_Noroff Exam Project

Documentation for API as described in the final exam assignment.

API is secured with Json Web Token. Some endpoint allow anonymous users, while other require authenticated user or admin account. Auth details are provided for each endpoint.

AUTHORIZATION Bearer Token

Token	<token>

Auth

Auth endpoints allow signup of new users and login for both users and admin.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

POST Signup

http://localhost:3000/signup

Create a new user account.

Endpoint open for anonymous users.

Requred fields are shown in examle below. Will respond 400 Bad Request and report if any required fields are missing for request.

Username must to be unique for each user.

Up to 4 user accounts can be registered to one single email address.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```
{
    "firstName": "John",
    "lastName": "Smith",
    "username": "johnsmith",

    "email": "john.smith@app.com",
    "password": "s3cr3tP@ssw0rd"
}
```

POST Login

http://localhost:3000/login

Login for registed users.

Endpoint open for anonymous users.

Successful login will provide a JWT token valid for 2 hours.

Requred fields are shown in examle below. Will respond 400 Bad Request and report if any required fields are missing for request.

Invalid username, password or both will result the same 400 bad request response.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```
{
    "username" : "johnsmith",
    "password" : "s3cr3tP@ssw0rd"
}
```

Items

Endpoints for items.

Anonymous users are allowed to get all items on request, while create, update and deleting items require authenticated user with admin role.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

GET Get All Items

http://localhost:3000/items

Get all item entities include nested category.

Anonymous users allowed.

NOTE: Guest user can only see in stock items.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

POST Post new Item

http://localhost:3000/item

Creates a new item entity.

Autheticated admin required.

Requred fields are shown in examle below. Will respond 400 Bad Request and report if any required fields are missing for request.

SKU (Product Code) must be unique for the item. Will result a 400 response if existing SKU is provided.

Item name is not required to be unique, but a warning on creating with existing name is returned with the response.

Note that a valid category ID is required for creating a new item entity.

Price and Stock Quantity can be set to 0, but can not be negative and will throw a bad request if provided.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```json
{
    "itemName": "Cellphone",
    "categoryId": 3,
    "imageUrl": "none",
    "sku": "EXMP_5",
    "price": 100,
    "stockQuantity": 10
}
```

PUT Update Item

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

Update an item entity.

Autheticated admin required.

Follows the same pattern as a post request, but no fields are required in the request body. Will only update data from provided fields.

Reponds 404 not found if Item ID not valid.

SKU (Product Code) must be unique for the item. Will result a 400 response if new existing SKU is provided.

Item name is not required to be unique, but a warning on replacing with existing name is returned with the response.

Note that a valid category ID is required for updating the item entity.

Price and Stock Quantity can be set to 0, but can not be negative and will throw a bad request if provided.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	167 Integer - Item ID

Body raw (json)

json

```json
{
    "itemName": "Cellphone",
    "categoryId": 3,
    "imageUrl": "none",
    "sku": "EXMP_5",
    "price": 250,
    "stockQuantity": 10
}
```

DELETE Delete Item

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

Deletes a item entity by provided ID in path variable.

Autheticated admin required.

Will result a 404 response if invalid ID was provided.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	167 Integer - Item ID

Category

Endpoints for Category entities.

Anonymous users are allowed GET requests retriving all category items. Other request types require admin role for authenticated user.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

GET Get All Categories

```
http://localhost:3000/categories
```

Get all category items.

Anonymoyus users allowed.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

POST Post new Category

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

Creates a new Category

Autheticated admin required

Requred fields are shown in examle below. Will respond 400 Bad Request and report if any required fields are missing for request.

Category names are unique values. Providing an existing category name will result a 400 response.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```json
{
    "name": "New Category"
}
```

PUT Update Category

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

Update a category with entity id set in the path variable.

Autheticated admin required

Requred fields are shown in examle below. Will respond 400 Bad Request and report if any required fields are missing for request.

Respond a 404 not found if provided id not valid.

Do not allow updating to an existing category name, request will result a 400 response.

API do not accept the nonsense of updating the existing name with the existing name. Request will result the exotic reponse 406 - Not Acceptable.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	2 Integer - Category Id

Body raw (json)

json

```json
{
    "name" : "Category 1"
}
```

DELETE Delete Category

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

Deletes a category item from provided ID in path variable.

Require autheticated user with admin role.

Will respons a 404 not found if invalid id was provided.

Cannot delete a category entity if it has a dependent item entity, and will result a 400 response.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	36 Integer - Category ID

Cart

Endpoint for managing carts.

All endpoint required autheticated user. Only admin user can extract data for all user carts.

Cart does not exist for new users, but will be created when the first item gets added.

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

GET User Cart

http://localhost:3000/cart

Get cart registered to logged in user.

Require autheticated user with User role.

Will return not found if cart does not exist. A cart item must be added to create a new cart if it does not exist

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

GET All Carts (Admin)

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

Return all carts to all user including cart items and basic user data.

Require autheticated user with Admin role.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

POST Add Cart Item

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

Add a new cart item to cart.

Items can be added by providing a valid item id. Amout is not required as missing valie will set 1 as default.

Only new items can be added. Use PUT request to edit existing cart items.

Autenticated user required!

See examples below for required fields in request body. If required values are missing, API will return a 400 response reporting the missing values.

Will report 404 if a non existing item id is provided.

Will report 400 if item exists in cart.

Will report 400 if item is out of stock or requested amout exceed items in stock.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```json
{
    "itemId": 157,
    "amount": 1
}
```

PUT Update Cart Item

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

Updates the amout of items in a cart item

NOTE: The path variable must reference the Item ID, and **NOT** the Cart Item ID

Authenticated user required!

See examples below for required fields in request body. If required values are missing, API will return a 400 response reporting the missing values.

Will return 404 if provided item ID does not exist among cart items in user cart.

Will return 400 if new amout requested exceed items in stock.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	132 Integer - Item ID

Body raw (json)

json

```json
{
    "newAmount": 1000
}
```

DELETE Delete Cart item

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

Removes an item from cart.

NOTE: The path variable must reference the Item ID, and **NOT** the Cart Item ID

Require authenticated user.

Will respond 404 if invalid cart item ID is provided

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	135 Integer - Cart Item ID

DELETE Delete All Cart Items

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

Deletes all items from cart.

Require autheticated user with User role.

Will respond 404 if provided cart ID is invalid.

NOTE that this method only removes all existing items from cart entity. The cart entity will remain thought wihtout any related entities in database.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	1 Integer - Cart ID

Orders

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

GET Get Order (User)

http://localhost:3000/orders

Get all orders for user if user signed in.

Logged in admin will get same result as from **GET /allorders.**

Endpoint require signed in user with Admin or User role.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

GET Get All Orders (Admin)

http://localhost:3000/allorders

Get all orders for all users

Required logged in user with Admin role

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

POST Post new Order (User)

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

Check out cart and place it as an order. Will remove all cart items as they are transformed to order item. Will also subtract requested amout in cart from items in stock.

Require logged in user with User role.

Will respond with 404 if cart Id is invalid or doesn't belong to logged in user.

Will respond with 404 if not items are added to cart.

Will respond 400 if requested cart item amout exceed items in stock, or if item is out of stock.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	1 Integer - Cart ID

PUT Update Order (Admin)

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

StartFragment

Update a category with entity id set in the path variable.

Change status for order with entity id set in the path variable.

Valid statues are COMPLETE, IN PROCESS and CANCELLED.

Require autheticated user with admin role.

See examples below for required fields in request body. If required values are missing, API will return a 400 response reporting the missing values.

Respond a 404 not found if provided order does not exist.

Respond 400 if invalid status is provided

EndFragment

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

PATH VARIABLES

id	1 Integer - Order ID

Body raw (json)

json

```json
{
    "orderStatus": "complete"
}
```

Utils

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection _Noroff Exam Project

POST Setup endpoint

http://localhost:3000/setup

This endpoint inserts a collection of values to database, as well as importing data from external sources.

Will always respond a report on what have been created.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

POST Search

http://localhost:3000/search

Search by SKU, category or partial item name.

If sku is provded, other values are ignored and item is searched by SKU only.

Categories require full name. Item name can be partial.

SKU search will provide a single item, while search by partial item name and / or category will return a list.

A 404 response is provided if no items are found.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project

Body raw (json)

json

```json
{
    "itemName": "mar"
}
```

GET Alive@Index

http://localhost:3000

Base url will respond 200 I am alive if reciving a get request.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection _Noroff Exam Project