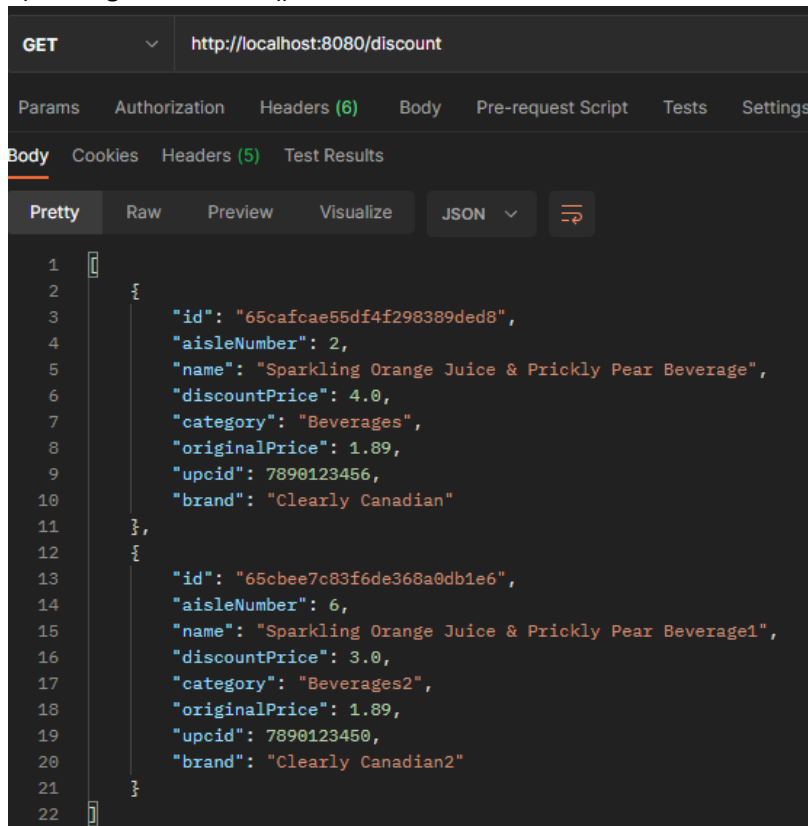


TEST Report:

1. Discount:

a) findingAllDiscounts () ->

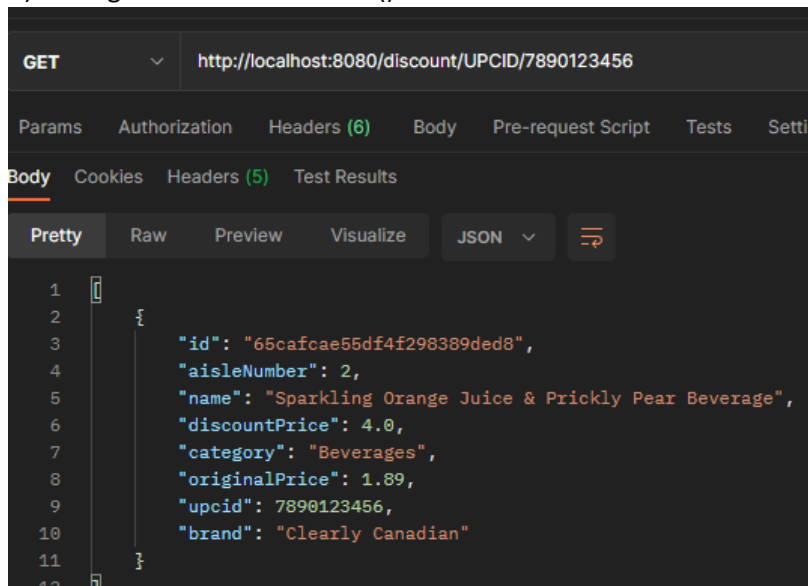


```
GET http://localhost:8080/discount

Params Authorization Headers (6) Body Pre-request Script Tests Settings
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON ↕

1  [
2    {
3      "id": "65cafcae55df4f298389ded8",
4      "aisleNumber": 2,
5      "name": "Sparkling Orange Juice & Prickly Pear Beverage",
6      "discountPrice": 4.0,
7      "category": "Beverages",
8      "originalPrice": 1.89,
9      "upcid": 7890123456,
10     "brand": "Clearly Canadian"
11   },
12   {
13     "id": "65cbee7c83f6de368a0db1e6",
14     "aisleNumber": 6,
15     "name": "Sparkling Orange Juice & Prickly Pear Beverage1",
16     "discountPrice": 3.0,
17     "category": "Beverages2",
18     "originalPrice": 1.89,
19     "upcid": 7890123450,
20     "brand": "Clearly Canadian2"
21   }
22 ]
```

b) findingDiscountsWithUPCID () ->



```
GET http://localhost:8080/discount/UPCID/7890123456

Params Authorization Headers (6) Body Pre-request Script Tests Settings
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON ↕

1  {
2    "id": "65cafcae55df4f298389ded8",
3    "aisleNumber": 2,
4    "name": "Sparkling Orange Juice & Prickly Pear Beverage",
5    "discountPrice": 4.0,
6    "category": "Beverages",
7    "originalPrice": 1.89,
8    "upcid": 7890123456,
9    "brand": "Clearly Canadian"
10  }
```

c) updateDiscountPrice () ->

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/discount/7890123456`. The 'Body' tab is selected, showing a JSON payload. The 'Pretty' view is active, displaying the JSON structure. The payload contains two objects, each representing a beverage item with its discount price updated.

```
PUT http://localhost:8080/discount/7890123456

{"discountPrice": 7.0}

[
  {
    "id": "65cafcac55df4f298389ded8",
    "aisleNumber": 2,
    "name": "Sparkling Orange Juice & Prickly Pear Beverage",
    "discountPrice": 7.0,
    "category": "Beverages",
    "originalPrice": 1.89,
    "upcid": 7890123456,
    "brand": "Clearly Canadian"
  },
  {
    "id": "65cbee7c83f6de368a0db1e6",
    "aisleNumber": 6,
    "name": "Sparkling Orange Juice & Prickly Pear Beverage1",
    "discountPrice": 3.0,
    "category": "Beverages2",
    "originalPrice": 1.89,
    "upcid": 7890123450,
    "brand": "Clearly Canadian2"
  }
]
```

2. Expiry :

a) findingAllExpiringProducts () ->

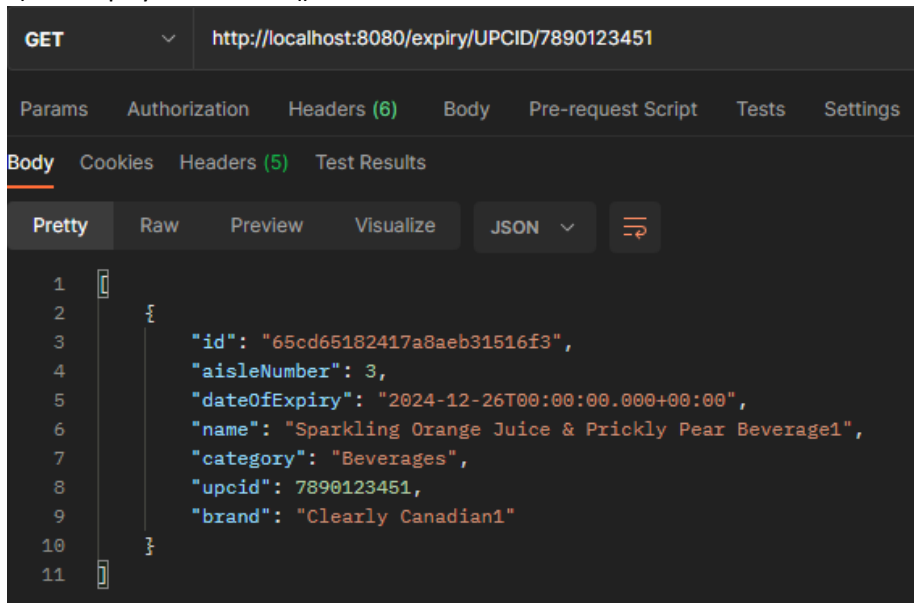
The screenshot shows a REST client interface with a GET request to `http://localhost:8080/expiry`. The 'Body' tab is selected, showing a JSON payload. The 'Pretty' view is active, displaying the JSON structure. The payload contains three objects, each representing a beverage item with its expiration date.

```
GET http://localhost:8080/expiry

[
  {
    "id": "65cd65182417a8aeb31516f3",
    "aisleNumber": 3,
    "dateOfExpiry": "2024-12-26T00:00:00.000+00:00",
    "name": "Sparkling Orange Juice & Prickly Pear Beverage1",
    "category": "Beverages",
    "upcid": 7890123451,
    "brand": "Clearly Canadian1"
  },
  {
    "id": "65cd65182417a8aeb31516f4",
    "aisleNumber": 4,
    "dateOfExpiry": "2024-12-24T00:00:00.000+00:00",
    "name": "Sparkling Orange Juice & Prickly Pear Beverage1",
    "category": "Beverages",
    "upcid": 7890123453,
    "brand": "Clearly Canadian111"
  },
  {
    "id": "65cd65182417a8aeb31516f2",
    "aisleNumber": 2,
    "dateOfExpiry": "2024-12-24T00:00:00.000+00:00",
    "name": "Sparkling Orange Juice & Prickly Pear Beverage",
    "category": "Beverages",
    "upcid": 7890123456,
    "brand": "Clearly Canadian"
  }
]
```

b) addExpiry() ->

c) findExpiryWithUPCID () ->



```
GET http://localhost:8080/expiry/UPCID/7890123451

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

Body Cookies Headers (5) Test Results

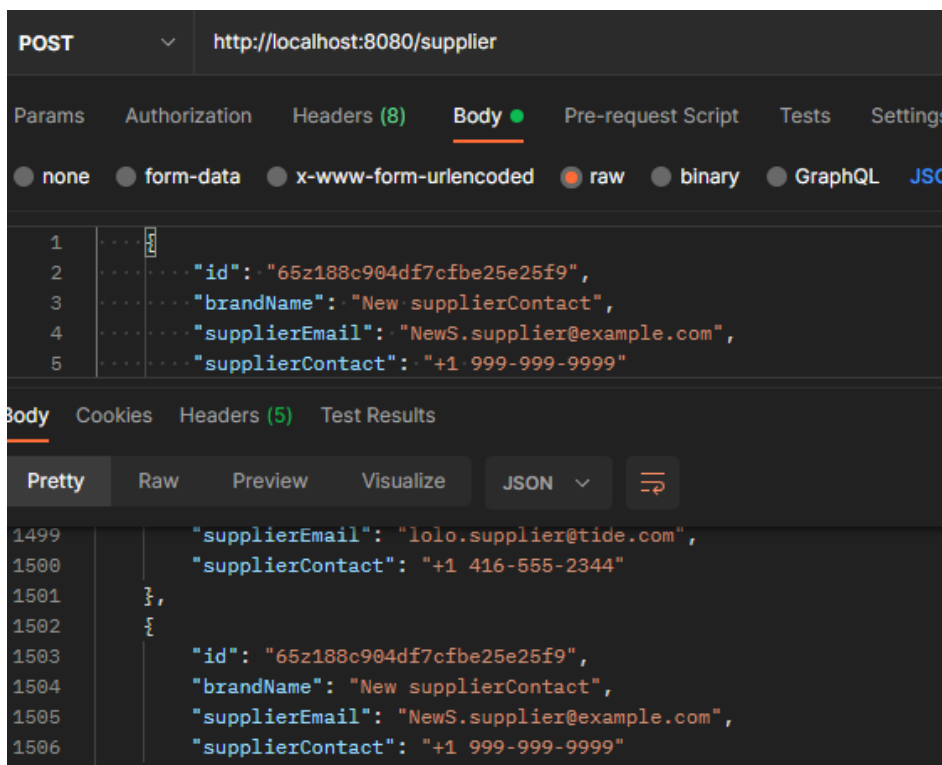
Pretty Raw Preview Visualize JSON

1 {
2   "id": "65cd65182417a8aeb31516f3",
3   "aisleNumber": 3,
4   "dateOfExpiry": "2024-12-26T00:00:00.000+00:00",
5   "name": "Sparkling Orange Juice & Prickly Pear Beverage1",
6   "category": "Beverages",
7   "upcid": 7890123451,
8   "brand": "Clearly Canadian1"
9 }
10
11
```

d) findExpiryWithDate () ->

3. Supplier:

a) addingBrand () ->



```
POST http://localhost:8080/supplier

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

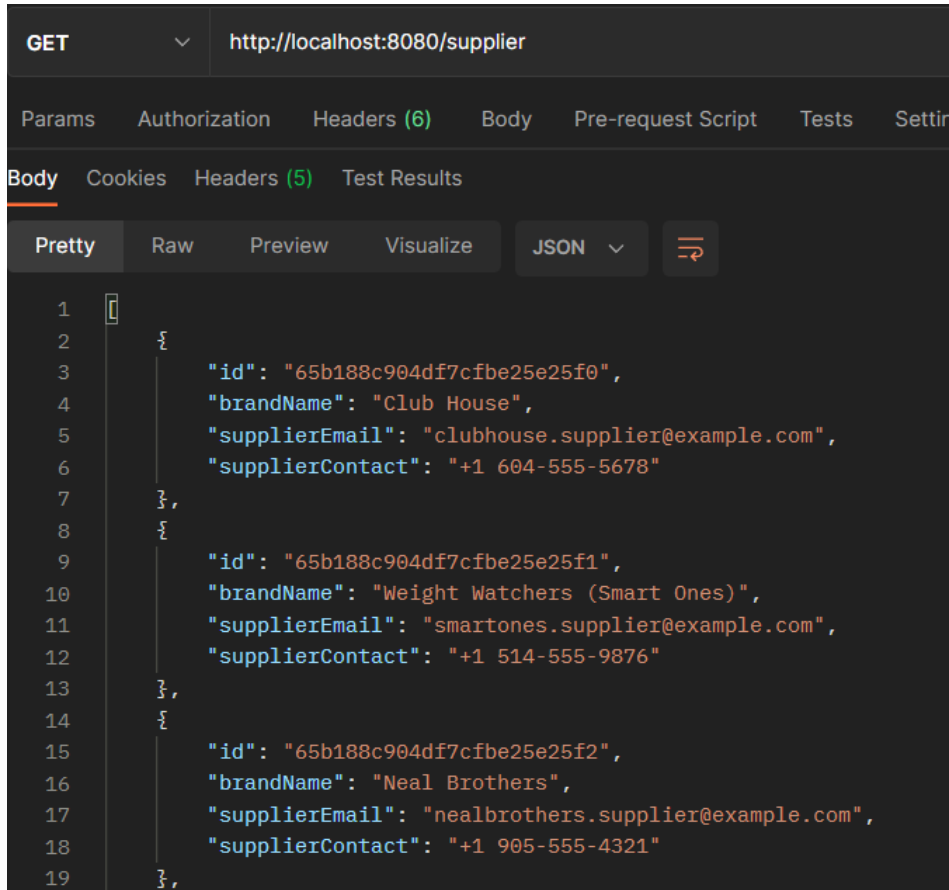
1 {
2   "id": "65z188c904df7cfbe25e25f9",
3   "brandName": "New supplierContact",
4   "supplierEmail": "NewS.supplier@example.com",
5   "supplierContact": "+1 999-999-9999"
6 }

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

1499 "supplierEmail": "lolo.supplier@tide.com",
1500 "supplierContact": "+1 416-555-2344"
1501 },
1502 {
1503   "id": "65z188c904df7cfbe25e25f9",
1504   "brandName": "New supplierContact",
1505   "supplierEmail": "NewS.supplier@example.com",
1506   "supplierContact": "+1 999-999-9999"
1507 }
```

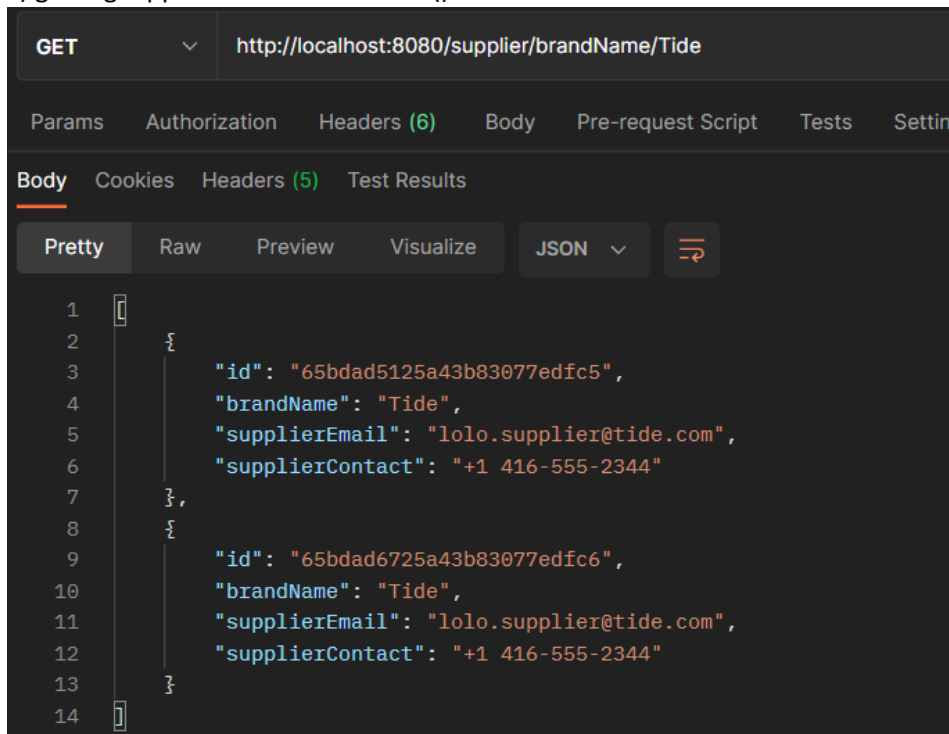
b) findAllBrands() ->



```
GET http://localhost:8080/supplier

Body
Pretty
1  [
2    {
3      "id": "65b188c904df7cfbe25e25f0",
4      "brandName": "Club House",
5      "supplierEmail": "clubhouse.supplier@example.com",
6      "supplierContact": "+1 604-555-5678"
7    },
8    {
9      "id": "65b188c904df7cfbe25e25f1",
10     "brandName": "Weight Watchers (Smart Ones)",
11     "supplierEmail": "smartones.supplier@example.com",
12     "supplierContact": "+1 514-555-9876"
13   },
14   {
15     "id": "65b188c904df7cfbe25e25f2",
16     "brandName": "Neal Brothers",
17     "supplierEmail": "nealbrothers.supplier@example.com",
18     "supplierContact": "+1 905-555-4321"
19   },
20 ]
```

c) gettingSuppliersWithBrandName () ->

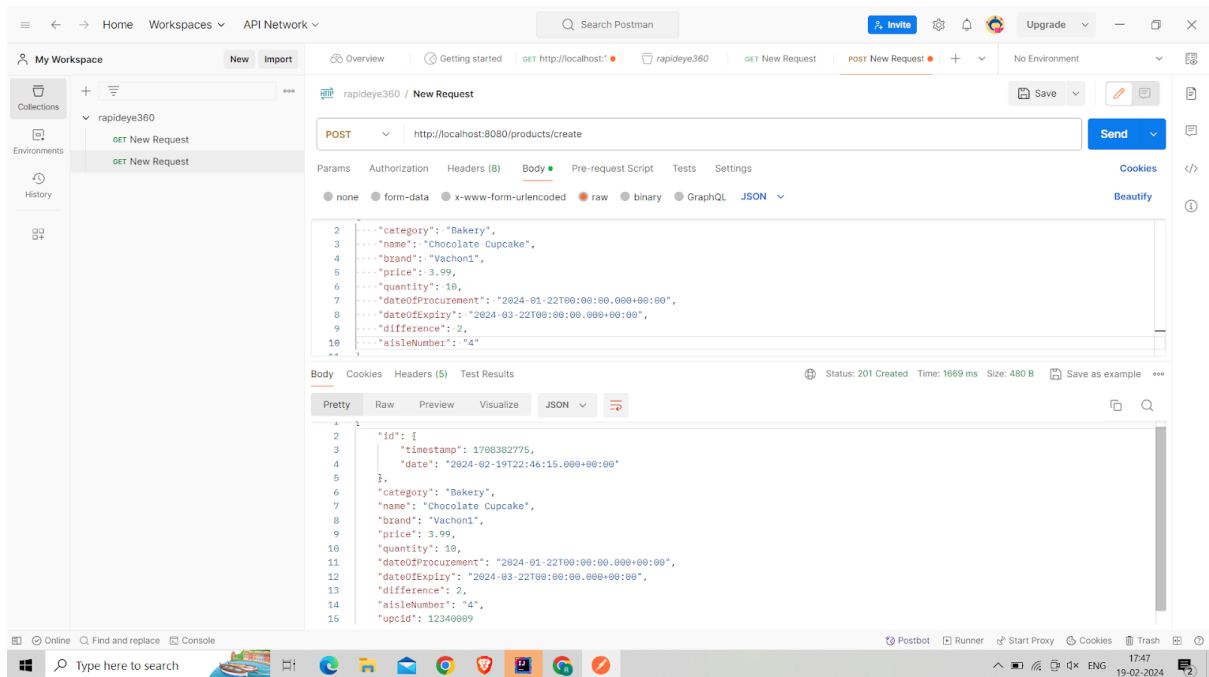


```
GET http://localhost:8080/supplier/brandName/Tide

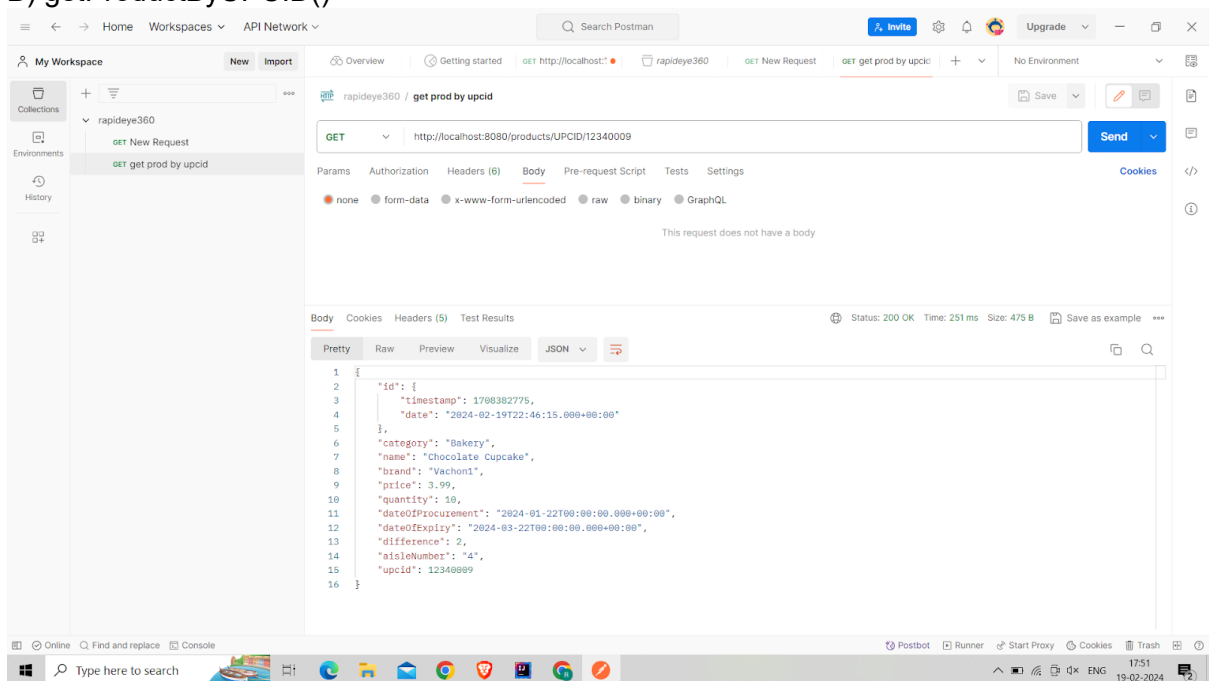
Body
Pretty
1  [
2    {
3      "id": "65bdad5125a43b83077edfc5",
4      "brandName": "Tide",
5      "supplierEmail": "lolo.supplier@tide.com",
6      "supplierContact": "+1 416-555-2344"
7    },
8    {
9      "id": "65bdad6725a43b83077edfc6",
10     "brandName": "Tide",
11     "supplierEmail": "lolo.supplier@tide.com",
12     "supplierContact": "+1 416-555-2344"
13   },
14 ]
```

4. Product :

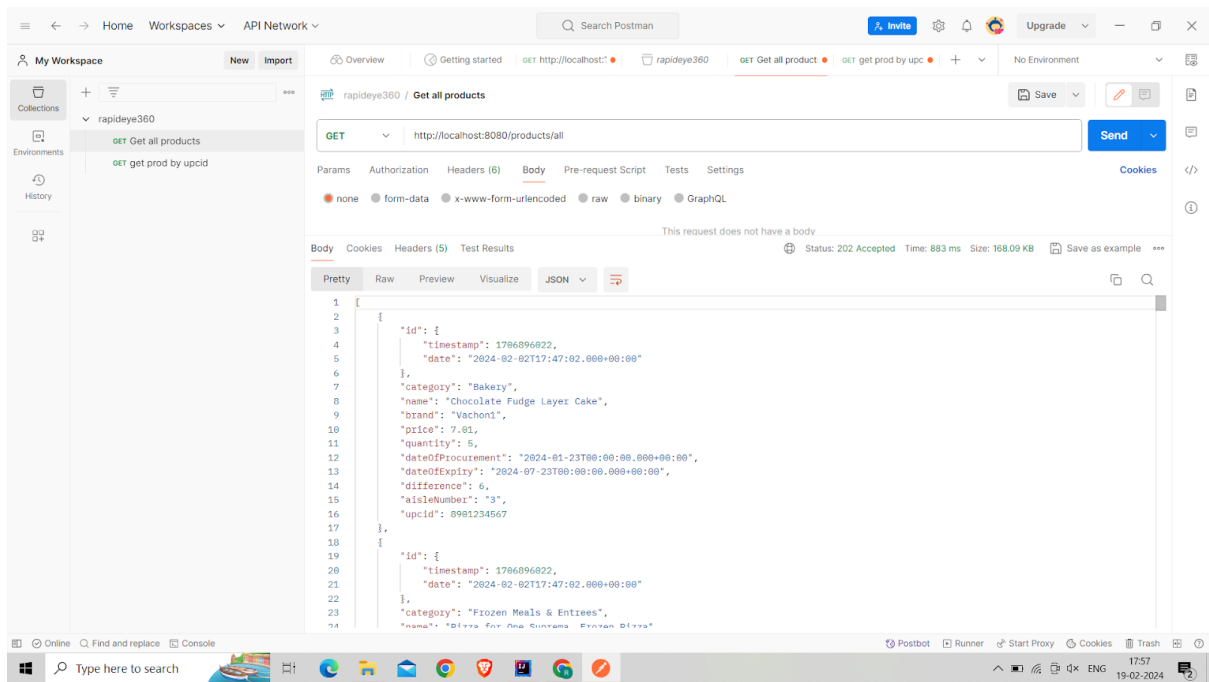
a) addProduct()



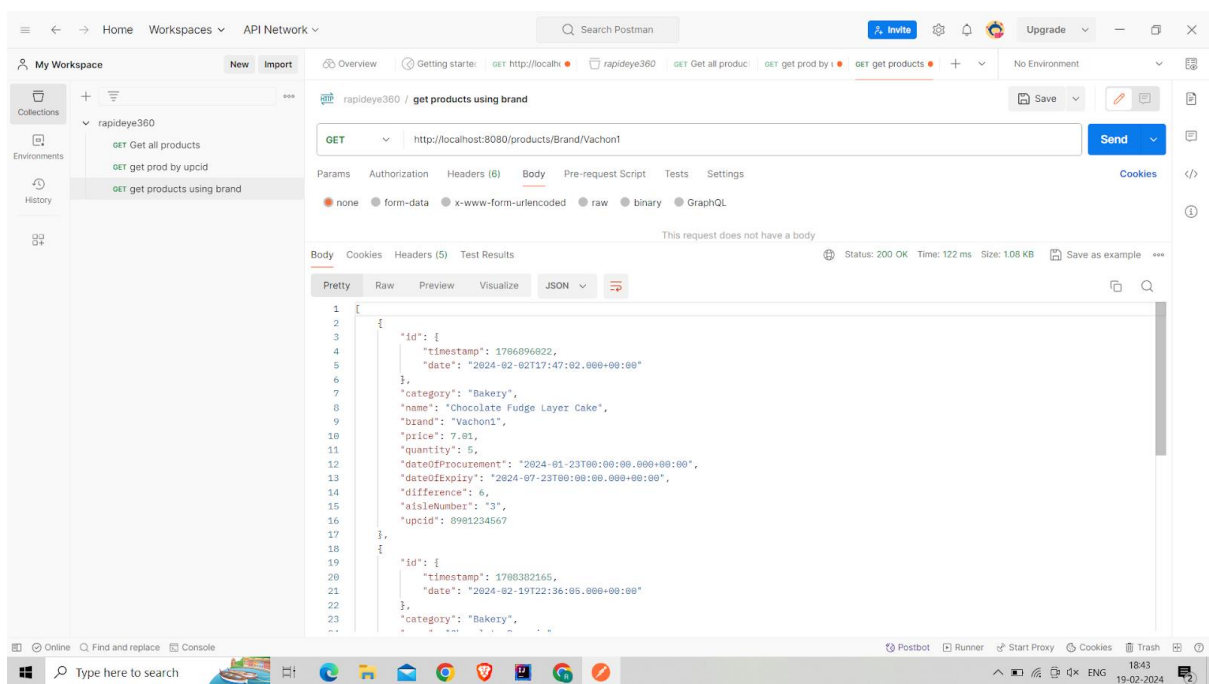
B) getProductByUPCID()



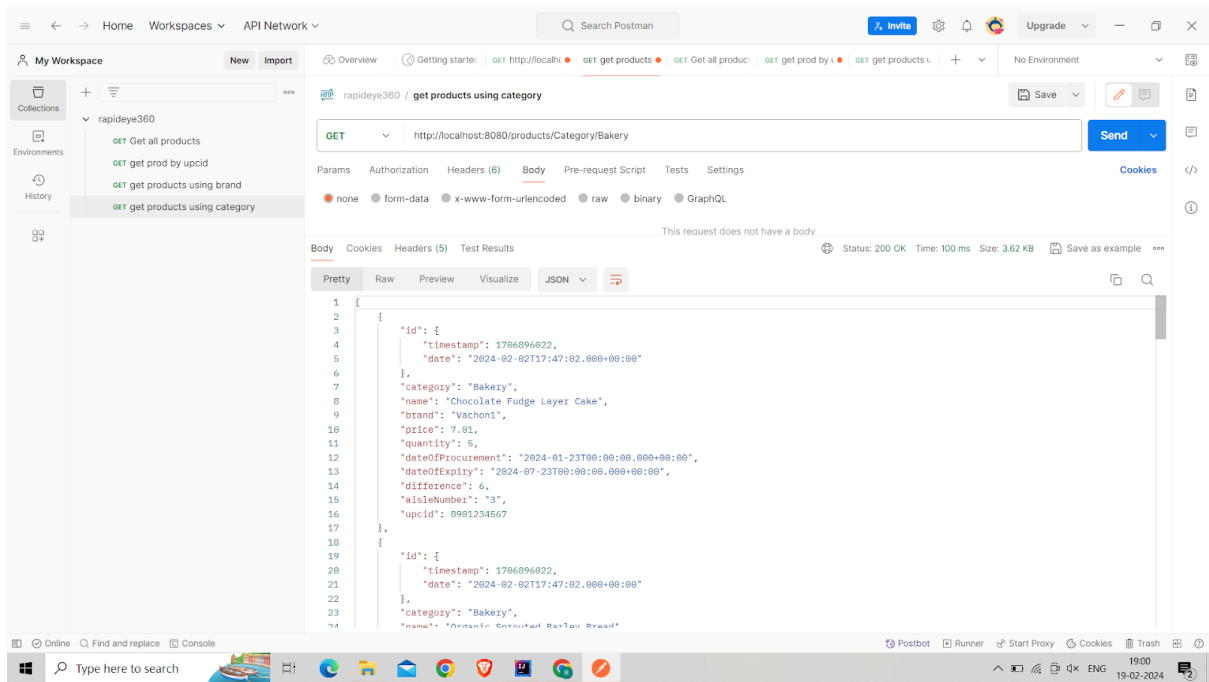
c) getAllProducts()



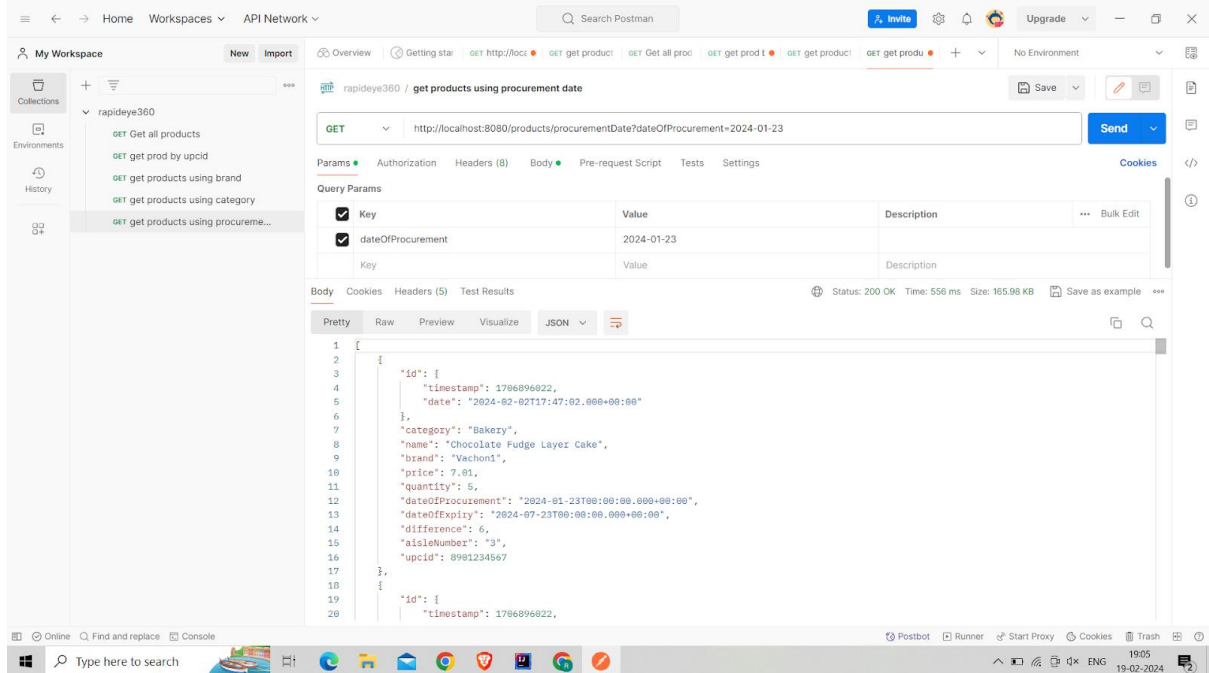
d) getProductsUsingBrandName()



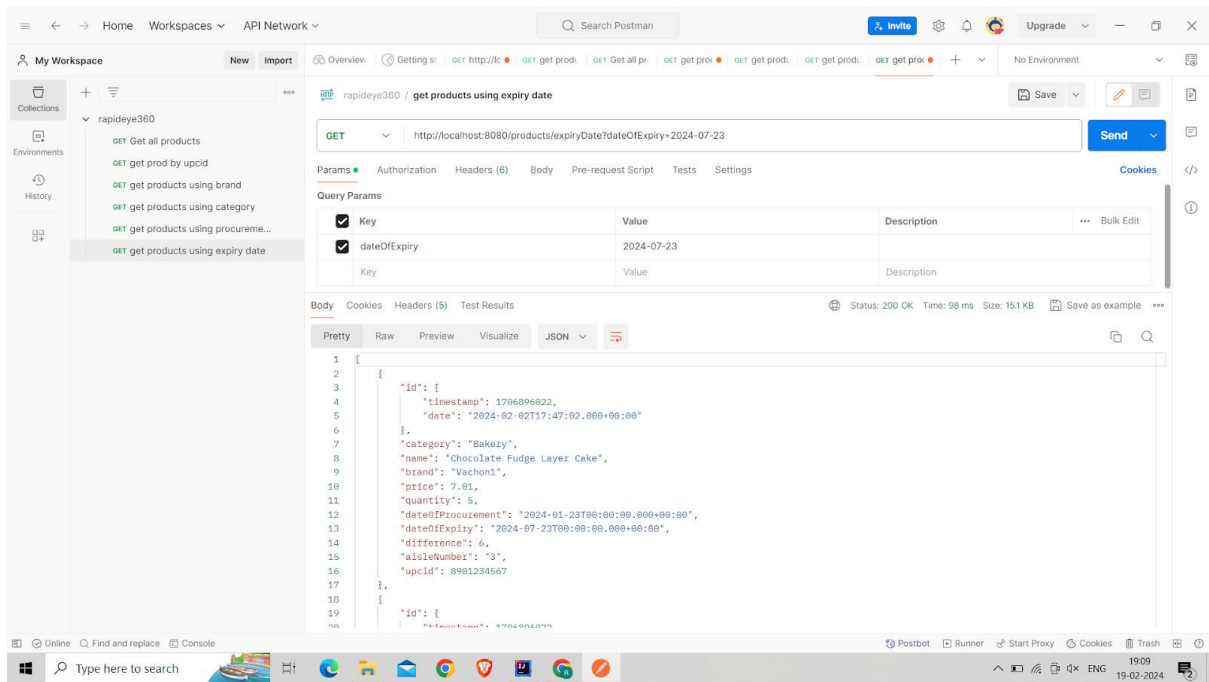
e) getProductsUsingCategory()



f) getProductsUsingProcurementDate()

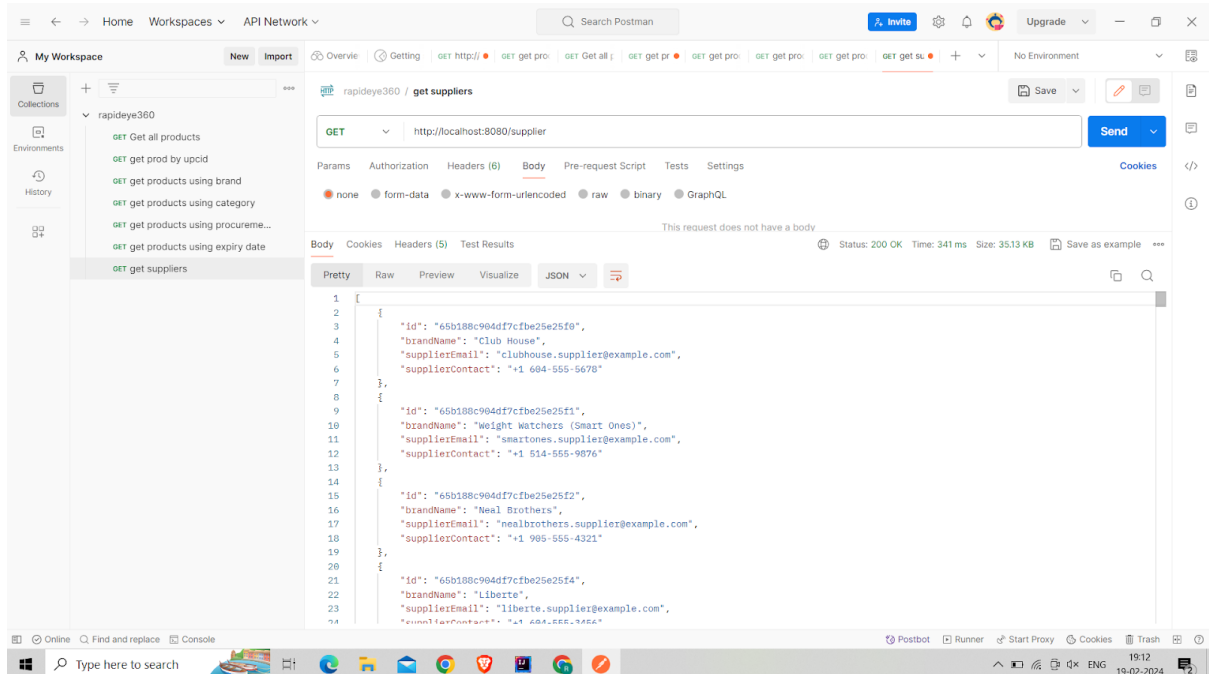


h) getProductsUsingExpiryDate()



Additional Tests:

i) AllSuppliers()



ii) CreateSuppliers()

Postman interface showing a REST client request and response.

Request:

- Method: POST
- URL: http://localhost:8080/supplier
- Body (JSON):

```
1 {
2   "brandName": "Brew Stall",
3   "supplierEmail": "brewstall.supplier@example.com",
4   "supplierContact": "+1 604-555-1222"
5 }
```

Response:

- Status: 200 OK
- Time: 1912 ms
- Size: 309 B
- Body (JSON):

```
1 {
2   "id": "65d3efbce75a86677c550a2a",
3   "brandName": "Brew Stall",
4   "supplierEmail": "brewstall.supplier@example.com",
5   "supplierContact": "+1 604-555-1222"
6 }
```

The interface includes a sidebar with collections and environments, a top search bar, and a bottom status bar with system information.