Note App Documentation

Node.js/MySQL/Docker/Redis/JWT

AUG 2023

RESPOND.IO

CREATED BY

QASIM HAFEEZ

API Routes

The following are guidelines & requirements apply to this API:

Users

Base Route: /api/v1/users

1. **Register**

   Route: /register

   Body: {
   “name”: “Qasim Hafeez”,
   “email”: “qasim@hafeez.com”,
   “password”: “your_password”
   }

   Response: {
   “message”: “User Created Successfully!”
   }

2. **Login**

   Route: /login

   Body: {
   “email”: “qasim@hafeez.com”,
   “password”: “your_password”
   }

   Response: {
   “token”: “jwt token”,
   “message”: “Logged In Successfully!”
   }

Notes

Base Route: /api/v1/notes

1. Create a new Note

Create a new note for a user.

Route: POST /

Request Body:
```
{
        "title": "Title of the note",
        "content": "Content of the note",
        "user_id": "ID of the user"
}
```

Response:

- Status Code: 201 Created
- **Content**: The created note object

2. Get All Notes for a User

Retrieve all notes associated with a user.

Route: GET /:userId

Headers: Authorization Bearer Token (JWT)

URL Parameters:

userId - The ID of the user for whom to retrieve notes

Response:

- Status Code: 200 OK
- **Content**: An array of note objects

3. Get a Note with User Information

Retrieve a specific note along with information about the associated user.

Route: GET /:noteId

Headers: Authorization Bearer Token (JWT)

URL Parameters:

noteId - The ID of the note to retrieve

Response:

- Status Code: 200 OK
- **Content**: The note object with associated user information

4. Update a Note

Update the title and content of a specific note.

Route: PUT /:noteId

Headers: Authorization Bearer Token (JWT)

URL Parameters:

noteId - The ID of the note to update

Request Body:{

"title": "Title of the note",

"content": "Content of the note",

"user_id": "ID of the user"

}

Response:

- Status Code: 200 OK
- **Content**: The updated note object

5. Delete a Note

Delete a specific note.

Route: DELETE /:noteId

Headers: Authorization Bearer Token (JWT)

URL Parameters:

noteId - The ID of the note to delete

Response:

- Status Code: 200 OK
- **Content**: A message confirming the successful deletion

The process of running Note Application

1. Pull and run Redis docker container

Commands:

```
- docker pull redis:latest
- docker run -d --name redis-container -p 6379:6379
  —network note-network redis:latest
```

2. Pull and run MySQL docker container

Commands:

```
- docker pull mysql:latest
- docker run --name mysql -e
  MYSQL_ROOT_PASSWORD=mysecretpassword -e
  MYSQL_DATABASE=note-app-db -p 3306:3306 -d
  —network note-network mysql:latest
```

3. Note App in docker container

Commands: (Run these commands in the project root directory)

```
- docker build -t note-app .
- Docker run —name note-app-container -p 8080:8080
  —network note-network note-app
```