APIs

Auth

POST SignUp

http://localhost:3000/users/signup

This API endpoint handles user signup. It validates the input data and creates a new user in the database with a hashed password and salt.

Body raw (json)

```json
{
    "username": "test",
    "email": "test@gmail.com",
    "password": "1234"
}
```

POST LogIn

http://localhost:3000/users/login

This API validates user login credentials and generates a JSON Web Token upon successful login. If login credentials are invalid, it returns an error message accordingly.

Body raw (json)

```json
{
    "email": "gmail.com",
```

```
    "password": "1234"
}
```

Todo

AUTHORIZATION Bearer Token

Token	<token>

POST Insert Todo

http://localhost:3000/create

This API creates a new todo item for a specific user using the provided todo name and category. The user ID is obtained from the authorization token, and the API responds with a success status code of 201 and the created todo, or a failure status code of 400 along with an error message if the todo creation fails.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```
{
    "name": "exam",
    "category": "WorkFake"
}
```

GET View Todo

http://localhost:3000/

This API returns all the todos for a specific user based on their ID obtained from the authorization token in the request headers. The response contains the todos in JSON format with a success status code of 200, and sends a failure status code of 404 if no todos are found.

AUTHORIZATION Bearer Token

Token	<token>

PUT Edit Todo

http://localhost:3000/edit

StartFragment

This API edits the name of a specific todo item using the new todo name provided in the request body and the original todo name. If successful, the API sends a success status code of 200 in the response indicating that the todo was updated, otherwise it sends a failure status code of 404 or 400, indicating that the todo was not found or an error occurred while updating the todo.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```
{
    "name": "exam26",
    "todoName": "examFake"
}
```

DELETE Delete Todo

http://localhost:3000/delete

This API endpoint deletes a specific todo item based on the todo name provided in the request body. If the deletion is successful, the API sends a success status code of 200 in the response with a message indicating that the todo was deleted, otherwise it sends a failure status code of 404 indicating that the todo was not found or the user does not have permission, or a failure status code of 400 indicating that an error occurred while deleting the todo.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```
json

{
    "todoName": "examFake"
}
```

Category

POST Insert Category

http://localhost:3000/category/create

This API endpoint creates a new category based on the category name provided in the request body. If the category already exists, the API sends a failure status code of 404 indicating that the category already existed. If the creation is successful, the API sends a success status code of 201 in the response with a message indicating that the category was created, otherwise it sends a failure status code of 400 indicating that an error occurred while creating the category.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```
json

{
    "nae": "Work"
}
```

GET View Category

http://localhost:3000/category

This API fetches all existing categories and sends them in the response to the client. If no categories are found, it returns a 404 status code, and for any error, it returns a 500 status code.

AUTHORIZATION Bearer Token

Token	<token>

PUT Edit Category

http://localhost:3000/category/edit

This API updates a category's name in the database based on its old name. If successful, it returns a success message, otherwise it returns an error message.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```json
{
    "name": "Work2",
    "categoryName": "Workfake"
}
```

DELETE Delete Category

http://localhost:3000/category/delete

This API deletes a category from the database based on the given category name in the request body. If the category is successfully deleted, the function returns a success message with a 200 status code. If the category is not found or the user does not have permission to delete it, the function returns a fail message with a 404 status code.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

json

```json
{
    "categoryName": "Work2"
}
```