|  |  |
| --- | --- |
| **Ex. No. 8** | **Building a REST API with Express, Node, and MongoDB** |
|  |

**CODE:**

**Server.js**

require('dotenv').config()

const express = require('express')

const app = express()

const mongoose = require('mongoose')

mongoose.connect('mongodb://localhost:27017/subscribers', { useNewUrlParser: true })

const db = mongoose.connection

db.on('error', (error) => console.error(error))

db.once('open', () => console.log('Connected to Database'))

app.use(express.json())

const subscribersRouter = require('./routes/subscribers')

app.use('/subscribers', subscribersRouter)

app.listen(3000, () => console.log('Server Started'))

**Subscribers.js**

const express = require('express')

const router = express.Router()

const Subscriber = require('../models/subscriber')

*// Getting all*

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

    try {

      const subscribers = await Subscriber.find()

      res.json(subscribers)

    } catch (err) {

      res.status(500).json({ message: err.message })

    }

  })

*// Getting One*

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

    res.json(res.subscriber)

  })

*// Creating one*

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

    const subscriber = new Subscriber({

      name: req.body.name,

      subscribedToChannel: req.body.subscribedToChannel

    })

    try {

      const newSubscriber = await subscriber.save()

      res.status(201).json(newSubscriber)

    } catch (err) {

      res.status(400).json({ message: err.message })

    }

  })

*// Updating One*

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

    if (req.body.name != null) {

      res.subscriber.name = req.body.name

    }

    if (req.body.subscribedToChannel != null) {

      res.subscriber.subscribedToChannel = req.body.subscribedToChannel

    }

    try {

      const updatedSubscriber = await res.subscriber.save()

      res.json(updatedSubscriber)

    } catch (err) {

      res.status(400).json({ message: err.message })

    }

  })

*// Deleting One*

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

    try {

      await res.subscriber.remove()

      res.json({ message: 'Deleted Subscriber' })

    } catch (err) {

      res.status(500).json({ message: err.message })

    }

  })

  async function getSubscriber(req, res, next) {

    let subscriber

    try {

      subscriber = await Subscriber.findById(req.params.id)

      if (subscriber == null) {

        return res.status(404).json({ message: 'Cannot find subscriber' })

      }

    } catch (err) {

      return res.status(500).json({ message: err.message })

    }

    res.subscriber = subscriber

    next()

  }

module.exports = router

**Subscriber.js**

const mongoose = require('mongoose')

const subscriberSchema = new mongoose.Schema({

    name: {

        type: String,

        required: true

    },

    subscribedToChannel: {

        type: String,

        required: true

    },

    subscribeDate: {

        type: Date,

        required: true,

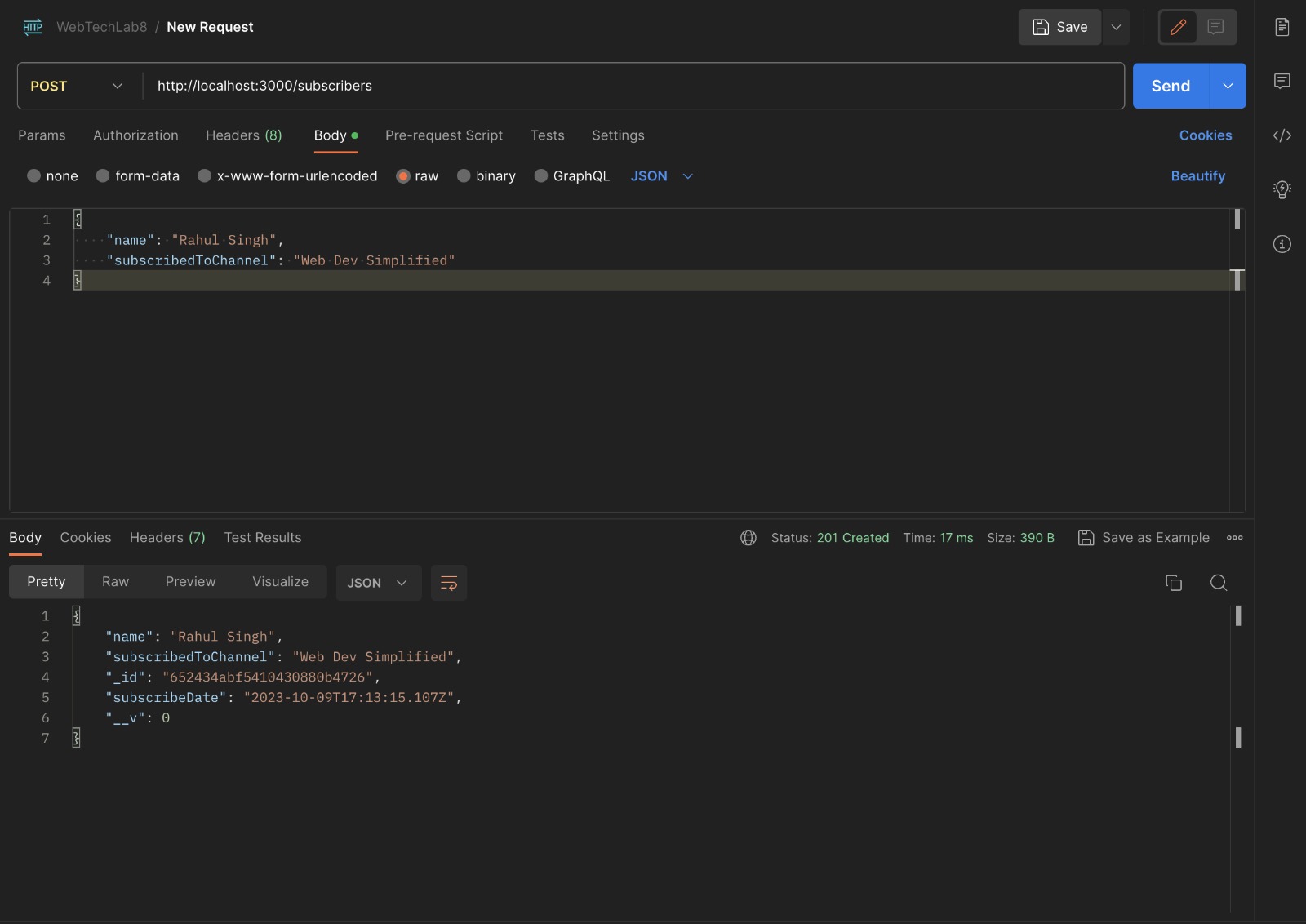
        default: Date.now

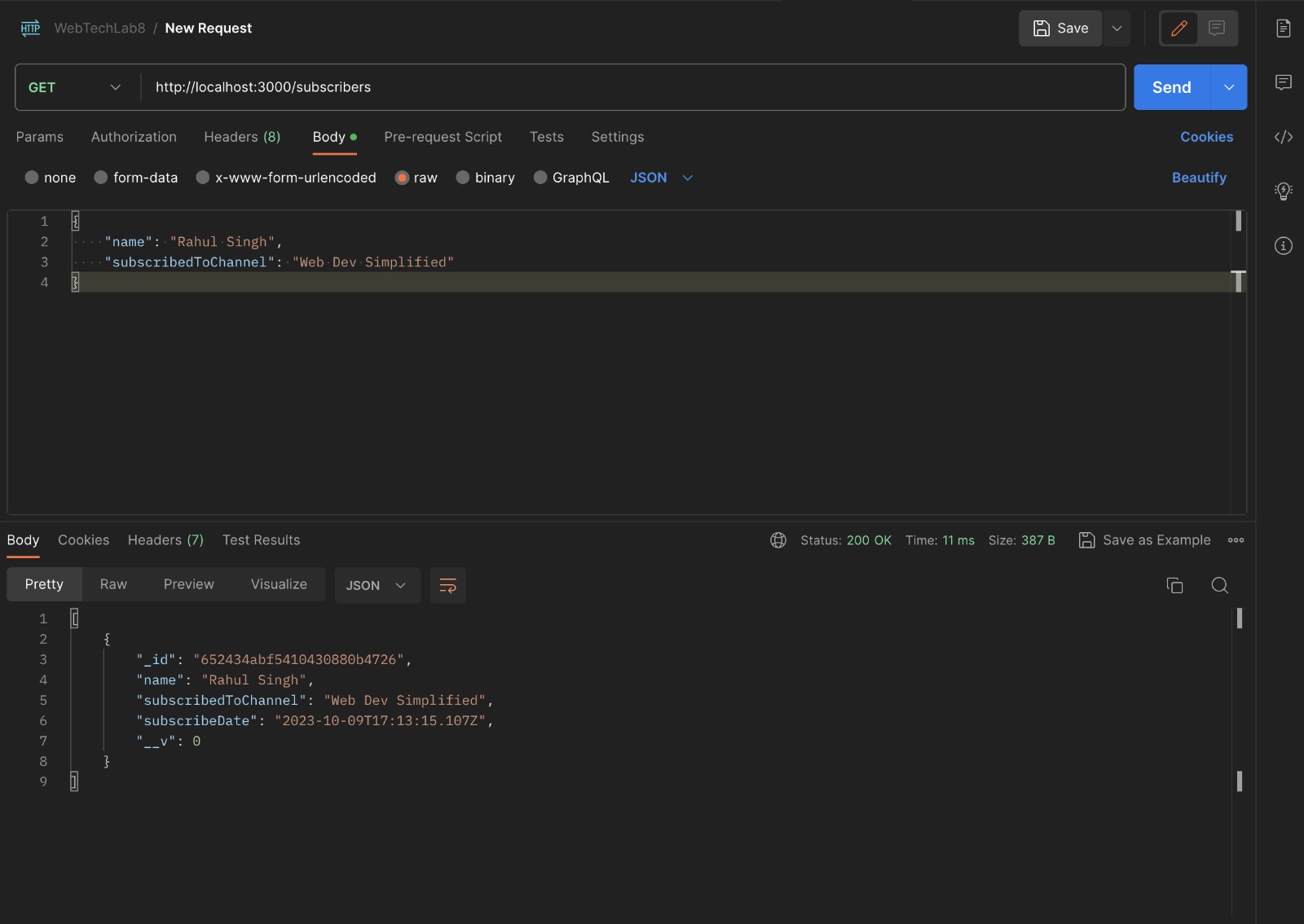
    }

})

module.exports = mongoose.model('Subscriber', subscriberSchema)

**OUTPUT:**

****

****