

Ex. No . 8	Building a REST API with Express, Node, and MongoDB
08/10/2023	

AIM:

To build a simple REST API using Express, Node and MongoDB.

PROGRAMS:

server.js

```
const express = require('express')
const app = express()
const mongoose = require('mongoose')

mongoose.connect("mongodb://0.0.0.0/students")
const db = mongoose.connection
db.on('error', (error) => console.error(error))
db.once('open', () => console.log('Connected to Database'))

app.use(express.json())

const studentsRouter = require('./routes/students')
app.use('/students', studentsRouter)

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

students.js

```
const express = require('express')
const router = express.Router()
const Student = require('../models/student')

//Getting all
router.get('/', async (req, res) => {
  try {
    const students = await Student.find()
    res.json(students)
  } catch (err) {
    res.status(500).json({message: err.message})
  }
})

//Getting one
router.get('/:id', getStudent, (req, res) => {
  res.json(res.student)
```

```

}))

//Creating one
router.post('/', async(req, res) => {
  const student = new Student({
    name: req.body.name,
    branch: req.body.branch,
    dept: req.body.dept
  })
  try{
    const newStudent = await student.save()
    res.status(201).json(newStudent)
  }catch(err){
    res.status(400).json({message: err.message})
  }
})

//Updating one
router.patch('/:id', getStudent, async(req, res) => {
  if (req.body.name !== null){
    res.student.name = req.body.name
  }
  if (req.body.branch !== null){
    res.student.branch = req.body.branch
  }
  if (req.body.dept !== null){
    res.student.dept = req.body.dept
  }
  try{
    const updatedStudent = await res.student.save()
    res.json(updatedStudent)
  }catch(err){
    res.status(400).json({message: err.message})
  }
})

//Deleting one
router.delete('/:id', getStudent, async(req, res) => {
  try{
    await res.student.deleteOne()
    res.json({message: "deleted student"})
  }catch(err){
    res.status(500).json({message: err.message})
  }
})

async function getStudent (req, res, next){
  let student

```

```

    try{
      student = await Student.findById(req.params.id)
      if(student == null){
        return res.status(404).json({message: 'Cannot find student'})
      }
    }catch(err){
      return res.status(500).json({message: err.message})
    }

    res.student = student
    next()
  }

module.exports = router

```

student.js

```

const mongoose=require('mongoose')

const studentsSchema = new mongoose.Schema({
  name : {
    type: String,
    required: true
  },
  branch:{
    type: String,
    required: true
  },
  dept:{
    type: String,
    required: true
  }
})

module.exports = mongoose.model('Student',studentsSchema)

```

route.http

```

GET http://localhost:3000/students

###
GET http://localhost:3000/students/6522ddee93ef971c351b20fb

###
POST http://localhost:3000/students
Content-Type: application/json

```

```

{
  "name" : "Tara",
  "branch" : "Iot",
  "dept" : "CSE"
}

###
DELETE http://localhost:3000/students/6522db57b2c124a433d6336a

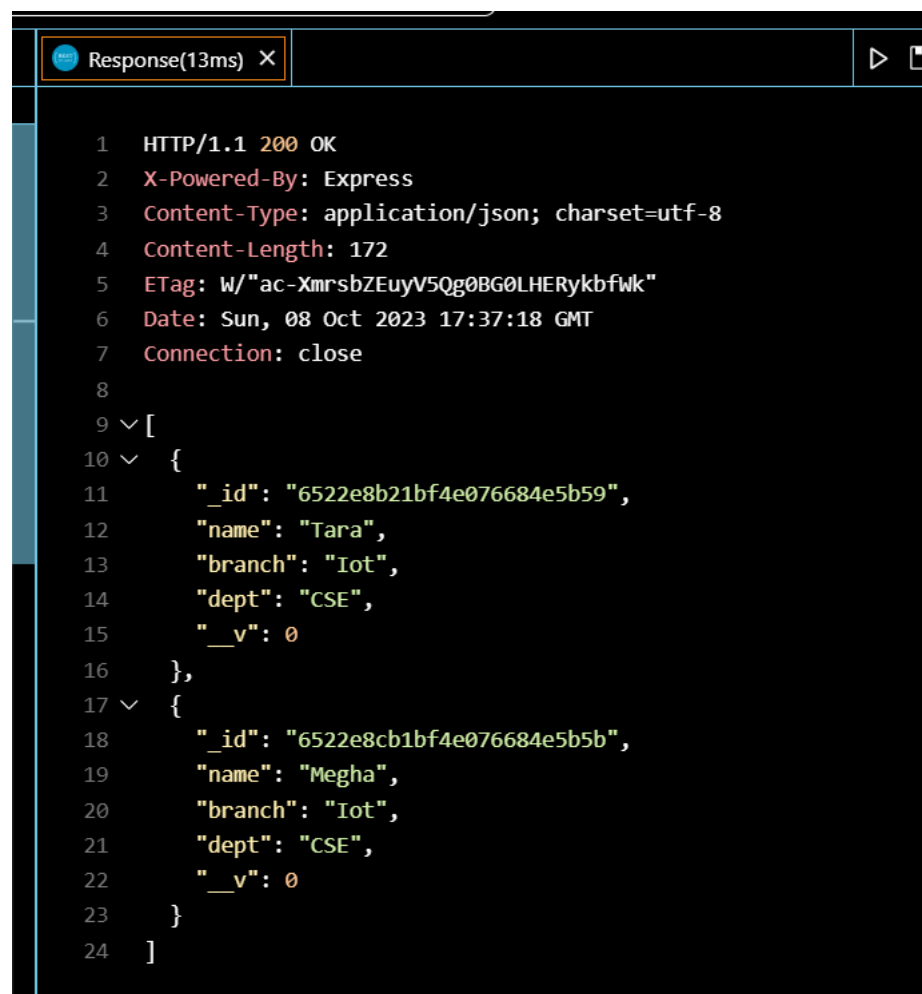
###
PATCH http://localhost:3000/students/6522dde93ef971c351b20fb
Content-Type: application/json

{
  "name": "Megha"
}

```

OUTPUT:

GET ALL: Get all students



```

Response(13ms) X
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 172
5 ETag: W/"ac-XmrsbZEuyV5Qg0BG0LHERykbFwk"
6 Date: Sun, 08 Oct 2023 17:37:18 GMT
7 Connection: close
8
9 ∨ [
10 ∨ {
11   "_id": "6522e8b21bf4e076684e5b59",
12   "name": "Tara",
13   "branch": "Iot",
14   "dept": "CSE",
15   "__v": 0
16 },
17 ∨ {
18   "_id": "6522e8cb1bf4e076684e5b5b",
19   "name": "Megha",
20   "branch": "Iot",
21   "dept": "CSE",
22   "__v": 0
23 }
24 ]

```

GET ONE: Get one student Tara

```
Response(13ms) X

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 84
5 ETag: W/"54-U0QAiSpLhdedsJd0Z3+g1IeMvs"
6 Date: Sun, 08 Oct 2023 17:40:32 GMT
7 Connection: close
8
9 {
10   "_id": "6522e8b21bf4e076684e5b59",
11   "name": "Tara",
12   "branch": "Iot",
13   "dept": "CSE",
14   "__v": 0
15 }
```

CREATE: creating new student Megha

```
Response(20ms) X

1 HTTP/1.1 201 Created
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 85
5 ETag: W/"55-9jjK1/6cM5LQdw78Mdn1kRoah1E"
6 Date: Sun, 08 Oct 2023 17:41:12 GMT
7 Connection: close
8
9 {
10   "name": "Megha",
11   "branch": "Iot",
12   "dept": "CSE",
13   "_id": "6522e9b81bf4e076684e5b61",
14   "__v": 0
15 }
```

UPDATE: name updated from Megha to Vikram

```
Response(25ms) X

1  HTTP/1.1 200 OK
2  X-Powered-By: Express
3  Content-Type: application/json; charset=utf-8
4  Content-Length: 86
5  ETag: W/"56-wfWbuG80LE//6nEUfum5x7hSo4I"
6  Date: Sun, 08 Oct 2023 17:41:58 GMT
7  Connection: close
8
9  {
10   "_id": "6522e9b81bf4e076684e5b61",
11   "name": "Vikram",
12   "branch": "Iot",
13   "dept": "CSE",
14   "__v": 0
15 }
```

DELETE: delete student vikram

```
Response(21ms) X

1  HTTP/1.1 200 OK
2  X-Powered-By: Express
3  Content-Type: application/json; charset=utf-8
4  Content-Length: 29
5  ETag: W/"1d-WzmVKiUqvFLpGm5l5Ccy62nLQIA"
6  Date: Sun, 08 Oct 2023 17:43:12 GMT
7  Connection: close
8
9  {
10   "message": "deleted student"
11 }
```

RESULT:

Thus, REST API was created successfully and tested with commands GET,POST,PATCH and DELETE.