This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

TodoAPIDocumentation

This is an Api to help you keep track of tasks that need to be done, and categories to assign the tasks to. In this Api, you can save the taskt to the database, view you tasks, update them, and delete them when done. You can also create, view, update and delete categories for your tasks. The Api requires you to create a user by signing up, then logging in to acces the todo functionality

Authenticaiton

These request are used to signup to the Api, and logging in so that the functionality becomes available.

POST Sign up new user

http://localhost:3000/signup

The signup request takes three parameters in the body of the request:

- Name
- Email
- Password

Below are a successfull and a negative example for this request.

Body raw (json)

json

```
}
    "name": "example",
    "email": "example@gmail.com",
    "password": "examplePassword"
}
```

POST Login user to acces functionality

http://localhost:3000/login

The login requrest takes two parameters in the request body:

- Email
- Password

This requrest will return a token in the response body. Copy this token and place it as a Bearer token in the authorization tab to use the todo-functionality.

Below are a successfull and a negative example for this request.

Body raw (json)

json

```
{
    "email": "example@gmail.com",
    "password": "examplePassword"
}
```

Basic Todo Operations

Here are examples for the todo-request.

To access all these, take the token from the login request, and paste it as a bearer token in the

autorizations tab.

GET Get all todo items

http://localhost:3000/todo

This Get request return a list of all items that have been logged in the database. If there are no items available, the request is still successfull, but message will prompt you that there are no items to view.

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token	<token>

POST Create new todo item

http://localhost:3000/todo

This POST request creates a new item and saves it to the database. The requrest takes two parameters in the requrest body:

- Name - string
- Category - integer, id of the category it belongs to.

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token	<token>

Body raw (json)

```
json
```

```
{
    "name": "Dusting",
    "category": 1
}
```

PUT Update todo item by name

http://localhost:3000/todo

This PUT request updates one item that is already in the database. This request takes two parameters in the request body.

- oldName - Name of item you wish to update
- newName - New name for the updated item.

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
    "oldName": "Dusting",
    "newName": "Cleaning"
}
```

DELETE Delete todo item

http://localhost:3000/todo

This DELETE request deletes one item. The requrest takes a name of the item you wish to delete as parameter in the request body

parameter in the request body.

- Name

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

```json
{
    "name": "Cleaning"
}
```

Basic Category Operations

Here you will find examples for creating, viewing, updating and deleting categories for you api-tasks.

GET Get all categories request

http://localhost:3000/category

This request gets all avaiable categories for you to view. You need to be logged in, and paste the token in as a bearer token in the authtorization tab to access this request.

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token <token>

POST Create new category request

http://localhost:3000/category

This request created a new category for you to assign items to. This request takes a name as a parameter in the request body.

- Name

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

json

```
{
    "name": "Kitchen"
}
```

PUT Update category request

http://localhost:3000/category

This request updates a current category. Here you can change the name of a category by providing the old name, and the new name in the request body.

- oldName
- newName

Below are a successfull and a negative example for

this request.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

```
json

{
    "oldName": "Kitchen",
    "newName": "LivingRoom"
}
```

DELETE Delete category request

http://localhost:3000/category

This request allows you to delete a category. Provide the name of the category you want to delete in the request body.

- Name

Below are a successfull and a negative example for this request.

AUTHORIZATION Bearer Token

Token <token>

Body raw (json)

```
json

{
    "name": "LivingRoom"
}
```