

Documentatie OnlineShop

Utilizator API

Un utilizator se poate crea cu ajutorul metodei POST createUser.

<http://localhost:8080/users/createUser?email=ion@gmail.com&surname=ion&name=marcel&usertype=client>

Aceasta intoarce noul user creat.

```
1  {
2    "id": 17,
3    "email": "ion@gmail.com",
4    "surname": "ion",
5    "name": "marcel",
6    "usertype": "client"
7  }
```

Userii pot fi de două tipuri: administrator sau client. În cazul în care se introduce un utilizator de un alt tip, va fi aruncată o excepție.

De asemenea, se pot afisa toți utilizatorii cu ajutorul metodei GET getAll.

<http://localhost:8080/users/allusers>

Lot API

Fiecare produs ce este introdus aparține unui lot. Cu ajutorului acestuia, se menține evidenta stocului unui produs. Se formează o relație One-To-One.

Disponibilitate unui produs se poate verifica cu ajutorul metodei GET GetNumber. Aceasta returnează numărul de produse disponibile de un anumit tip. Vă fi nevoie de un singur parametru, numele produsului.

<http://localhost:8080/loturi/thisproducts?name=tablou cu rocky>

De asemenea, se poate introduce un nou lot cu ajutorul metodei POST create.

<http://localhost:8080/loturi>

Categorie

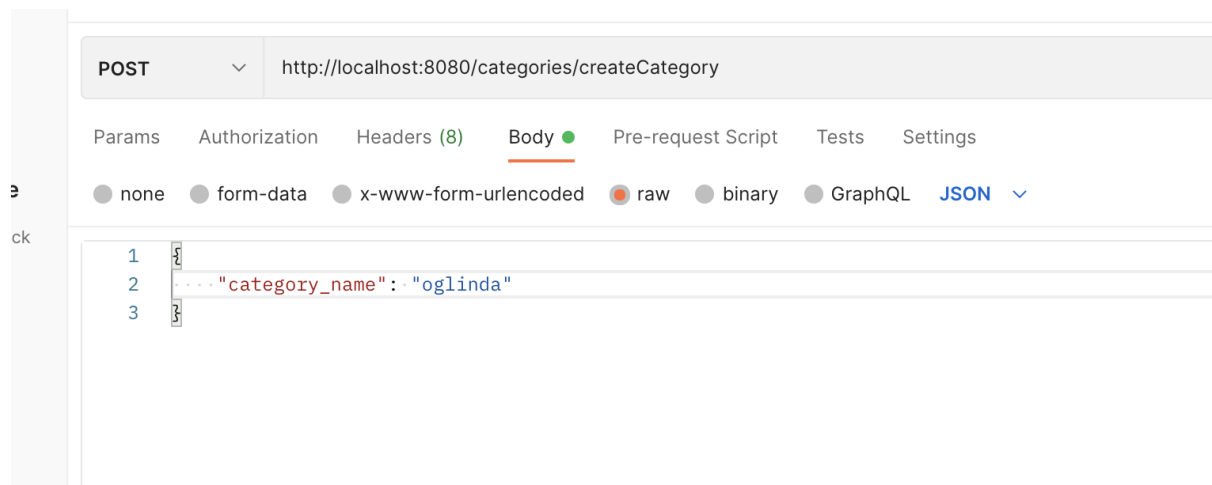
Fiecare produs are asignată o categorie (One to One).

Se pot afișa toate categoriile cu ajutorul metodei GET GetAll.

<http://localhost:8080/categories/list>

De asemenea, se creează noi categorii cu metoda POST Create și se afișează toate produsele dintr-o anumită categorie.

<http://localhost:8080/categories/createCategory>



`http://localhost:8080/categories/categories?category=tablou`

Produs API

Se pot crea produse noi cu ajutorul metodei create.

De asemenea, se pot șterge produse cu metoda POST delete.

`http://localhost:8080/products/delete?product_id=1`

Se vor afișa produsele cu ajutorul GET getALL.

`http://localhost:8080/products/list`

Cart API & CartItem API

Fiecare utilizator are un coș de cumpărături (One-To-one). În acesta se afla mai multe produse(One-To-Many). Produsele ce sunt introduse în coșul de cumpărături sunt de tipul CartItem. Un CartItem este reprezentat de către un produs, deci este o relație de tipul One-To-One.

Un produs este adăugat în coșul unui client cu ajutorul metodei POST addToCart. Deci este nevoie 2 parametrii: id-ului produsului ce dorește a fi adăugat și email-ul clientului. Dacă clientul nu are încă coș de cumpărături creat, acesta se va crea automat. În tabelul cartItems se menține id-ul utilizatorului și al produsului.

http://localhost:8080/cartItems/addToCart?product_id=1&email=malina@gmail.com

Comanda

În momentul plasării unei comenzi, produsele din coș vor dispărea, se va calcula prețul total și insera o nouă comandă. Pentru a plasa comanda, este nevoie de email-ul utilizatorului, id-ul acestuia urmând a fi căutat.

Rezultatul returnat este reprezentat de suma totală.

<http://localhost:8080/comenzi/placeComandaFor?email=malina@gmail.com>

The screenshot displays a REST client interface with the 'Params' tab selected. The 'Query Params' section shows a table with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email	malina@gmail.com	
Key	Value	Description

Below the params section, the 'Body' tab is active, showing the response of the API call. The response is a JSON object with the following structure:

```
{  "1": "Comanda plasata in valoare de 2000"}
```

The status bar at the bottom indicates a 200 OK response with a response time of 295 ms and a body size of 198 B.

Adresa

Fiecărui User îi este asignată o adresă. Astfel, apare o relație de tipul One-To-One. Adresa asignata unui uşer se poate verificata cu ajutorul metodei de tipul GET getAll. Se afişează, astfel, un String ce reprezintă câmpul adresă.

<http://localhost:8080/getadresa/1>