Web Programming I - 5A

August - December Semester
2023

The following document will describe all the important points that the REST API that will be made must contain.

The development of the API will be carried out in a team of 2 people, who will work as a team sharing knowledge, assigning activities, validating the code of their teammates.

Tools to use:

- Typescript
- Github (public repository)
- MYSQL
- TypeORM
- REST architecture
- MVC design pattern
- Server to deploy
- Postman (For endpoint testing and documentation)
- OOP (Classes, abstraction, interfaces, polymorphism, inheritance, encapsulation must be used)

Minimum actions that the API must perform:

1. Endpoint login which will serve for token-based authentication (JWT)
2. Crud for users (create, read, update, delete)
3. Crud of products (depending on the topic assigned)
4. Crud of promotional products
5. Purchase module
6. Statistics module (Only applies to a team of 3 members, having at least two endpoints)

Login

To access our API and obtain information, it is necessary for the user to complete an authentication process through the 'login' endpoint. At this point, the user must provide the appropriate credentials, which will be verified by our system to determine if they are eligible to receive an access token. This access token will be used in subsequent requests to obtain and send information.

User CRUD

Users (GET)
This endpoint allows you to obtain a list of all users registered in the system.
Url:mydomain.com/team-N/users?queryparameters

User By Id (GET)
Allows you to obtain detailed information about a specific user identified by their ID.
mydomain.com/team-N/user/{id}

Users (PUT):
This endpoint allows updating information for a specific user using their ID.

Url:mydomain.com/team-N/user/{id}

Users(DELETE):
Allows the deletion of a specific user identified by their ID. *Url: mydomain.com/team-N/user/{id}*

Users (POST):
This endpoint allows the registration of a new user in the system. The record must include the following fields:

- id
- name
- email
- password
- creation date
- creation user
- update date
- update user
- active (logical erase)
- (Add 2 additional fields)

Url:mydomain.com/team-N/user

Crud products

products (GET)
With this method we can obtain a list of all the products that we have stored in our database. But we must take into account that we can make certain filters through the queryparameters indicated in the url

Url: mydomain.com/team-N/products?queryparameters

product By Id (GET)
We can obtain all the information about a specific product through its ID *Url: mydomain.com/team-N/products/{id}*

product (PUT)
With this action, our system will update any product that we indicate by means of the ID.
Url: mydomain.com/team-N/products/{id}

product (DELETE)
Deletion of a product through its respective ID *Url: mydomain.com/team-N/products/{id}*

product (POST)
Creation of a new product through our system. At least a product must have:

- id
- name
- description
- price
- category
- manufacturer
- quantity in stock
- unit of measurement (pc, liters, kg, etc.)
- creation date
- creation user
- update date
- update user
- active (logical erase)
- Add two additional fields

Url: mydomain.com/team-N/product

CRUD Promotional Products

promotionalproduct (GET)

Get a list of all the products that are on promotion in the system. *Url: mydomain.com/team-N/promotionalproducts?queryparameters*

promotionalproduct By Id (GET)
Get detailed information of a specific promotional product by its ID. *Url: mydomain.com/team-N/promotionalproducts/{id}*

promotional product (PUT)
Updates the information for a specific promotional product identified by its ID. Url: mydomain.com/team-N/*promotionalproducts*/{id}

promotionalproduct (DELETE)
Delete a specific promotional product by its ID. *Url: mydomain.com/team-N/promotionalproducts/{id}*

promotionalproduct (POST)
Register a new promotional product in the system. The record must include at least the following fields:

- id
- name
- description
- promotional price
- promotion start date
- promotion end date
- active (logical erase)
- (Add 2 additional fields)

Url: mydomain.com/team-N/promotionalproducts

Shopping module

purchases (GET)
Get the entire list of completed purchases *Url: mydomain.com/equipo-N/purchases*

purchase By Id (GET)
Obtain all the information about a purchase through its ID (take into account that the purchase has a purchase detail)
Url: mydomain.com/team-N/purchases/{id)

purchase (POST)
Creation of a completed purchase *Url: mydomain.com/equipo-N/purchases*

- id
- description
- customer name
- total price
- total products
- creation date
- creation user
- update date
- update user
- active (logical erase)
- Purchase details
    - id
    - product
    - order
    - creation user
    - update date

NOTE:

- This project will be delivered in a complete and 100% functional manner to obtain your unit 2 and ordinary exam right.
- The teacher may request progress at any time from any team and ask questions to the team members.
- If the student does not make considerable progress over time or is unable to answer the questions asked by the teacher, the student will lose the right to take the unit 2 and ordinary exam. Automatically failing the semester in the web programming I subject
- OOP programming will be very important, failing to comply with this point could be a risk of the API not being accepted as it is not scalable and maintainable