

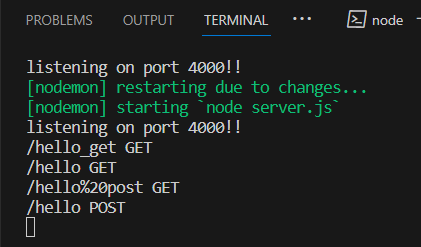
Create a sample middleware to get requested requests using next() method

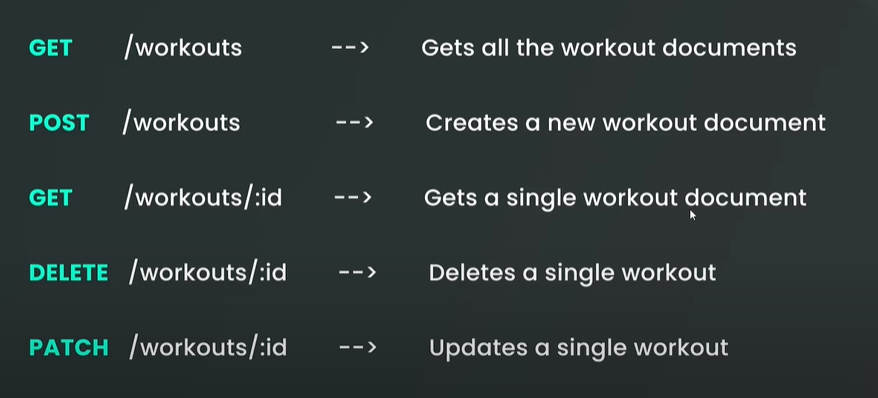
//middleware

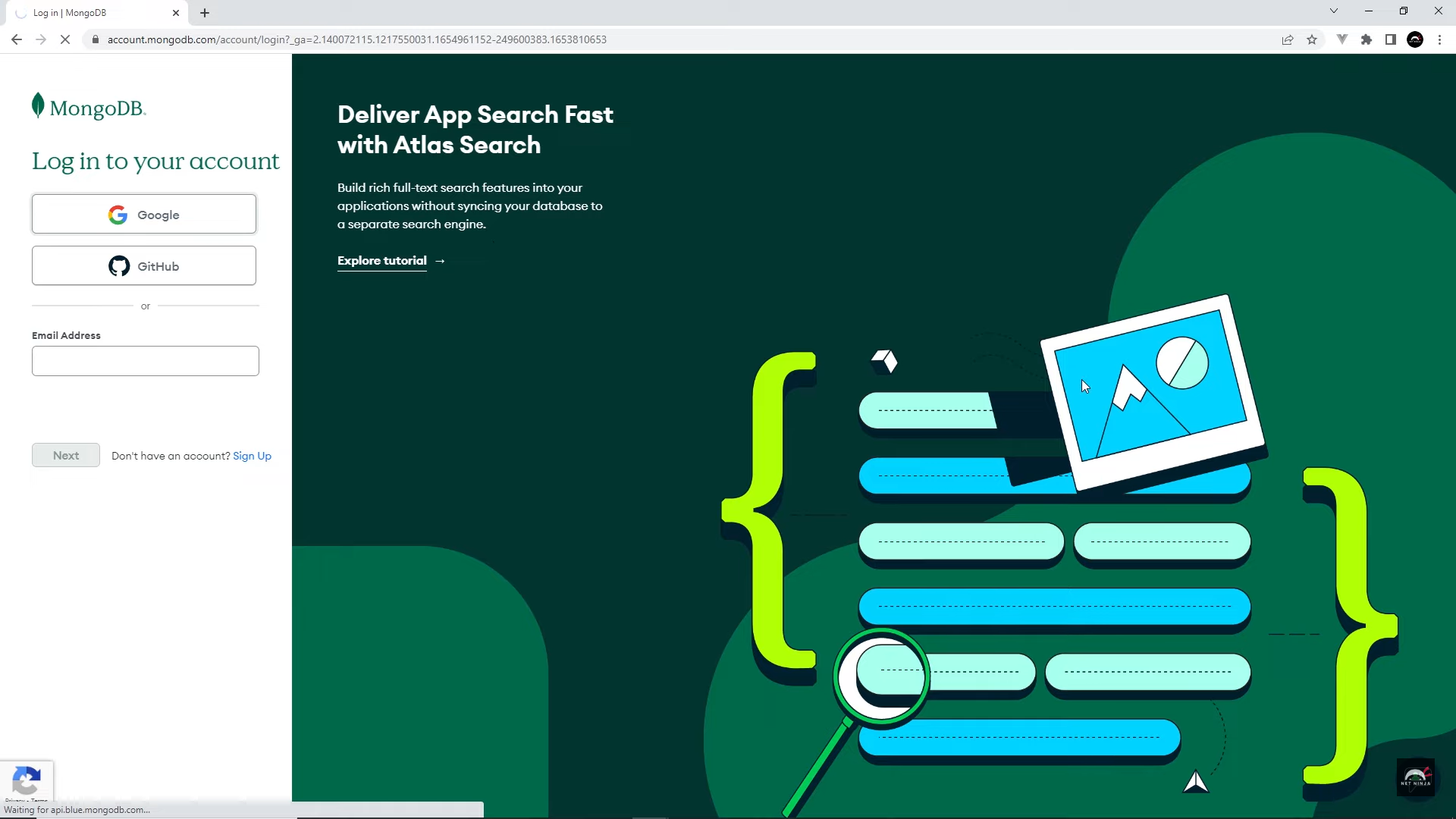
app.use((req,res,next)=>{

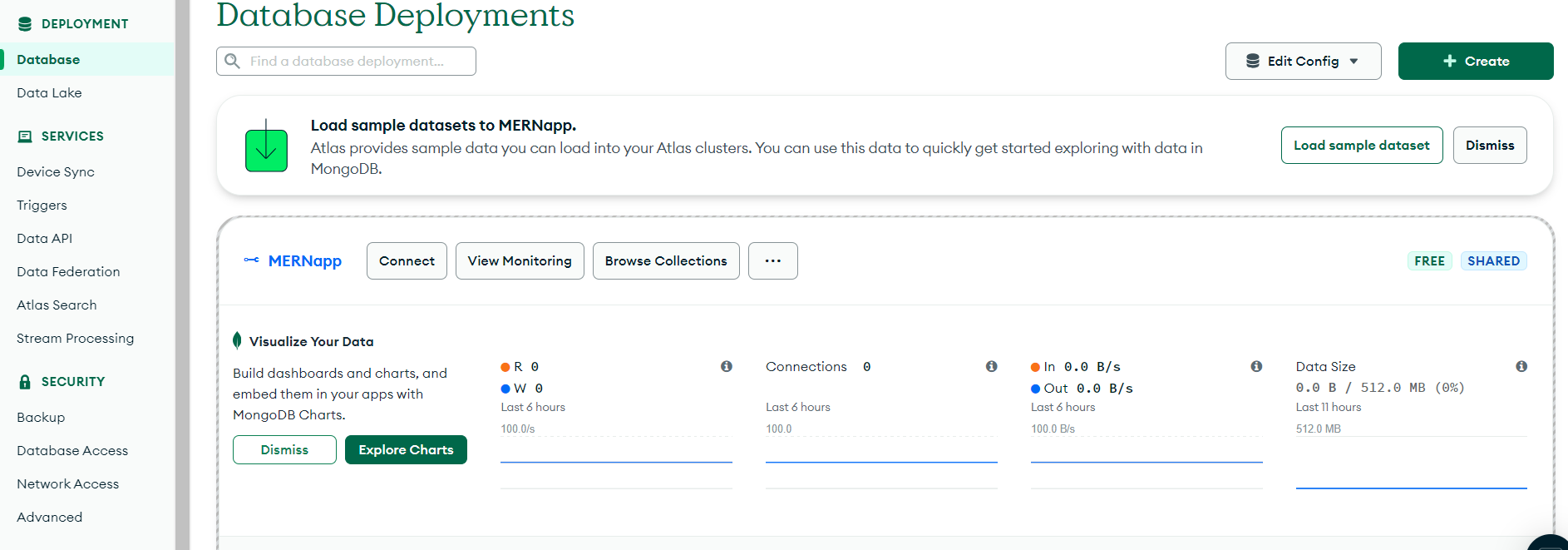
    console.log(req.path,req.method)

    })









Final server.js:

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const workoutRoutes = require('./routes/workouts')

// express app

const app = express()

// middleware

app.use(express.json())

app.use((req, res, next) => {

  console.log(req.path, req.method)

  next()

})

// routes

app.use('/api/workouts', workoutRoutes)

// connect to db

mongoose.connect(process.env.MONGO\_URI)

  .then(() => {

    console.log('connected to database')

    // listen to port

    app.listen(process.env.PORT, () => {

      console.log('listening for requests on port', process.env.PORT)

    })

  })

  .catch((err) => {

    console.log(err)

  })

Final workout.js:

const express = require('express')

const router = express.Router()

// GET all workouts

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

  res.json({mssg: 'GET all workouts'})

})

// GET a single workout

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

  res.json({mssg: 'GET a single workout'})

})

// POST a new workout

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

  res.json({mssg: 'POST a new workout'})

})

// DELETE a workout

router.delete('/:id', (req, res) => {

  res.json({mssg: 'DELETE a workout'})

})

// UPDATE a workout

router.patch('/:id', (req, res) => {

  res.json({mssg: 'UPDATE a workout'})

})

module.exports = router