Title: Api blogs note taker

Group Memebr: Yash Bavishi A003

Language sued: javascript

Database: Mongodb atlas

Module:

React js

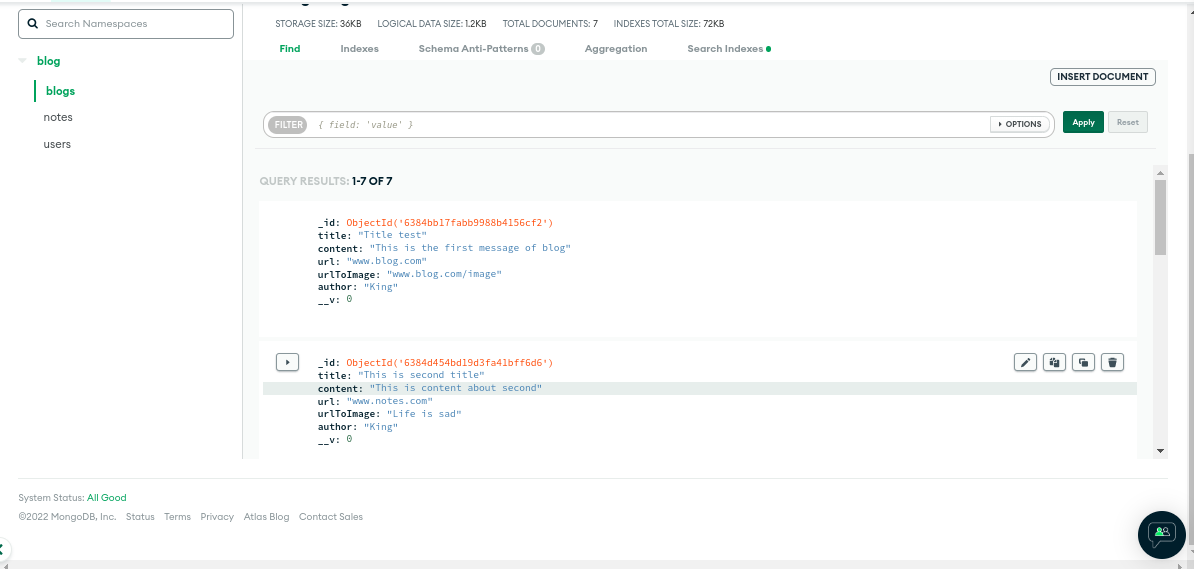
Node Js

Express js

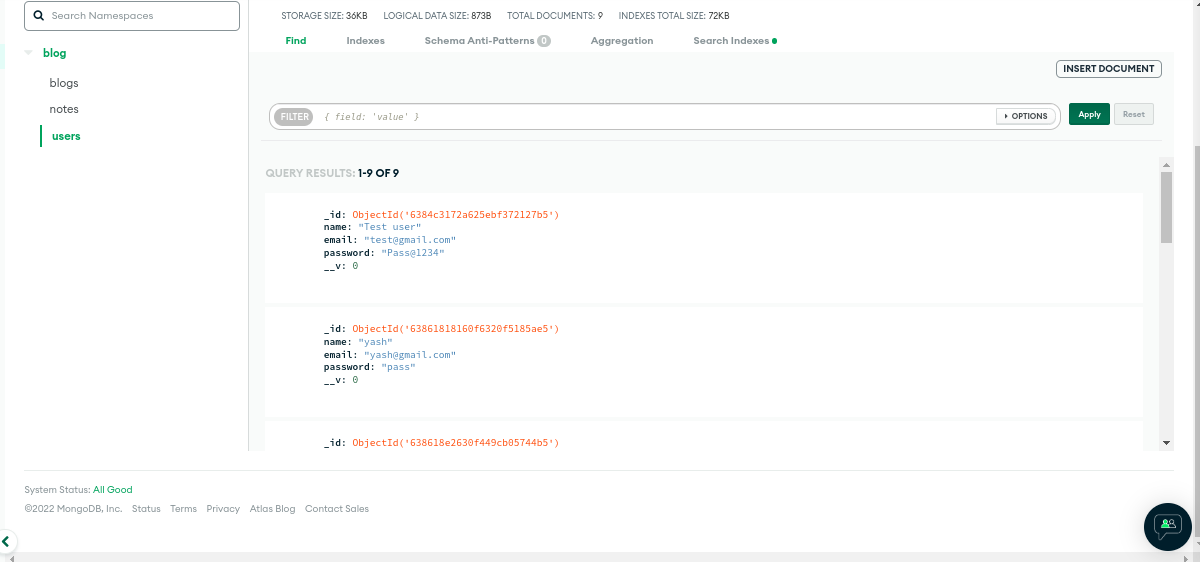
Mongoose

Data base collections

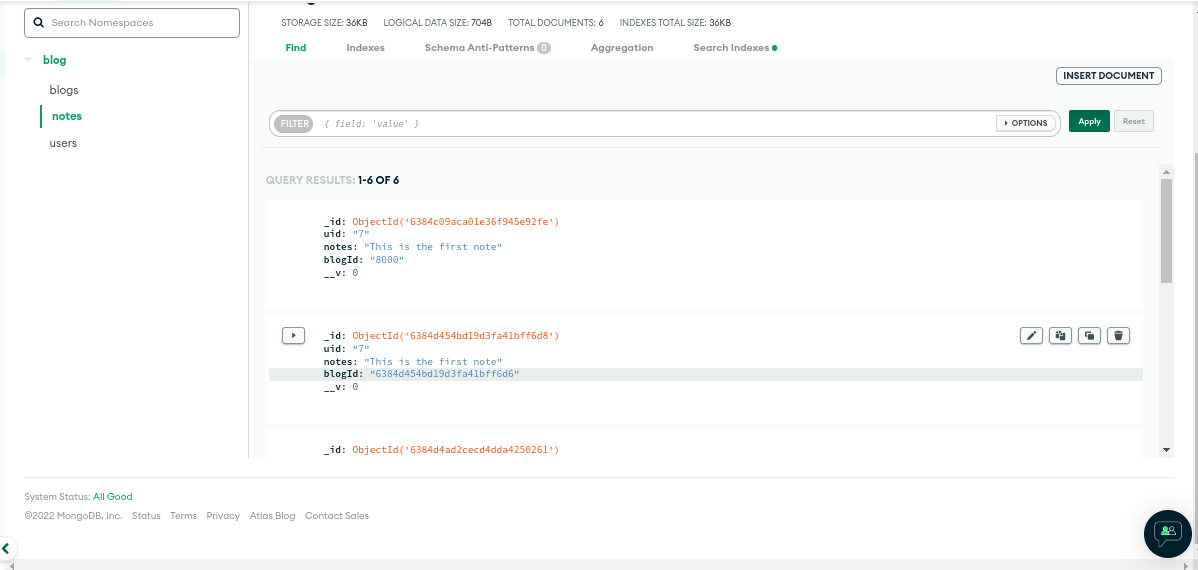
Blogs collection



User collection

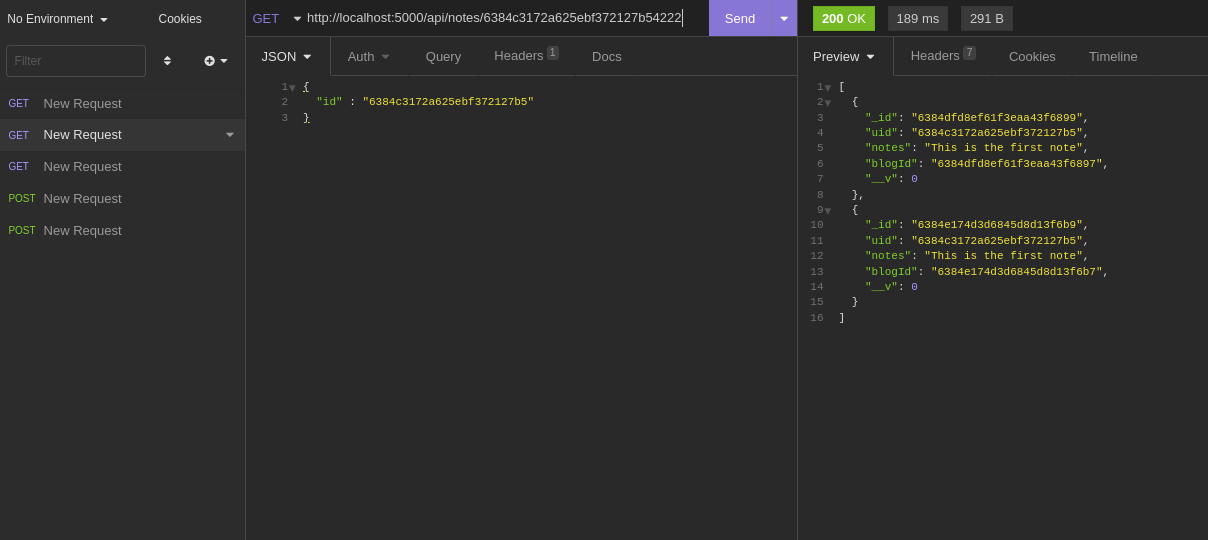


Notes collection

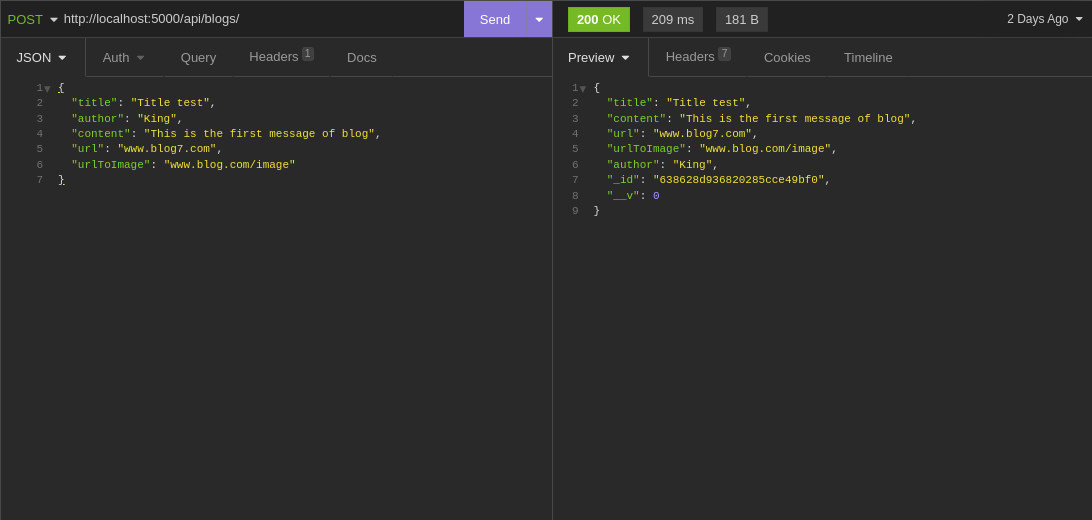


API Route

1. Getting notes fro user



1. Sending blogs to data api route



Main.js

import React from 'react';

import { useState, useEffect } from 'react'

import Blog from './Blog'

//const mongoose = require("mongoose")

let connectionString = 'mongodb+srv://admin:admin@blog.yku72j1.mongodb.net/?retryWrites=true&w=majority'

function traverse(item, index) {

return index + ": " + item.title

}

function Main() {

const [item, setItem] = useState([])

const [isLoaded, setIsLoaded] = useState(false)

const [error, setError] = useState(null)

useEffect(() => {

fetch('https://newsapi.org/v2/everything?q=cars&from=2022-11-26&pageSize=10&sortBy=publishedAt&apiKey=d098bd1c8b71425fa640061e0158648c&language=en')

.then(res => res.json())

.then(

(result) => {

setItem(result);

setIsLoaded(true)

},

(error) => {

setIsLoaded(false)

setError(error);

}

)

}, [])

let arr = null

if (isLoaded) {

arr = item.articles

}

return (

<div className="main">

<div className="blogs">

{

isLoaded ? arr.map((article, index) => <Blog blog={article} key={index} />) : console.log("error")

}

</div>

<div className="notes">

<form action="">

<textarea id="notes" name="notes" cols="30" rows="10" placeholder="Notes section"></textarea>

</form>

<button>Save to database</button>

</div>

</div>

)

}

export default Main

Routes.js

const express = require("express")

const router = express.Router()

const blog = require('./models/blog')

const notes = require("./models/notes")

const users = require("./models/user")

// Blog

router.get('/blogs', async (req, res) => {

const blogs = await blog.find()

res.send(blogs)

})

router.post('/blogs', async (req, res) => {

const sendBlogs = new blog({

title: req.body.title,

content: req.body.content,

author: req.body.author,

url: req.body.url,

urlToImage: req.body.urlToImage,

})

await sendBlogs.save()

console.log(sendBlogs)

res.send(sendBlogs)

})

// notes

router.get('/notes/:id', async (req, res) => {

const user\_id = req.params.id;

const note = await notes.find({ uid: user\_id })

note ? res.send(note) : res.send({ msg: 'no user found' })

console.log(note[0].blogId)

const blogs = await blog.find({ \_id: note[0].blogId })

console.log(user\_id)

console.log(blogs)

})

/\*

router.get('/notes', async (req, res) => {

const note = await notes.find()

const blogs = await blog.findOne({ \_id: note[1].blogId })

console.log(note[1].blogId)

console.log(blogs)

res.send(blogs)

})

\*/

router.post('/notes', async (req, res) => {

const sendBlogs = new blog({

title: req.body.title,

content: req.body.content,

author: req.body.author,

url: req.body.url,

urlToImage: req.body.urlToImage,

})

await sendBlogs.save()

console.log(sendBlogs)

const sendNotes = new notes({

uid: req.body.uid,

notes: req.body.notes,

blogId: sendBlogs.\_id

})

await sendNotes.save()

console.log(sendNotes)

res.send(sendNotes)

})

// user

router.get('/user', async (req, res) => {

const user = await users.find();

res.send(user)

})

router.post('/user', async (req, res) => {

console.log(req.body.json)

if (!req.body.name || !req.body.email || !req.body.password) {

res.status(400).json({

"msg": "Not all parameters found"

})

} else {

const sendUser = new users({

name: req.body.name,

email: req.body.email,

password: req.body.password

})

await sendUser.save()

console.log(sendUser)

res.status(200)

}

})

module.exports = router

‘

Server.s

const mongoose = require("mongoose")

const express = require("express");

const app = express();

const routes = require("./routes");

const PORT = process.env.PORT || 5000;

// Connection string

const connectionString = 'mongodb+srv://admin:admin@blog.yku72j1.mongodb.net/blog?retryWrites=true&w=majority'

mongoose.connect(connectionString, {

useNewUrlParser: true,

useUnifiedTopology: true

})

.then(() => {

const app = express();

app.use(express.urlencoded({ extended: true }))

app.use(express.json())

app.use('/api', routes)

app.listen(PORT, () => {

console.log("Server has started");

})

})

console.log("sad")