

HC-EP

Detailed documentation for Exam Project which is a Stock-control and sales system for a warehouse, selling directly to customers from a catalogue.

Users can access their own cart and order, including CRUD.

whilst an Admin can do CRUD operations on items, categories and complete/delete orders.

Authentication

User login, this enables the user to see certain endpoints and the ability to perform CRUD on own cart- and order items.

Here each user will receive a JWT token when submitting valid credentials.

Registration is also done in this category, where new users are able to shop directly from our itemscatalogue.

POST /login

```
http://localhost:3000/login
```

login endpoint for registered users.

Users send their credentials (username + password) in the request body

POST /signup

```
http://localhost:3000/signup
```

Registration for new users.

Users must provide username, firstname, lastname, email and password in the body request.

Various checks are done to check that it's the right format, no duplicate usernames etc.

The same email can be used on up to 4 users, this also provides a generous discount when multiple family members use the same email.

User details are then stored in the database,

Cart endpoints

Here lies all endpoints relevant to Carts and cartitems.

Users can add items to their carts, update desired quantity and delete if they change theyre mind on an item.

Users can also view their own cart, and Admin can view all carts that exists, including full names of the person ordering said carts.

GET /cart

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

A place to users to view their cart. Guest user have no access to this endpoint.

Users with items in their cart will get info about price, quantity etc about each item.

Users with noitems will be told no items are in the cart.

DELETE /cart/:id

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

Users can delete their cart in its entierty at this endpoint, this is done by them providing their cartid in the URL.

They can see they're cart ID in the get/cart endpoint.

PATH VARIABLES

id

POST /cart_item

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

This endpoint is used to add items to the users cart, users can only have one cart so either one will be created or it will put in the same cart if you allready have one.

Putting the same itemId in multiple times will update the quantity of that item, instead of adding multiple lines. Keep in mind that you cant add more to cart than what is available in the warehouse.

itemId and quantity needs to be send in the request body.

PUT /cart_item/:id

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

This endpoints update the wanted quantity in the item which is specified in the URL.

quantity must be send in the body request, and it cannot be higher than what is currently in stock.

PATH VARIABLES

id

DELETE /cart_item/:id

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

Users can from this endpoint remove an item from their cart. the id of the item is specified in the url.

PATH VARIABLES

id

GET /allcarts

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

This endpoint is only for Admin.

The endpoint will return every cart that is currently available, returning the items in the cart, quantity of said item and the fullname of the user the cart belongs to.

Category endpoint

All category related endpoints.

This is endpoints mostly for Admins where they can add, update and delete categories, however guests and users are able to check what categories the shop has in its store.

POST /category

```
http://localhost:3000/category
```

Endpoint only available for Admin.

This endpoint adds a new category to the `categorystable` in the database. `Categoryname` must be unique and be sent in the body.

PUT /category/:id

```
http://localhost:3000/category/:id
```

Admin accessible endpoint, that allows to change the name of an existing category.

The id of the category you want to change goes in to the url, whilst the `categoryname` is required in the request body.

Some checks are done to make sure the `categoryname` can't be set to empty and/or a name that doesn't give enough info. Also can't be the same name as an already existing category.

PATH VARIABLES

id

GET /categories

Open endpoint accessible by everyone.

This endpoint returns a list of all categories in the database, so everyone can quickly check what categories are in the catalogue.

DELETE /category/:id

```
http://localhost:3000/category/:id
```

Admin accessible endpoint only.

Admin can delete a category that is no longer needed. The id of the category that should be deleted goes in the url.

A category cannot be deleted if items exist in the items database with that category id.

PATH VARIABLES

id

Item endpoints

All itemendpoints.

Guests can view items in the database, provided they are in stock.

Users can view items, regardless of stockstatus.

Admin can add, update and delete items as needed.

PUT /item/:id



```
http://localhost:3000/item/1
```

Admin accessible endpoint, which is used to update items either name, image, price, sku, imagelink or categoryId can be updated alone or all together.

Various checks are in place to ensure that name and sku must be unique, and categoryId must be a valid Id.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
  "price": 500,
  "Quantity": 5,
  "CategoryId": 1,
  "image": "http://exampe.com/imagetest"
}
```

DELETE /item/:id

```
http://localhost:3000/item/:id
```

Admin accessible endpoint, which deletes and item that is no longer needed.

The id of the item is put in the url.

PATH VARIABLES

id

POST /item

http://localhost:3000/item

StartFragment

Admin accessible endpoint.

This endpoint is used to add a new item to the database, the data of the item goes into the requestbody.

Various checks are done to avoid among others duplicate itemnames and duplicate SKU's.

EndFragmen

Body raw (json)

json

```
{
  "name": "noQuantityItem",
  "price": 65,
  "SKU": "FR999",
  "Quantity": 0,
  "CategoryId": 1
}
```

GET /items

Endpoint that can be viewed by guests and users.

Users will see all items, including those that are currently not in stock, whilst guests will not see items that are currently out of stock.

Order endpoints

All orderrelated endpoints,

These endpoints are not available for guests. Only users, and/or Admin.

These endpoints allows users to view they're existing orders, and add items to their order.

Admin can see ALL orders, whilst also updating the status of these orders to either In proccess, Canceled or Complete.

PUT /order/:id

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

Admin accessible endpoint only.

Admin use this endpoint to update the status of Orders. the id of the selected Order you want to update goes into the url.

PATH VARIABLES

id

GET /allorders

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

Admin accessible endpoint, where an admin can view ALL orders currently in the system.

GET /orders

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

Endpoint that allows for users to view their orders, regardless of status.

When an Admin visits this endpoint ALL orders from ALL users are returned.

POST /order/:id

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

Endpoint that updates your order, with the selected items from your cart.

the itemId in your cart is the id you put in the url.

When the id is sent the cartitem gets removed, and added to the order and the item stock level in the itemstable is updated to reflect the new stock level. If there is not enough stock the order will not go through.

PATH VARIABLES

id

Misc

Misc endpoints are under this category, which in this case is the Search endpoint which allows for item, category and sku searches.

Also the setup endpoint which is used for initial database population.

POST /setup

```
http://localhost:3000/setup
```

Endpoint to populate the database with an API call.

This endpoint populates the Category and Items table, adds two roles to the roles table and adds an Admin user.

If the database is already populated nothing will happen.

POST /search

```
http://localhost:3000/search
```

This endpoint is accessible for everyone, including guests.

This endpoint allows to search for:

- Partial itemnames, either with or without a specific Category.
- Specific Categories
- Specific SKU's