Shopping cart

Data base initial setup

The database should be created in MySQL with the name "soppingcartdb".

```sql
CREATE SCHEMA `soppingcartdb` ;
```

The tables are created automatically but you must click on MySQL, on update schemas after running the project, for it to be reflected.

The script to view the table where you store all products:

```sql
use shoppingcartdb;
select * from product
```

The script to view the table where you store all carts:

```sql
use shoppingcartdb;
select * from cart
```

The script to view the table where you store all products in all carts:

```sql
use shoppingcartdb;
select * from middle
```

Roles

- **ADMIN**

  Username: admin

  Password: admin

- **USER**

  Username: user

  Password: user

Endpoints

- **Create a new product**

  Endpoint: /api/v1/product

  Body required

  Method: POST

  Access: ADMIN

- **Get all products**

  Endpoint: /api/v1/product

Method: GET

Access: ADMIN, USER

- **Get one product by ID**

  Endpoint: /api/v1/product/{id}

  Method: GET

  Access: ADMIN, USER

- **Update a product by ID**

  Endpoint: /api/v1/product/{id}

  Body required

  Method: PUT

  Access: ADMIN

- **Delete all products**

  Endpoint: /api/v1/product

  Method: DELETE

  Access: ADMIN

- **Delete a product by ID:**

  Endpoint: /api/v1/product/{id}

  Method: DELETE

  Access: ADMIN

- **Create a new cart**

  Endpoint: /api/v1/cart

  Body required

  Method: POST

  Access: ADMIN, USER

- **Get all carts**

  Endpoint: /api/v1/cart

  Method: GET

  Access: ADMIN

- **Get a cart by ID**

  Endpoint: /api/v1/cart/{id}

  Method: GET

Access: ADMIN, USER

- **Update a cart by ID**

  Endpoint: /api/v1/cart/{id}

  Body required

  Method: PUT

  Access: ADMIN, USER

- **Delete all carts**

  Endpoint: /api/v1/cart

  Method: DELETE

  Access: ADMIN

- **Delete a cart by ID**

  Endpoint: /api/v1/cart/{id}

  Method: DELETE

  Access: ADMIN, USER

▲ **Put a product in a cart**

Endpoint: /api/v1/relation

Body required

Method: POST

Access: ADMIN, USER

- **Get all relation product - cart**

  Endpoint: /api/v1/relation

  Method: GET

  Access: ADMIN

▲ **Get all products in a cart**

Endpoint: /api/v1/relation/cart/{id}

ID is from cart

Method: GET

Access: ADMIN, USER

- **Get all carts with a specific product**

  Endpoint: /api/v1/relation/product/{id}

  ID is from product

Method: GET

Access: ADMIN

▲ **Get a product in a cart**

Endpoint: /api/v1/relation/{idCart}/{idProduct}

Method: GET

Access: ADMIN, USER

■ **Update a product in a cart**

Endpoint: /api/v1/relation/{idCart}/{idProduct}

Body required

Method: PUT

Access: ADMIN, USER

■ **Delete all products in all carts**

Endpoint: /api/v1/relation

Method: DELETE

Access: ADMIN

▲ **Delete all products in a cart**

Endpoint: /api/v1/relation/cart/{id}

Method: DELETE

Access: ADMIN, USER

■ **Delete a product in all carts:**

Endpoint: /api/v1/relation/product/{id}

Method: DELETE

Access: ADMIN

▲ **Delete a product in a cart**

Endpoint: /api/v1/relation/{idCart}/{idProduct}

Method: DELETE

Access: ADMIN, USER

Collections to perform CRUD operations

■

- **To create a new product**
- **To update a product by ID**

```json
{
    "name": "producto1",
    "description": "descripcion1",
    "price": 100
}
```

- **To create a new cart**
- **To update a cart by ID**

```json
{
    "user": "user1"
}
```

- **To put a product in a cart**
- **To update a product in a cart (the quantity)**

```json
{
    "quantity": 2,
    "cart": {
        "id": 1
    },
    "product": {
        "id": 1
    }
}
```

Basic rules

1. When you’re updating a product in a cart, you just can change the quantity of the product.
2. A cart can only be deleted if it has no products.
3. A product can only be deleted if it is not in any cart.
4. To add a product in a cart, the product and the carte should already exist