

Notes API Documentation

Overview

Welcome to the Notes API documentation. Manage a collection of notes with the following endpoints:

1. Get All Notes

Retrieve a list of all notes.

- **URL:** GET `http://localhost:4000/api/note/`

2. Get Single Note

Retrieve a single note by its ID.

- **URL:** GET `http://localhost:4000/api/note/:id`

3. Post New Note

Create a new note.

- **URL:** POST `http://localhost:4000/api/note/`

4. Delete Note

Delete a note by its ID.

- **URL:** DELETE `http://localhost:4000/api/note/delete/:id`

5. Update Note

Update a note by its ID.

- **URL:** PATCH `http://localhost:4000/api/note/update/:id`

How to use

Overview

Welcome to the API documentation for the Notes API. This API allows you to manage and interact with a collection of notes.

Installation

1. Go inside the Backend dir and run command :- **"npm install"** it will install all the Node.js dependencies

2(optinal). Go to .env and change the **"CONECTIONSTRING"** to your mongoDB conection string otherwise live it by default it uses mine

The base URL for all API endpoints is: <http://localhost:4000>

Endpoints

1. Get All Notes

Endpoint

GET /api/note/

Description

Retrieve a list of all notes.

Request

- Method: GET
- URL: <http://localhost:4000/api/note/>

Response

- Status Code: 200 OK
- Body:

```
[
  {
    "_id": "note_id",
    "title": "Note Title",
    "note": "Note Content",
    "createdAt": "Timestamp",
    "updatedAt": "Timestamp"
  },
  // ... all notes
]
```

2. Get Single Note

Endpoint

GET /api/note/:id

Description

Retrieve a single note by its ID.

Request

- Method: GET
- URL: http://localhost:4000/api/note/:id

Response

- Status Code: 200 OK
- Body:

```
jsonCopy code
{
  "_id": "note_id",
  "title": "Note Title",
  "note": "Note Content",
  "createdAt": "Timestamp",
  "updatedAt": "Timestamp"
}
```

3. Post New Note

Endpoint

POST /api/note/

Description

Create a new note.

Request

- Method: POST
- URL: http://localhost:4000/api/note/
- Body:

```
{
  "title": "New Note Title",
  "note": "New Note Content"
}
```

Response

- Status Code: 200 OK
- Body:

```
{
  "message": "Note Saved Successfully!!",
  "color": "green"
}
```

4. Delete Note

Endpoint

```
DELETE /api/note/delete/:id
```

Description

Delete a note by its ID.

Request

- Method: `DELETE`
- URL: `http://localhost:4000/note/delete/:id`

Response

- Status Code: `200 OK`
- Body:

```
{
  "message": "Note Deleted Successfully!!",
  "color": "DarkPink"
}
```

5. Update Note

Endpoint

PATCH /api/note/update/:id

Description

Update a note by its ID.

Request

- Method: PATCH
- URL: http://localhost:4000/api/note/update/:id
- Body (Include only fields to be updated):

```
{  
  "title": "Updated Note Title",  
  "note": "Updated Note Content"  
}
```

Response

- Status Code: 200 OK
- Body:

```
{  
  "message": "Note Updated Successfully!!",  
  "color": "green"  
}
```