

# Summary of design considerations

Retry class consists of rest crud methods which implement retry logic.

- retryGet() : get request with retry logic and error handling.
- retryPost() : post request with retry logic and error handling.
- retryDelete(): delete request with retry logic and error handling.

Routes.js module consists of the routes which handles incoming requests..

- List products : /products
- Add a new product : /products with request body.
- Delete a product: /products/:id
- Update an existing product : /product/:id with the request body

## Working screenshots :

- 1) List products : <http://localhost:3000/products>

localhost:3000/products

localhost:3000/products

```
1 // 20220427110906
2 // http://localhost:3000/products
3
4 {
5   "bundle": [
6     {
7       "quantity": 1,
8       "id": "18776b14-cc73-4ea4-81f9-1b4595ee70c5",
9       "price": 123.987,
10      "name": "string"
11    },
12    {
13      "quantity": 2089087,
14      "id": "eaaf44ed-ee50-4808-ab7c-992874783aa7",
15      "price": 10,
16      "name": "Niveditha 123ghf"
17    },
18    {
19      "quantity": 10,
20      "id": "11150cb0-d23b-4d34-8587-4478bb610c8e",
21      "price": 30,
22      "name": "Shampoo Tipo 2"
23    },
24    {
```

```
> nodemon server.js
```

```
[nodemon] 2.0.15
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching path(s): *.*
```

```
[nodemon] watching extensions: js,mjs,json
```

```
[nodemon] starting `node server.js`
```

```
Serve at http://localhost:3000
```

```
Error status:500.Cannot fetch data.Retry attempt 1
```

```
Error status:500.Cannot fetch data.Retry attempt 1
```

2) Add a new product : <http://localhost:3000/products>

The screenshot shows a REST client interface with a POST request to `http://localhost:3000/products`. The 'Body' tab is selected, and the request body is a JSON object: `{ "quantity": 1, "price": 123.987, "name": "Niveditha testing1" }`. The response status is 201 Created, and the response time is 112 ms. The response body is also shown in a 'Pretty' JSON format.

```
POST http://localhost:3000/products
```

Params Auth Headers (8) **Body** Pre-req. Tests Settings

raw JSON

```
1 {
2   ... "quantity": 1,
3   ... "price": 123.987,
4   ... "name": "Niveditha testing1"
5 }
```

Body Cookies Headers (7) Test Results 201 Created 112 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "quantity": 1,
3   "price": 123.987,
4   "name": "Niveditha testing1"
5 }
```

3) Update an existing product: <http://localhost:3000/products/:id>

PATCH

▼

http://localhost:3000/products/eaaf44ed-ee50-4808-ab7c-992874783aa7

Params

Auth

Headers (8)

Body

●

Pre-req.

Tests

Settings

raw

▼

JSON

▼

1

{

2

"quantity": 123,

3

"price": 100,

4

"name": "Niveditha update testing"

5

}

Body

Cookies

Headers (7)

Test Results

🌐

200 OK

425 ms

306 B

Pretty

Raw

Preview

Visualize

HTML

▼

↺

1

Document with id eaaf44ed-ee50-4808-ab7c-992874783aa7 is updated successfully.

If the product doesn't exist :

PATCH

▼

http://localhost:3000/products/eaaf44ed-ee50-4808-ab7c-9928

ParamsAuthHeaders (8)Body●Pre-req.TestsSettings

raw▼JSON▼

1

{

2

...."quantity":123,

3

...."price":100,

4

...."name": "Niveditha update testing"

5

}

BodyCookiesHeaders (7)Test Results

404 Not Found

PrettyRawPreviewVisualizeJSON▼

1

{

2

"message": "Request failed with status code 404"

3

}

4) Delete a product : <http://localhost:3000/:id>

DELETE ▼ http://localhost:3000/products/eaaf44ed-ee50-4808-ab7c-992874783aa7

Params Auth Headers (6) Body Pre-req. Tests Settings

raw ▼ JSON ▼

1

Body Cookies Headers (7) Test Results 🌐 200 OK 321 ms 305 B S

Pretty Raw Preview Visualize HTML ▼ 🔄

1 Document with id eaaf44ed-ee50-4808-ab7c-992874783aa7 is deleted successfully.

If the product id doesn't exist:

DELETE ▼ http://localhost:3000/products/eaaf44ed-ee50-4808-ab7c-992874783aa7

Params Auth Headers (6) Body Pre-req. Tests Settings

raw ▼ JSON ▼

1

Body Cookies Headers (7) Test Results 🌐 404 Not Found 344 ms S

Pretty Raw Preview Visualize JSON ▼ 🔄

```
1 {
2   "message": "Request failed with status code 404"
3 }
```

## Proposed next steps/improvements:

Currently the implemented post rest api does not guarantee the persistence of the data as the data is stored in memory.