Test dla list 4

Michał Waluś

Testy robione za pomocą programu postman. Authorization - odnosi się do nagłówka o tej samej nazwie, jeśli wpisane użytkownik/administrator to znajduje się tam "Bearer (token)", gdzie token jest tokenem JWT dla odpowiedniego typu konta.

Wygaśnięcie tokenu (po 1h):

```
{
    "message": "Token is invalid!"
}
```

Użytkownicy

Login (konto nie istnieje)

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/login
Body:

```
{
    "username":"test",
    "password":"test"
}
```

Output:

```
{
    "message": "Invalid email or password"
}
```

Code: 400

Rejestracja

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body:

```
{
    "username":"test",
    "password":"test"
}
```

Output:

```
{
    "message": "User registered successfully."
}
```

Code: 201

Rejestracja (Użytkownik istnieje)

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body:

```
{
    "username":"test",
    "password":"test"
}
```

Output:

```
{
    "message": "User already exists. Please login."
}
```

Code: 400

Rejestracja (Błędne dane)

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body (4 zapytania):

```
{
    "username":"",
    "password":"test"
}
{
    "username":"a",
    "password":"test"
}
{
    "username":"ab",
    "password":"test"
}
{
    "username":"abc",
    "password":"test"
}
```

Output:

```
{
    "message": "Username must be at least 4 characters long."
}
```

Code: 400

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body (4 zapytania):

```
{
    "username":"test",
    "password":""
}
{
    "username":"test",
    "password":"a"
}
{
    "username":"test",
    "password":"ab"
}
{
    "username":"test",
    "password":"abc"
}
```

Output:

```
{
    "message": "Password must be at least 4 characters long."
}
```

Code: 400

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body (4 zapytania):

```
{
    "username":"1234567890_1234567890_1234567890_1234567890_1234567890_1234567890",
    "password":"test"
}
```

Output:

```
{
    "message": "Username might be at most 64 characters long."
}
```

Code: 400

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/register
Body (4 zapytania):

```
{
    "username":"test",
    "password":"1234567890_1234567890_1234567890_1234567890_1234567890_1234567890"
}
```

Output:

```
{
    "message": "Password might be at most 64 characters long."
}
```

Code: 400

Logowanie

Authorization: brak (nie wpływa)
Typ: POST
Link: http://localhost:5000/login
Body:

```
{
    "username":"test",
    "password":"test"
}
```

Output:

```
{
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiZXhwIjoxNzQ3NjE3NTY2fQ.Ac
dVW_P4eJ4faBYhU_HUS3Sl_gk8qj9hae2BoPkYg6g"
}
```

Code: 200

Authorization: brak (nie wpływa)
Link: http://localhost:5000/login
Body:

```
{
    "username":"Admin",
    "password":"adminadmin"
}
```

Output:

```
{
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiZXhwIjoxNzQ3NjE3NjM2fQ.dS
hkuRXfQEJrjY-x4xpoTZ80iY2a6rSHJ1Z2wdpmcBQ"
}
```

Code: 200

GET (list użytkowników)

Authorization: Użytkownik
Typ: GET
Link: http://localhost:5000/users
Output:

```
{
    "message": "Access denied."
}
```

Code: 403

Authorization: Brak
Typ: GET
Link: http://localhost:5000/users
Output:

```
{
    "message": "No authentication token present"
}
```

Code: 401

Authorization: Administrator
Typ: GET
Link: http://localhost:5000/users
Output:

```
{
    "users": [
        {
            "id": 1,
            "type": "admin",
            "username": "Admin"
        },
        {
            "id": 3,
            "type": "user",
            "username": "test"
        }
    ]
}
```

Code: 200

POST (nowy użytkownik)

Authorization: Brak
Typ: POST
Link: http://localhost:5000/users
Body:

```
{
    "username":"JanKowalski",
    "password":"1234",
    "type":"user"
}
```

Output:

```
{
    "message": "No authentication token present"
}
```

Code: 401

Authorization: Użytkownik
Typ: POST
Link: http://localhost:5000/users
Body:

```
{
    "username":"JanKowalski",
    "password":"1234",
    "type":"user"
}
```

Output:

```
{
    "message": "Access denied."
}
```

Code: 403

Authorization: Administrator
Typ: POST

Link: http://localhost:5000/users
Body:

```
{
    "username":"JanKowalski",
    "password":"1234",
    "type":"user"
}
```

Output:

```
{
    "message": "User created successfully."
}
```

Code: 201

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/users
Body:

```
{
    "username":"JanuszKowalski",
    "password":"12345",
    "type":"admin"
}
```

Output:

```
{
    "message": "User created successfully."
}
```

Code: 201

POST (niewłaściwy typ)

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/users
Body:

```
{
    "username":"JanekKowalski",
    "password":"12345",
    "type":""
}
```

Output:

```
{
    "message": "Invalid user type."
}
```

Code: 200

POST (próba stworzenia istniejącego użytkownika)

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/users
Body:

```
{
    "username":"JanKowalski",
    "password":"1234",
    "type":"user"
}
```

Output:

```
{
    "message": "User already exists."
}
```

Code: 400

DELETE (Usuwanie konta)

Authorization: Użytkownik (zalogowany jako JanKowalski)
Typ: DELETE
Link: http://localhost:5000/users
Body: brak Output:

```
```

```
{
    "message": "Deleted account"
}
```

Code: 200

Authorization: Brak
Typ: DELETE
Link: http://localhost:5000/users
Body: brak Output:

```
{
    "message": "No authentication token present"
}
```

Code: 401

Lista użytkowników w tym momencie:

```
{
    "users": [
        {
            "id": 1,
            "type": "admin",
            "username": "Admin"
        },
        {
            "id": 3,
            "type": "user",
            "username": "test"
        },
        {
            "id": 6,
            "type": "admin",
            "username": "JanuszKowalski"
        }
    ]
}
```

DELETE (Usuwanie innych użytkowników)

Authorization: Użytkownik
Typ: DELETE
Link: http://localhost:5000/users/6
Body: brak Output:

```
{
    "message": "Access denied."
}
```

Code: 403

Authorization: Administrator
Typ: DELETE
Link: http://localhost:5000/users/6
Body: brak Output:

```
{
    "message": "User deleted successfully."
}
```

Code: 200

Produkty

GET (Lista produktów)

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products
Body: brak Output:

```
{
    "products": [
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "Perfect for bricks",
            "id": 8,
            "name": "Mortar",
            "price": 40.0
        }
    ]
}
```

Code: 200

POST (Dodawanie)

Authorization: Brak
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Plank",
    "description": "Oak wood",
    "price": 350
}
```

Output:

```
{
    "message": "No authentication token present"
}
```

Code: 401

Authorization: Użytkownik
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Plank",
    "description": "Oak wood",
    "price": 350
}
```

Output:

```
{
    "message": "Access denied."
}
```

Code: 403

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Plank",
    "description": "Oak wood",
    "price": 350
}
```

Output:

```
{
    "message": "Added product."
}
```

Code: 201

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Plank",
    "description": "Birch wood",
    "price": 350
}
```

Output:

```
{
    "message": "Added product."
}
```

Code: 201

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Plank",
    "description": "Birch wood",
    "price": 350
}
```

Output:

```
{
    "message": "Product already exists."
}
```

Code: 400

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Door",
    "description": "Made from wood",
    "price": -2
}
```

Output:

```
{
    "message": "Price must be positive."
}
```

Code: 400

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products
Body:

```
{
    "name": "Door",
    "description": "Made from wood",
    "price": 100000
}
```

```
```

Output:

```
{
    "message": "Added product."
}
```

Code: 201

GET (z parametrami)

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products
Body: brak Output:

```
{
    "products": [
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "Perfect for bricks",
            "id": 8,
            "name": "Mortar",
            "price": 40.0
        },
        {
            "description": "Oak wood",
            "id": 9,
            "name": "Plank",
            "price": 3.5
        },
        {
            "description": "Birch wood",
            "id": 10,
            "name": "Plank",
            "price": 3.5
        },
        {
            "description": "Made from wood",
            "id": 11,
            "name": "Door",
            "price": 1000.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?page=1
Body: brak Output:

```
    {
        "products": [
            {
                "description": "A cheap Brick",
                "id": 1,
                "name": "Brick",
                "price": 7.0
            },
            {
                "description": "A cheap Brick too",
                "id": 3,
                "name": "Brick",
                "price": 7.0
            },
            {
                "description": "Perfect for bricks",
                "id": 8,
                "name": "Mortar",
                "price": 40.0
            },
            {
                "description": "Oak wood",
                "id": 9,
                "name": "Plank",
                "price": 3.5
            },
            {
                "description": "Birch wood",
                "id": 10,
                "name": "Plank",
                "price": 3.5
            }
        ]
    }
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?page=2
Body: brak Output:

```
{
    "products": [
        {
            "description": "Made from wood",
            "id": 11,
            "name": "Door",
            "price": 1000.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?page=3
Body: brak Output:

```
{
    "message": "No such page."
}
```

Code: 400

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?more=500
Body: brak Output:

```
{
    "products": [
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "Perfect for bricks",
            "id": 8,
            "name": "Mortar",
            "price": 40.0
        },
        {
            "description": "Made from wood",
            "id": 11,
            "name": "Door",
            "price": 1000.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=500
Body: brak Output:

```
{
    "products": [
        {
            "description": "Oak wood",
            "id": 9,
            "name": "Plank",
            "price": 3.5
        },
        {
            "description": "Birch wood",
            "id": 10,
            "name": "Plank",
            "price": 3.5
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?more=500&less=1000
Body: brak Output:

```
{
    "products": [
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?more=500&sort=price
Body: brak Output:

```
{
    "products": [
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "Perfect for bricks",
            "id": 8,
            "name": "Mortar",
            "price": 40.0
        },
        {
            "description": "Made from wood",
            "id": 11,
            "name": "Door",
            "price": 1000.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?more=500&sort=description&desc=true
Body: brak Output:

```
{
    "products": [
        {
            "description": "Perfect for bricks",
            "id": 8,
            "name": "Mortar",
            "price": 40.0
        },
        {
            "description": "Made from wood",
            "id": 11,
            "name": "Door",
            "price": 1000.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=1000&sort=name&desc=true
Body: brak Output:

```
{
    "products": [
        {
            "description": "Oak wood",
            "id": 9,
            "name": "Plank",
            "price": 3.5
        },
        {
            "description": "Birch wood",
            "id": 10,
            "name": "Plank",
            "price": 3.5
        },
        {
            "description": "A cheap Brick",
            "id": 1,
            "name": "Brick",
            "price": 7.0
        },
        {
            "description": "A cheap Brick too",
            "id": 3,
            "name": "Brick",
            "price": 7.0
        }
    ]
}
```

Code: 200

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=1000&desc=true
Body: brak Output:

```
{
    "message": "What do you wanna sort by?"
}
```

Code: 400

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=100&more=200
Body: brak Output:

```
{
    "message": "More should be smaller than less."
}
```

Code: 400

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=100&more=100e
Body: brak Output:

```
{
    "message": "More should be smaller than less."
}
```

Code: 400

Authorization: Brak (nie ma znaczenia)
Typ: GET
Link: http://localhost:5000/products?less=0
Body: brak Output:

```
{
    "message": "Less has to be a positive number."
}
```

Code: 400

PUT (aktualizowanie produktu)

Przy braku autoryzacji i dla użytkownika tak samo jak przy POST.

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products/10
Body:

```
{
    "name": "Plank",
    "description": "Spruce wood",
    "price": 350
}
```

Output:

```
{
    "message": "Product updated successfully"
}
```

Code: 200

Authorization: Administrator
Typ: POST
Link: http://localhost:5000/products/10
Body:

```
{
    "name": "Plank",
    "description": "Spruce wood",
    "price": 0
}
```

Output:

```
{
    "message": "Price must be positive."
}
```

Code: 400

DELETE (usuwanie produktu)

Przy braku autoryzacji i dla użytkownika tak samo jak przy POST.

Authorization: Administrator
Typ: DELETE
Link: http://localhost:5000/products/10
Body: brak Output:

```
{
    "message": "Product deleted successfully"
}
```

Code: 200

Authorization: Administrator
Typ: DELETE
Link: http://localhost:5000/products/10
Body: brak Output:

```
{
    "message": "Product not found"
}
```

Code: 404

Opinie

GET

Authorization: Brak (bez znaczenia)
Typ: GET
Link: http://localhost:5000/products/1/reviews
Body: brak Output:

```
{
    "reviews": []
}
```

Code: 200

POST

Authorization: Brak
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 8
}
```

Output:

```
{
    "message": "No authentication token present"
}
```

Code: 401

Authorization: Użytkownik/Administrator
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 8
}
```

Output:

```
{
    "message": "Review added successfully"
}
```

Code: 201

Authorization: Użytkownik/Administrator
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 8
}
```

Output:

```
{
    "message": "Review already exists for this product"
}
```

Code: 400

Authorization: Użytkownik/Administrator
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 0
}
```

Output:

```
{
    "message": "Rating has to be a number 1-10"
}
```

Code: 400

Authorization: Użytkownik/Administrator
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": "a"
}
```

Output:

```
{
    "message": "Rating has to be a number 1-10"
}
```

Code: 400

Authorization: Użytkownik/Administrator
Typ: POST
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": ""
}
```

Output:

```
{
    "message": "Rating is required"
}
```

Code: 400

PUT

Authorization: Użytkownik/Administrator
Typ: PUT
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 9
}
```

Output:

```
{
    "message": "Review updated successfully"
}
```

Code: 200

Authorization: Użytkownik/Administrator
Typ: PUT
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": 0
}
```

Output:

```
{
    "message": "Rating has to be a number 1-10"
}
```

Code: 400

Authorization: Użytkownik/Administrator
Typ: PUT
Link: http://localhost:5000/products/1/reviews
Body:

```
{
    "rating": "a"
}
```

Output:

```
{
    "message": "Rating has to be a number 1-10"
}
```

Code: 400

Authorization: Użytkownik/Administrator
Typ: PUT
Link: http://localhost:5000/products/3/reviews
Body:

```
{
    "rating": 2
}
```

Output:

```
{
    "message": "Review does not exist"
}
```

Code: 400

GET (ponownie)

Authorization: Brak (bez znaczenia)
Typ: GET
Link: http://localhost:5000/products/1/reviews
Body: brak Output:

```
{
    "reviews": [
        {
            "rating": 9,
            "username": "test"
        }
    ]
}
```

Code: 200

DELETE

Authorization: Użytkownik/Administrator
Typ: DELETE
Link: http://localhost:5000/products/3/reviews
Body: brak Output:

```
{
    "message": "Review does not exist"
}
```

Code: 404

Authorization: Użytkownik/Administrator
Typ: DELETE
Link: http://localhost:5000/products/1/reviews
Body: brak Output:

```
{
    "message": "Review deleted successfully"
}
```

Code: 200