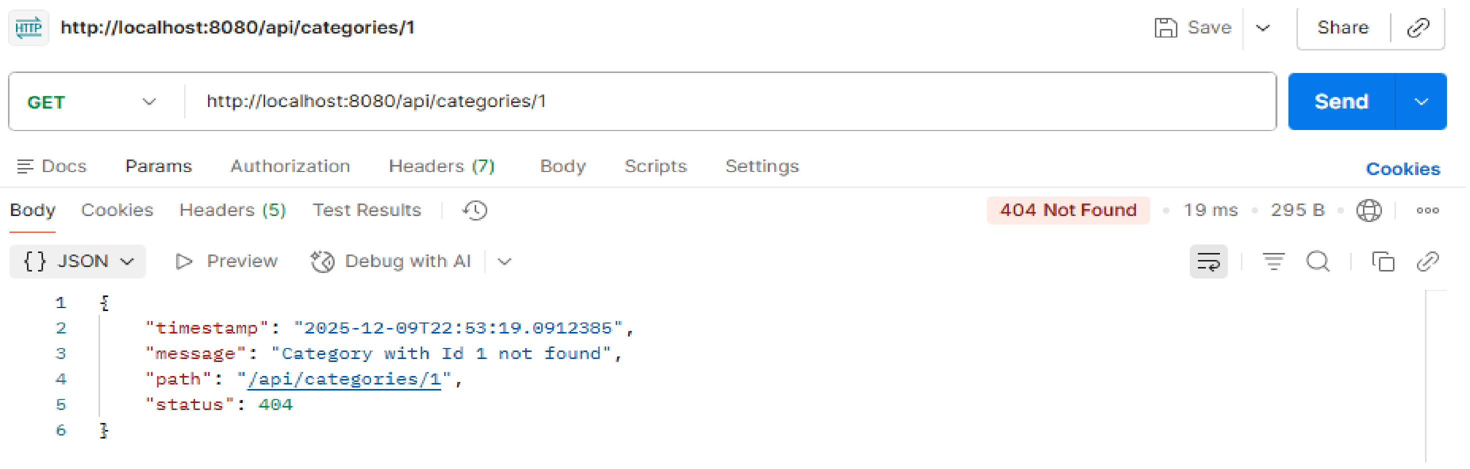
```
http://localhost:8080/api/products?categoryId=2&maxPrice=60&minPrice=36
```

	Key	Value	Description
✓	categoryId	2	
✓	maxPrice	60	
✓	minPrice	36	
	Key	Value	Description

```
[
    {
        "id": 3,
        "name": "Amul Butter",
        "price": 58.0,
        "quantity": 900,
        "category_id": 2,
        "category_name": "Dairy",
        "createdAt": "2025-12-09T22:25:50.347677",
        "updatedAt": "2025-12-09T22:25:50.347677"
    }
]
```

3) Validation and Exception

a) <u>Fetching category after deleting it</u>



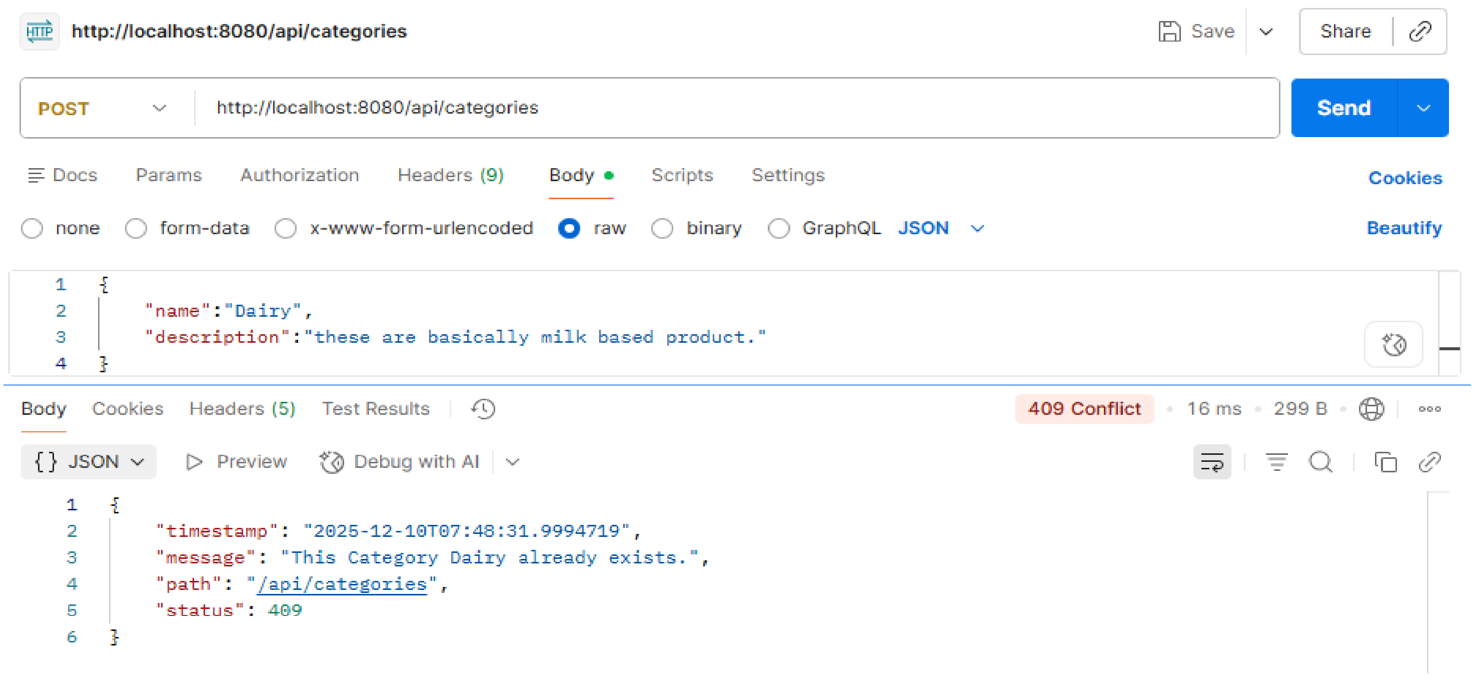
http://localhost:8080/api/categories/1

GET http://localhost:8080/api/categories/1

404 Not Found · 19 ms · 295 B

```
{
    "timestamp": "2025-12-09T22:53:19.0912385",
    "message": "Category with Id 1 not found",
    "path": "/api/categories/1",
    "status": 404
}
```

b) <u>Creating duplicate category</u>



http://localhost:8080/api/categories

POST http://localhost:8080/api/categories

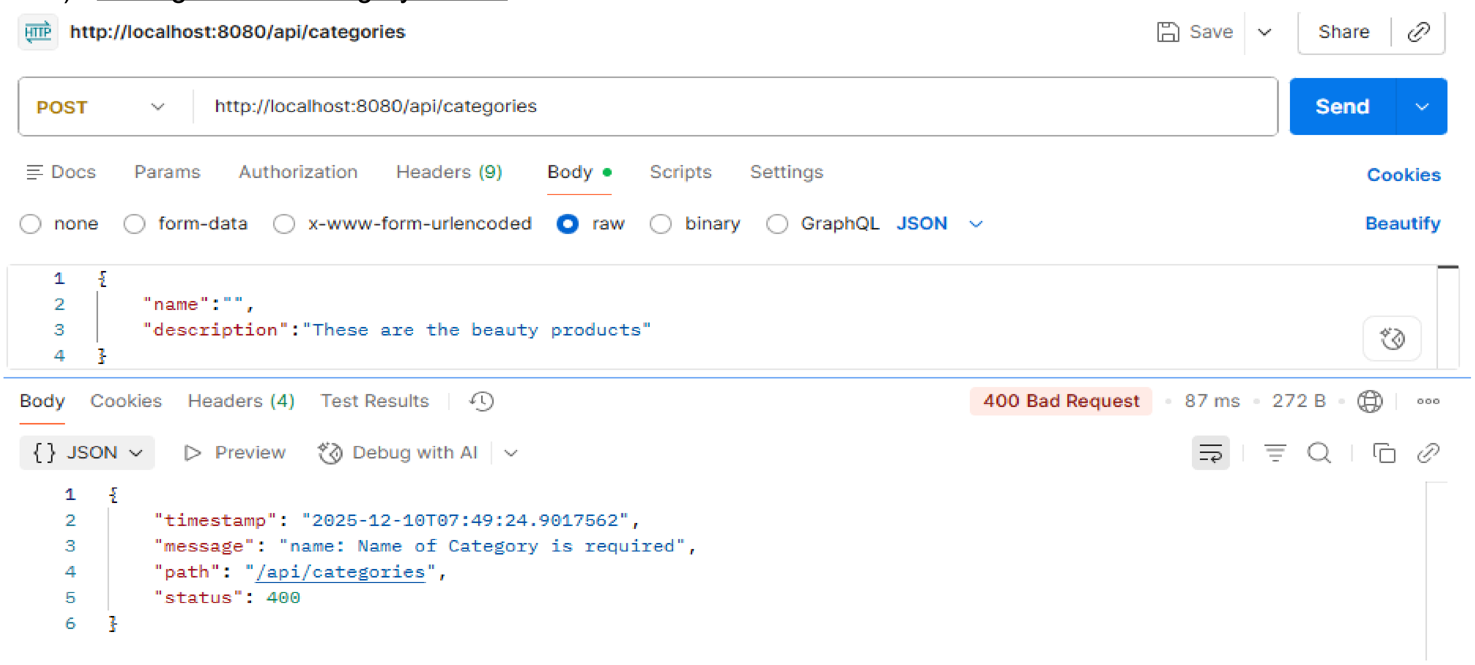
[✓ raw] JSON

```
{
    "name":"Dairy",
    "description":"these are basically milk based product."
}
```

409 Conflict · 16 ms · 299 B

```
{
    "timestamp": "2025-12-10T07:48:31.9994719",
    "message": "This Category Dairy already exists.",
    "path": "/api/categories",
    "status": 409
}
```

c) <u>Putting nameof category as null</u>



http://localhost:8080/api/categories

POST http://localhost:8080/api/categories

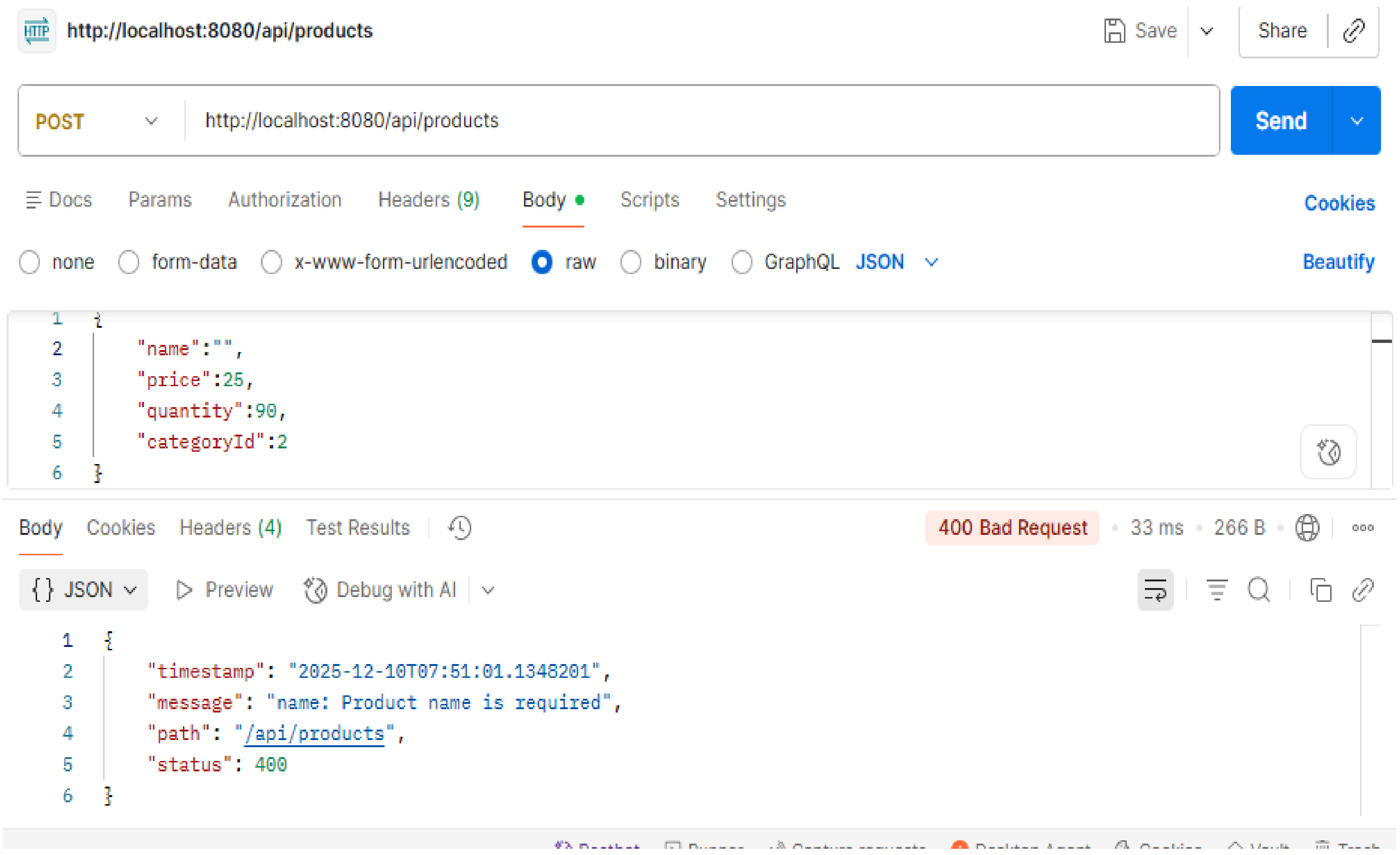
[✓ raw] JSON

```
{
    "name":"",
    "description":"These are the beauty products"
}
```

400 Bad Request · 87 ms · 272 B

```
{
    "timestamp": "2025-12-10T07:49:24.9017562",
    "message": "name: Name of Category is required",
    "path": "/api/categories",
    "status": 400
}
```

d) Putting name of Product as null



http://localhost:8080/api/products

POST http://localhost:8080/api/products

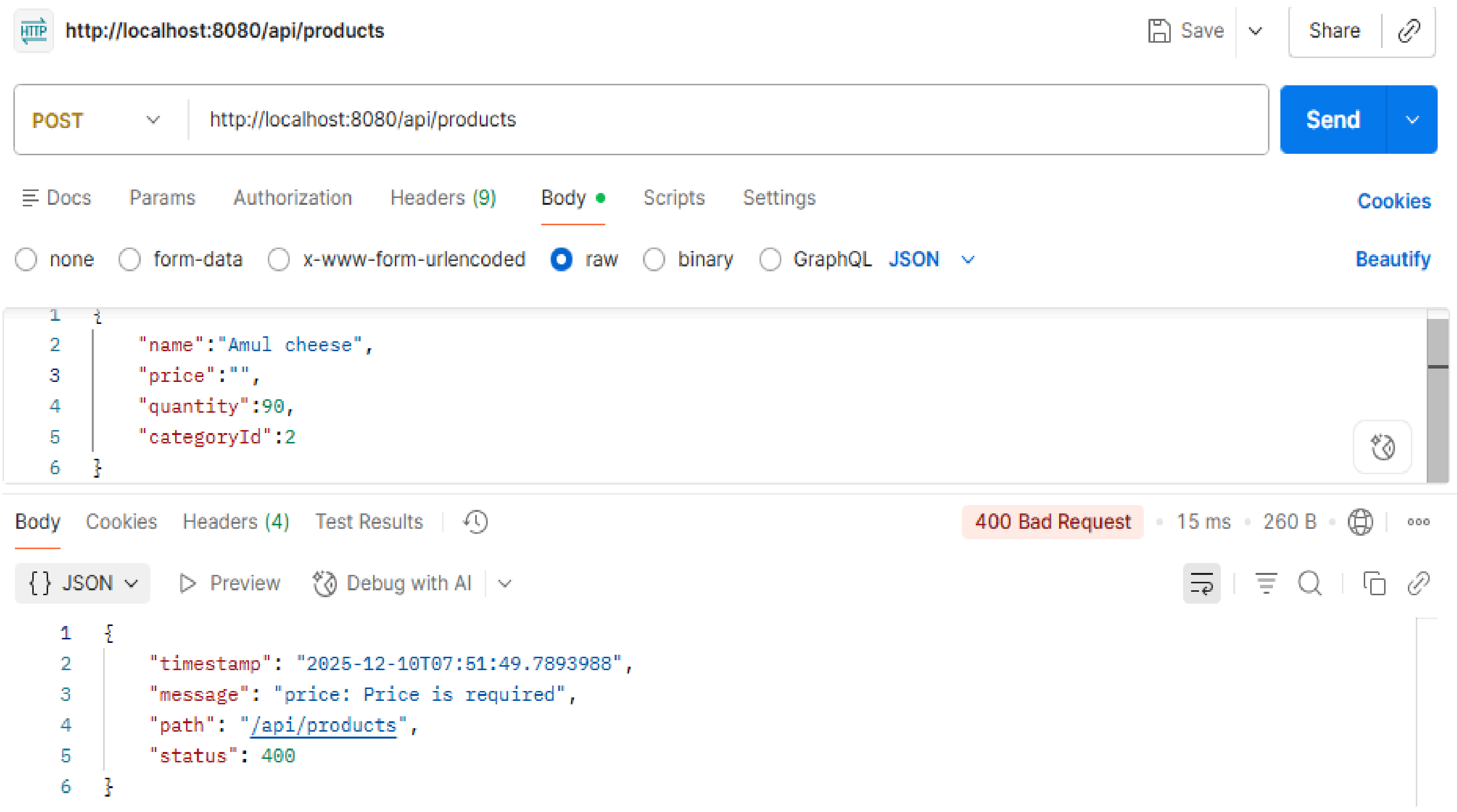
Body: raw JSON

```
{
    "name":"",
    "price":25,
    "quantity":90,
    "categoryId":2
}
```

400 Bad Request • 33 ms • 266 B

```
{
    "timestamp": "2025-12-10T07:51:01.1348201",
    "message": "name: Product name is required",
    "path": "/api/products",
    "status": 400
}
```

e) Putting Price of product as null



http://localhost:8080/api/products

POST http://localhost:8080/api/products

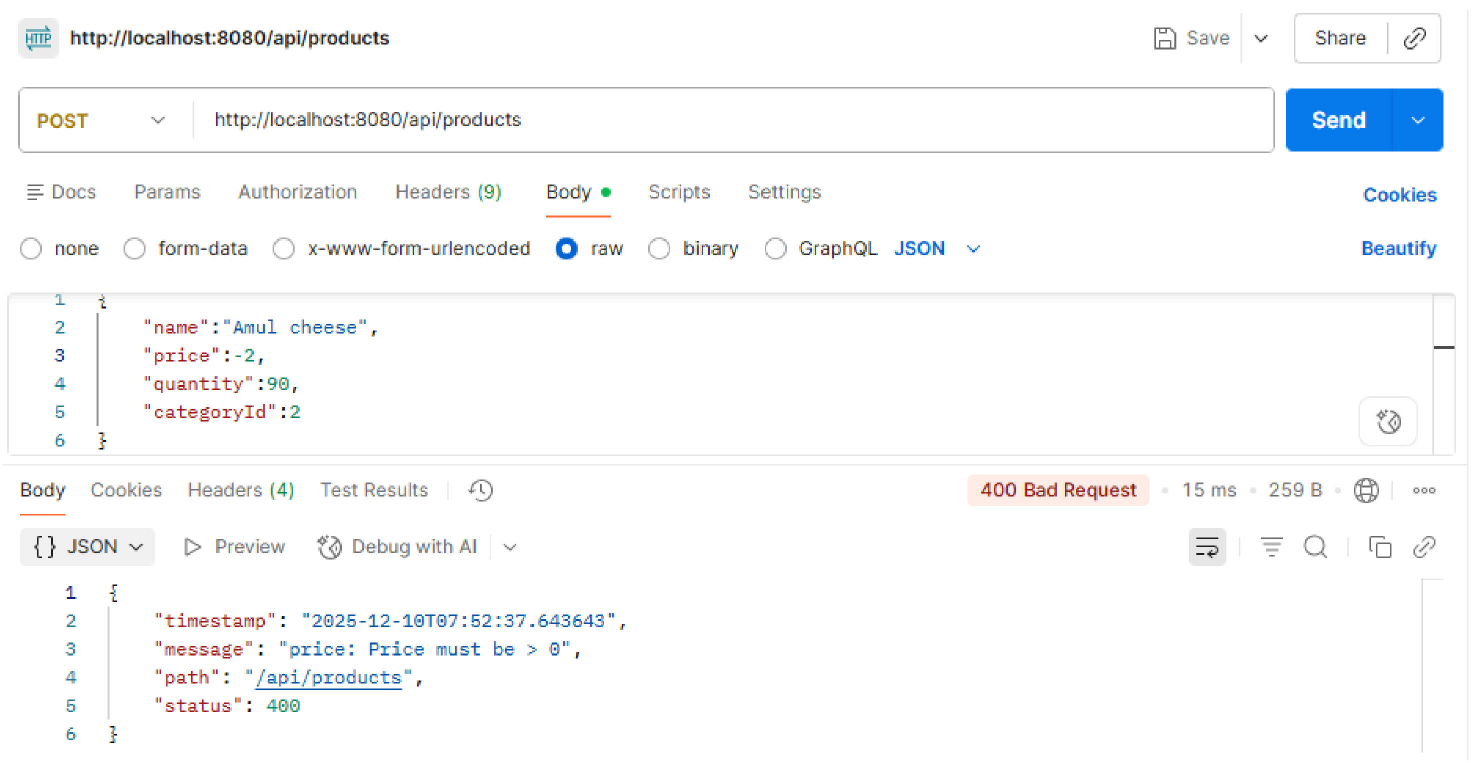
Body: raw JSON

```
{
    "name":"Amul cheese",
    "price":"",
    "quantity":90,
    "categoryId":2
}
```

400 Bad Request • 15 ms • 260 B

```
{
    "timestamp": "2025-12-10T07:51:49.7893988",
    "message": "price: Price is required",
    "path": "/api/products",
    "status": 400
}
```

f) Price < 0 (Negative price)



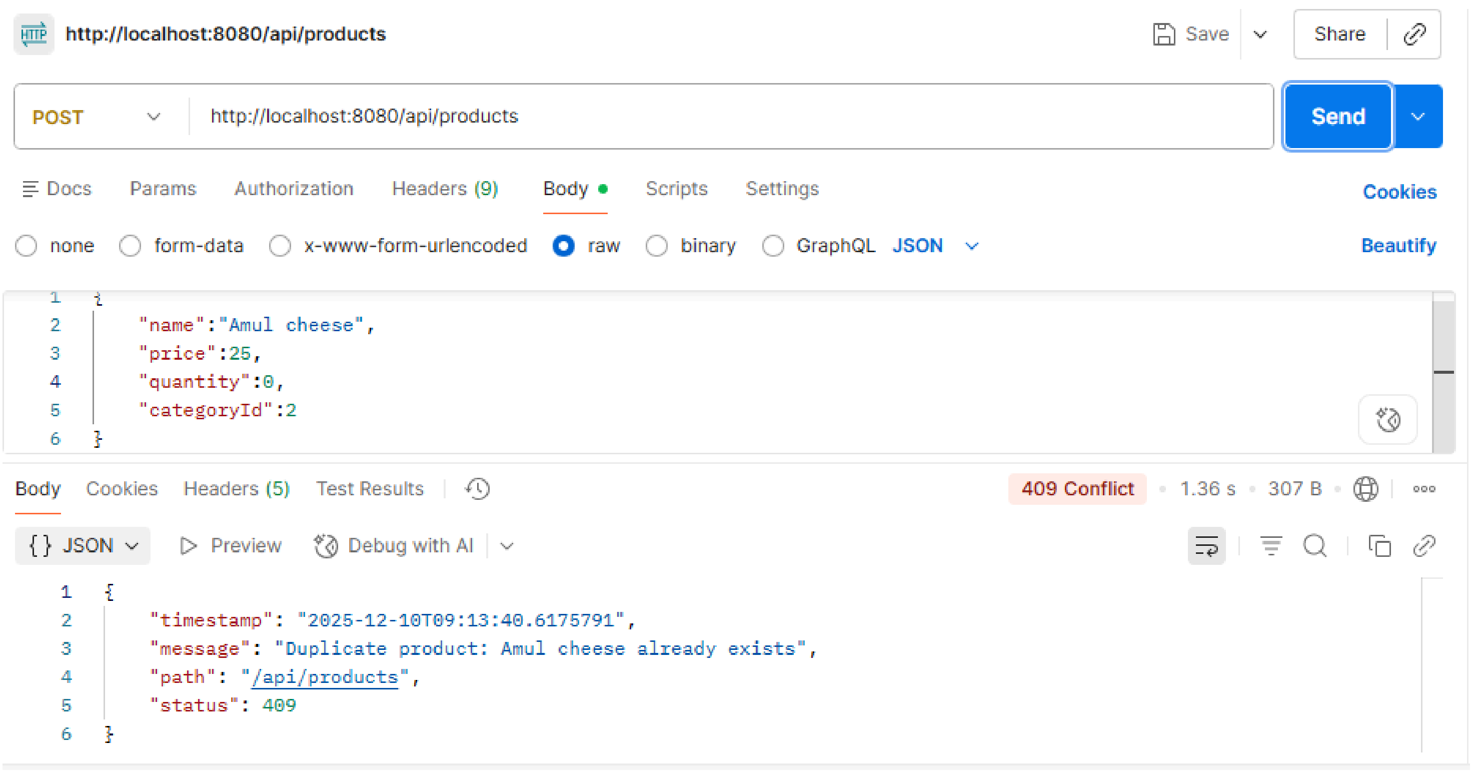
POST http://localhost:8080/api/products

```
{
    "name":"Amul cheese",
    "price":-2,
    "quantity":90,
    "categoryId":2
}
```

400 Bad Request

```
{
    "timestamp": "2025-12-10T07:52:37.643643",
    "message": "price: Price must be > 0",
    "path": "/api/products",
    "status": 400
}
```

g) Trying to add duplicate product



POST http://localhost:8080/api/products

```
{
    "name":"Amul cheese",
    "price":25,
    "quantity":0,
    "categoryId":2
}
```

409 Conflict

```
{
    "timestamp": "2025-12-10T09:13:40.6175791",
    "message": "Duplicate product: Amul cheese already exists",
    "path": "/api/products",
    "status": 409
}
```