



SOT

SuperMarket

RestAPI

Kheir Almosally:3039285



## Contents

Intro .....	3
<b>GET</b> getAll.....	4
<b>GET SEARCHPRODUCT</b> .....	6
<b>GET</b> getProduct by id .....	7
<b>POST</b> addProduct .....	8
<b>DEL</b> deleteProduct.....	9
<b>PUT</b> Buy & Sell Products .....	10
<b>Client Java interface</b> .....	12
<b>gui_client</b> .....	12

## Intro

this document is about Rest API assignment, contain the following

1. Service implements all CRUD operations (path, query and form parameter)
2. Service uses custom-made class (Product) as a parameter.
3. Service uses a collection (List<Product>) as a parameter and returns the value.
4. Client "1" in Java CLI that calls all methods of the service. has a welcoming message and menu include:
  - a. Show all products, Show product with specific id,
  - b. Search in list products by (ID, product name or product type)
  - c. Delete a product by ID.
  - d. Create a new product in the supermarket.
  - e. Sell a product.
  - f. Return a product to the supermarket
  - g. Exit program.
5. Create custom response with error messages.
6. Client "2" web interface use VueJS framework has,
  - a. search for a product.
  - b. Delete products.
  - c. Buy/sell a product.
7. All submitted projects/modules are Gradle projects.
8. The service is hosted on Apache Tomcat.
9. Shortly explaining HTTP messages exchanged between the service and the client.

# GET getAll

http://localhost:9090/rest/products/all

## Example Request

```
getAllProducts
var http = require('http');

var options = {
  'method': 'GET',
  'hostname': 'localhost',
  'path': '/rest/products/all',
  'headers': {
  }
};

var request = http.request(options, function (res) {
  var arr = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });

  res.on("end", function (chunk) {
    var body = Buffer.concat(arr);
    console.log(body.toString());
  });

  res.on("error", function (error) {
    console.error(error);
  });
});

request.end();
```

### Example Response

```
[
  - {
    id: 1,
    name: "Pepsi",
    price: 1.2,
    product: "",
    quantity: 1,
    type: "Drinks"
  },
  - {
    id: 2,
    name: "Chitos",
    price: 1.1,
    product: "",
    quantity: 20,
    type: "Chips"
  },
  - {
    id: 3,
    name: "Beans",
    price: 2,
    product: "",
    quantity: 15,
    type: "Legumes"
  },
  - {
    id: 4,
    name: "Rice",
    price: 1,
    product: "",
    quantity: 13,
    type: "Food"
  }
]
```

# GET SEARCHPRODUCT

http://localhost:9090/rest/products/search?name=Pepsi

## PARAMS

**name:** pepsi

## Example Request

```
getProduct
var http = require('http');

var options = {
  'method': 'GET',
  'hostname': 'localhost',
  'path': '/rest/products/search?name=Pepsi',
  'headers': {
  }
};

var request = http.request(options, function (res) {
  var arr = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });

  res.on("end", function (chunk) {
    var body = Buffer.concat(arr);
    console.log(body.toString());
  });

  res.on("error", function (error) {
    console.error(error);
  });
});

request.end();
```

## Example Response

```
[
  - {
    id: 1,
    name: "Pepsi",
    price: 1.2,
    product: "",
    quantity: 1,
    type: "Drinks"
  }
]
```

# GET getProduct by id

http://localhost:9090/rest/products?id=1

## PARAMS

**Id:** 1

### Example Request

getProduct

```
var http = require('http');
```

```
var options = {  
  'method': 'GET',  
  'hostname': 'localhost',  
  'path': '/rest/product?id=1',  
  'headers': {  
  }  
};
```

```
var request = http.request(options, function (res) {  
  var arr = [];
```

```
  res.on("data", function (chunk) {  
    chunks.push(chunk);  
  });
```

```
  res.on("end", function (chunk) {  
    var body = Buffer.concat(arr);  
    console.log(body.toString());  
  });
```

```
  res.on("error", function (error) {  
    console.error(error);  
  });  
});
```

```
request.end();
```

### Example Response

```
{  
  id: 1,  
  name: "Pepsi",  
  price: 1.2,  
  product: "",  
  quantity: 1,  
  type: "Drinks"  
}
```

# POST addProduct

http://localhost:9090/rest/products/

Headers

## Content-Type

application/x-www-form-urlencoded

BODY urlencoded

**id**

5

**name**

coca

**type**

Drink

**Price**

1.9

**Quantity**

22

Example Request

addProduct

```
var http = require('http');
```

```
var qs = require('querystring');
```

```
var options = {
  'method': 'POST',
  'hostname': 'localhost',
  'path': '/rest/products',
  'headers': {
    'Content-Type': 'application/x-www-form-urlencoded'
  }
};
```

```
var request = http.request(options, function (res) {
  var chunks = [];
```

```
  res.on("data", function (chunk) {
    chunks.push(chunk);
  });
```

```
  res.on("end", function (chunk) {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });
```

```
  res.on("error", function (error) {
    console.error(error);
```



```

    });
  });

  var data_Post = qs.stringify({
    'id': 5,
    'name': 'coca',
    'type': 'Drink',
    'price': 1.9,
    'quantity': 22
  });

  req.write(data_Post);

  request.end();

```

#### Example Response: -

```

Your input:
{
  ID:
  5
  Product:
  coca
  Type
  Drink
  Price
  1.9
  Quantity
  22
}
Created a product. InboundJaxrsResponse{context=ClientResponse{method=POST, uri=http://localhost:9090/rest/products, status=204, reason=No Content}}
Product; coca With ID 5 Quantity 22

```

## DEL deleteProduct

http://localhost:9090/rest/products/delete/2

#### Example Request

```

deleteProduct
var http = require('http');

var options = {
  'method': 'DELETE',
  'hostname': 'localhost',
  'path': '/rest/products/delete/2',
  'headers': {
  }
};

```

#### Example Response:

```

1: List of products (default)
2: Find a Product by id
3: Find Products
4: Add Product
5: Delete Product by id
6: Sell a Product by id
7: Return a Product by id 'Add quantity'
0: EXIT PROGRAM
Your input:
{
  Product id to be deleted:
  5
}
Successfully deleted product with id 5

```

# PUT Buy & Sell Products

http://localhost:9090/rest/products/

## HEADERS

### Content-Type

application/json

### Example Request

```
sellProduct
var http = require('http');

var options = {
  'method': 'PUT',
  'hostname': 'localhost',
  'path': '/rest/products/',
  'headers': {
    'Content-Type': 'application/json'
  }
};

var request = http.request(options, function (res) {
  var chunks = [];

  res.on("data", function (chunk) {
    chunks.push(chunk);
  });

  res.on("end", function (chunk) {
    var body = Buffer.concat(chunks);
    console.log(body.toString());
  });

  res.on("error", function (error) {
    console.error(error);
  });
});

var data_Post = "{\n\t\t\"id\": 2,\n\t\t\"name\": \"Chitos\",\n\t\t\"type\": \"Chips\",\n\t\t}";

req.write(data_Post);

request.end();
```

### Example Response: -

```
6: Sell a Product by id
7: Return a Product by id 'Add quantity'
0: EXIT PROGRAM
Your input:
Product ID:
ID: 2   Product : Chitos   Type: Chips Price: 1.1   Quantity: 20
number of sell
number of sell: 5   Of product Product { ID: 2, Name: Chitos, Type: Chips, Price: 1.1, Quantity: 15, product: }
```

# Client Java interface

```
> Task :ProductAdminClient.main()
WELCOME!

PLEASE CHOOSE AN ACTION:
=====
1: List of products (default)
2: Find a Product by id
3: Find Products
4: Add Product
5: Delete Product by id
6: Sell a Product by id
7: Return a Product by id 'Add quantity'
0: EXIT PROGRAM
Your input:
```

Another Client with web Interface using VUE framework

## gui\_client

### To start

- Project setup  
npm install
- Compiles and hot-reloads for development  
npm run serve
- Compiles and minifies for production  
npm run build
- Lints and fixes files  
npm run lint
- Deactivate browser security

GUI

localhost:8080

# SuperMarket

Search for your product:

id

product name

product type

Submit

add new product(s) to the shop:

product ID	product Name	product type	product price	product quantity

Submit

0

