

BED - AUG22FT - EXAM

This is my exam project for the backend study at Noroff, our project was to create the backend for shopping items from Noroff's provided api.

Authentication endpoints

Endpoints for user login and signup.

POST /signup

http://localhost:3000/signup



Endpoint for user registration. Creates a new user account with the provided information.

Body raw (json)

json

```
{
  "username": "Postman",
  "firstname": "Pat",
  "lastname": "Clifton",
  "email": "postman@pat.com",
  "password": "yougotmail"
}
```

POST /login

http://localhost:3000/login

Endpoint for user login. Requires a valid username and password. Returns a JWT token upon successful login.

Body raw (json)

json

```
{
```

```
"username": "postman",

"password": "yougotmail"
}
```

Items endpoints

Endpoints for managing product items, including adding, updating, and deleting items.

GET Get /Items - Show all items

http://localhost:3000/items

Endpoint to retrieve all items in the database, including their categories.

POST Post /item - Create a new item



http://localhost:3000/item

Endpoint for adding a new item to the database. Requires item information and a valid category ID.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "name": "Johnson 100HP",
  "sku": "Test123",
  "price": 15000,
  "img_url": "https://i.ytimg.com/vi/bNcFnundsiU/hqdefault.jpg",
  "categoryId": 4,
  "stock": 15
}
```

PUT Put /item/:id - Edit an item



http://localhost:3000/item/1

Endpoint for updating an existing item in the database. Requires a valid item ID in the url and updated item information in the body.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "name": "Johnson 100HP",
  "sku": "Test123",
  "price": 12000,
  "img_url": "https://i.ytimg.com/vi/bNcFnundsiU/hqdefault.jpg",
  "categoryId": 4,
  "stock": 9
}
```

DELETE DELETE /item/:id - Delete an item



http://localhost:3000/item/25

Endpoint for deleting an item from the database. Requires a valid item ID in the url.

AUTHORIZATION Bearer Token

Token <token>

Category endpoints

Endpoints for managing product categories, including adding, updating, and deleting categories.

GET /categories - Show all categories

http://localhost:3000/categories

Endpoint to retrieve all categories in the database.

POST /category - Add new category



http://localhost:3000/category

Endpoint for adding a new category to the database.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{  
  "name": "Mail Boxes"  
}
```

PUT /category/:id - Edit category name



http://localhost:3000/category/8

Endpoint for updating the name of an existing category. Requires a valid category ID and updated category name.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{  
  "name": "Boats"  
}
```

DELETE /category/:id - Delete a category



http://localhost:3000/category/9

Endpoint for deleting a category from the database. Requires a valid category ID.

AUTHORIZATION Bearer Token

Token <token>

Cart endpoints

Endpoints for managing user shopping carts, including adding items, updating quantities, and deleting items.

GET /cart - Show a users cart to the user



```
http://localhost:3000/cart
```

Endpoint to retrieve the user's shopping cart. Requires user authentication.

AUTHORIZATION Bearer Token

Token <token>

GET /allcarts - Get all carts that exist with full names



```
http://localhost:3000/allcarts
```

Endpoint to retrieve all carts in the database, including the items and user information associated with each cart. Requires admin privileges.

AUTHORIZATION Bearer Token

Token <token>

POST /cart_item - Add items to cart



```
http://localhost:3000/cart_item
```

Endpoint for adding an item to the user's shopping cart. Requires item ID and desired quantity.

AUTHORIZATION Bearer Token

Token <token>

Back to menu / Home

Body raw (json)

json

```
{
  "itemId": 19,
  "quantity": 2
}
```

PUT /cart_item/:id - Edit quantity of an item in cart



http://localhost:3000/cart_item/19

Endpoint for updating the quantity of an item in the user's cart.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "quantity": 3
}
```

DELETE /cart_item/:id - Delete an item from cart



http://localhost:3000/cart_item/11

Endpoint for deleting an item from the user's cart.

AUTHORIZATION Bearer Token

Token <token>

DELETE /cart/:id - Delete all items from cart



http://localhost:3000/cart/1

Endpoint for clearing all items from the user's cart.

AUTHORIZATION Bearer Token

Token <token>

Orders endpoint

Endpoints for managing user orders, including placing new orders, retrieving order details, and updating order statuses.

GET /orders - Get users order for user or all orders for admin



```
http://localhost:3000/orders
```

Endpoint to retrieve the user's orders. Requires user authentication. This endpoint will also show all orders for an admin user.

AUTHORIZATION Bearer Token

Token <token>

GET /allorders - Admin gets all orders and full names



```
http://localhost:3000/allorders
```

Endpoint to retrieve all orders in the database, including the items and user information associated with each order. Requires admin privileges.

AUTHORIZATION Bearer Token

Token <token>

POST /order/:id - Registered user should be able to order here



```
http://localhost:3000/order/222
```

Endpoint for placing a new order. Requires item ID and quantity. Updates the order, item stock, and calculates the total price.

AUTHORIZATION Bearer Token

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "itemId": 22,
  "quantity": 2
}
```

PUT /order/:id - Update order status



http://localhost:3000/order/1

At this endpoint an admin can update the orders status, admin can only update to either 'In process', 'Complete' or 'Cancelled'.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "status": "Complete"
}
```

Utility endpoints

Endpoints for database setup and search functionality.

POST /setup

http://localhost:3000/setup

Endpoint for initial database population. Populates the items, categories, and roles tables. Creates an admin user.

POST /search

http://localhost:3000/search

Endpoint for searching items or categories in the database.

Body raw (json)

json

```
{  
  "item_name": "Laptop"  
}
```