EP1-StockSales

This API is designed as a backend for a fictive online store. It manages users, roles, items, carts, orders, and categories. It includes endpoints for creating, retrieving, updating, and deleting these elements.

Authentication and User delete

This is the signup and login endpoints for authentication of users and getting a JWT token to further communicate with the other endpoints in this api. Remember to copy the "token" value and put it into the authentication tab / bearer token, to be able to handle user related tasks. Some of the endpoints also requires admin user to access.

It also includes a user delete route(requires admin user).

POST /signup

http://localhost:3000/signup

POST / Signup. Endpoint for registering new users. It takes "fullname", "username", "email", and "password".

Body raw (json)

json

```
{
  "fullname": "Your Full Name",
  "username": "YourUsername",
  "email": "your@email.com",
  "password": "YourPassword"
}
```

POST /login

http://localhost:3000/login

POST /signup. Endpoint for authenticating user with login and returning JWT token for using other endpoints. It takes "username" and "password".

This JWT token needs to be placed under authentication tab/ Bearer Token to make most of the endpoints in this api work.

Body raw (json)

```
json

{
  "username": "YourUsername",
  "password": "YourPassword"
}
```

DELETE /delete User

http://localhost:3000/users/:userId

POST /signup. Endpoint for authenticating user with login and returning JWT token for using other endpoints. It takes "username" and "password".

This JWT token needs to be placed under authentication tab/ Bearer Token to make most of the endpoints in this api work.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

userId

Items endpoints

These endpoints handles all that needs to be done in the Items table, Creating, Getting, Updating and Deleting items from the database.

POST Create Item

http://localhost:3000/item

Endpoint for creating new items in the database. Requires Admin user.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```json
{
  "item_name": "Item Name",
  "price": "Item Price",
  "categoryId": "Category Id",
  "description": "Item Description",
  "sku": "Item SKU",
  "stock_quantity": "Item Quantity"
}
```

GET Get all Items

http://localhost:3000/items

Endpoint that returns all the items in the database.

AUTHORIZATION Bearer Token

Token	<token>

GET Get Item by Id

http://localhost:3000/item/:itemId

Endpoint that return a spesific item in the database.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

itemId

PUT Update Item

http://localhost:3000/item/:itemId

Endpoint that updates an item in the database. Requires Admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

itemId

Body raw (json)

json

```
{
  "item_name": "New Item Name",
  "price": "New Item Price",
  "categoryId": "New Category Id",
  "description": "New Item Description",
  "sku": "New Item SKU",
  "stock_quantity": "New Item Quantity",
  "img_url": "New Image URL"
}
```

DELETE Delete Item

http://localhost:3000/item/:itemId

Endpoint that deletes an item from the database. Requires Admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

itemId

Categories endpoints

These endpoints handles all that needs to be done in the categories table, Creating, Getting, Updating and Deleting categories from the database.

GET Get all Categories

http://localhost:3000/categories

Endpoint that returns all categories in database.

AUTHORIZATION Bearer Token

Token	<token>

POST Create Category

http://localhost:3000/category

Endpoint that creates new categories in database. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```json
{
  "category": "New Category Name"
}
```

GET Get Category by Id

http://localhost:3000/category/:categoryId

Endpoint that returns a spesific category based on category id.

PATH VARIABLES

categoryId

PUT Update Category

http://localhost:3000/category/:categoryId

Endpoint that Updates a category name. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

categoryId

Body raw (json)

```json
{
  "category": "Updated Category Name"
}
```

DELETE Delete Category

http://localhost:3000/category/:categoryId

Endpoint that deletes a category based on category id. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

categoryId

Carts endpoints

These endpoints handles the addition of items to users's cart, updating cart items, viewing a user's cart, and removal of the items from the cart/removal of the cart.
Admin users can also retrieve information about all customers.

GET Get all Carts(admin user)

http://localhost:3000/allcarts

Endpoint that returns all carts for all the customers in the database. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

GET Get User's cart

http://localhost:3000/cart

Endpoint that returns cart for currently logged in user.

AUTHORIZATION Bearer Token

Token	<token>

POST Add Item To Cart

POST Add Item To Cart

http://localhost:3000/cart_item

Endpoint that adds items to cart.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```json
{
  "itemId": "item ID",
  "quantity": "item quantity"
}
```

PUT Update Cart Item Quantity

http://localhost:3000/cart_item/:id

Endpoint that updates the quantity of items already in cart.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

id

Body raw (json)

```json
{
  "quantity": "updated quantity"
}
```

DELETE Delete Cart Item

http://localhost:3000/cart_item/:id

Endpoint that deletes an item from cart.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

id

DELETE Delete User's Cart

http://localhost:3000/cart/:cartId

Endpoint that deletes the users cart.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

cartId

Orders Endpoints

These endpoints creates new orders, retrieves order details, and updates the status of existing orders.

GET Get Orders User/Get all orders Admin

http://localhost:3000/orders

Endpoint that returns all orders for current user or all users if admin is logged in.

AUTHORIZATION Bearer Token

Token	<token>

POST Create New Order

http://localhost:3000/order/:itemId?

Endpoint that creates new order. Using itemId of the items currently in cart, takes one by one item and places it into the order, but if you don't input an itemId it places all cart items into the order at the same time.

AUTHORIZATION Bearer Token

Token	<token>

PARAMS

PATH VARIABLES

itemId

PUT Update Order Status(admin)

http://localhost:3000/order/:id

Endpoint that changes the status of an order. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

id

Body raw (json)

```
json

{
  "status": "New status"
}
```

GET Get All Orders(admin)

http://localhost:3000/allorders

Endpoint that returns a list of all orders in the database. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

GET Get Orders By Id(admin)

http://localhost:3000/allorders/:orderId

Endpoint that returns order based on order id. Requires admin user.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

orderId

Utility

These endpoints populates the database, makes an admin user(setup endpoint), and has a search function to search for

These endpoints populates the database, makes an admin user(/setup endpoint), and has a search function to search for items in the database (/search endpoint).

POST Setup Database

http://localhost:3000/setup

Endpoint that sets up the database importing items from the api and also makes an admin user.

POST Search Items

http://localhost:3000/search

Endpoint that returns items, categories etc based on the search criterias.

The fields in the body can be used to filter the items. All fields are optional. If no fields are provided, the request returns all items.

Body raw (json)

json

```
{
  "item_name": "some item",
  "category": "some category",
  "sku": "some sku"
}
```