Todo

Todo” Application, which allows users to create, view, update and delete “Todo” items.

AUTHORIZATION Bearer Token

Token	<token>

Auth

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection Todo

POST Sign Up

http://localhost:3000/signup

The `/signup` endpoint enables user registration by providing their name, email, and password. The API checks for required fields, validates the email, and securely stores the password. If successful, a new account is created and a success message is returned. In case of missing fields or duplicate email, a failure message with error details is sent.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

Body raw (json)

json

```json
{
    "name": "ismail kaya",
    "email": "tor232191@hotmail.com",
    "password": "123456"
}
```

POST Login

http://localhost:3000/login

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

Body raw (json)

json

```json
{
    "email": "ismail@yahoo.com",
    "password": "1234"
}
```

DELETE Delete Account

http://localhost:3000/

The `/` DELETE endpoint allows for account deletion by providing an email in the request body. The API first checks for the email's presence and validates if it exists in the system. If the email is found, the associated account is deleted, and a success message is returned. If the email is missing or not registered, a failure message with error details is sent.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

Body raw (json)

json

```json
{
    "email":"tor232191@hotmail.com"
}
```

Todo

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection Todo

GET Get Todos

http://localhost:3000/todos/

The `/` GET endpoint retrieves a list of todos for an authenticated user. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the user's todos are fetched and returned as a success response. In case of missing or invalid token, an appropriate failure message is sent. If fetching todos fails, an error message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

POST Add Todo

http://localhost:3000/todos/

The `/` POST endpoint allows authenticated users to create a new todo. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the new todo is created with the provided `name` and `CategoryId` from the request body, and the success response includes the created todo. In case of missing or invalid token, an appropriate failure message is sent. If creating the todo fails, an error message with additional error details is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

Body raw (json)

json

```json
{
    "name":"test",
    "CategoryId":1
}
```

PUT Update Todo

http://localhost:3000/todos/:id

The `/:id` PUT endpoint allows authenticated users to update an existing todo by specifying its ID. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the todo with the given ID is updated using the provided `name` and `CategoryId` from the request body, and a success response containing the updated todo is returned. In case of missing or invalid token, a failure message is sent. If the todo is not found or updating the todo fails, an appropriate error or failure message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

PATH VARIABLES

id	16

Body raw (json)

json

```json
{
    "name":"test update",
    "CategoryId":1
}
```

DELETE Delete Todo

http://localhost:3000/todos/:id

The `/:id` DELETE endpoint enables authenticated users to delete an existing todo by providing its ID. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the todo with the given ID is deleted, and a success response containing the deleted todo is returned. In case of missing or invalid token, a failure message is sent. If the todo is not found or deleting the todo fails, an appropriate error or failure message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

PATH VARIABLES

id	3

Category

AUTHORIZATION Bearer Token

This folder is using Bearer Token from collection Todo

GET Get Categories

http://localhost:3000/categories/

The `/` GET endpoint retrieves a list of categories for an authenticated user. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the user's categories are fetched and returned as a success response. In case of missing or invalid token, an appropriate failure message is sent. If fetching categories fails, an error message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

POST Add Category

http://localhost:3000/categories/

The `/category/` POST endpoint enables authenticated users to create a new category. The API first verifies the presence and validity of an authorization token in the request headers. If the token is valid, a new category is created using the provided `name` from the request body, and a success response containing the created category is returned. If the token is missing or invalid, a failure message is sent. In case of an error in creating the category, an error message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

Body raw (json)

json

```json
{
    "name":"new category"
}
```

PUT Update Category

http://localhost:3000/categories/:id

The `/:id` PUT endpoint allows authenticated users to update an existing category by specifying its ID. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the category with the given ID is updated using the provided `name` from the request body, and a success response containing the updated category is returned. In case of missing or invalid token, a failure message is sent. If the category is not found or updating the category fails, an appropriate error or failure message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

This request is using Bearer Token from collection Todo

PATH VARIABLES

id	4

Body raw (json)

json

```json
{
    "name":"updated"
}
```

DELETE Delete Category

http://localhost:3000/categories/:id

The `/:id` DELETE endpoint enables authenticated users to delete an existing category by providing its ID. The API first checks for a valid authorization token in the request headers. If the token is present and valid, the category with the given ID is deleted, and a success response containing the deleted category is returned. In case of missing or invalid token, a failure message is sent. If the category is not found or deleting the category fails, an appropriate error or failure message is returned.

AUTHORIZATION Bearer Token

This request is using Bearer Token from collection Todo

PATH VARIABLES

id	5