

NAME : Rukmini Dubasi

Email : dubasirukmini@gmail.com

Phone No : 6305878824

College : Kakinada institute of technology and science, divili

Roll No : 20JQ1A0407

CREATING RESTFUL API USING EXPRESS.JS AND CREATING DATABASE AND INDEX USING MONGODB

SOURCE CODE:

Server.js:

```
require('dotenv').config();
const express = require('express'); const
mongoose = require('mongoose'); const
mongoData = process.env.DATABASE_URL;

mongoose.connect(mongoData); const
app = express();
database.on('error' (error) =>
{ console.log(error)
}); database.once('connected', ()
=> { console.log('Database
Connected')
}) const data =
mongoose.connection;

database.on('error', (error) => {
console.log(error)
}) database.once('connected', ()
=> {
```

```

        console.log('Database Connected');
    })

    app.use(express.json());
    const routes =
require('./routes')

app.use('/API');
    app.listen(3000, () => {
console.log(`Server is running on port
3000`)
    })

```

Route.js:

```

const express = require('express');
const router =
express.Router()
const Model = require('../models/model');
module.exports =
router;

//Post API router.post('/post',
(req, res) => {    const data =
new Model({        name:
req.body.name,        age:
req.body.age
    })    try {        const dataToSave =
await data.save();
res.status(200).json(dataToSave)
    }
    catch (error) {
res.status(400).json({message: error.message})
    }
})

//Get all API
router.get('/getAll', (req, res) => {
try{    const data = await
Model.find();    res.json(data)
    }    catch(error){
res.status(500).json({message: error.message})

```

```

    }
  })

  //Get by ID API
  router.get('/getOne/:id', async (req, res) => {
    try{
      const data = await
Model.findById(req.params.id);
      res.json(data)
    }
    catch(error){
res.status(500).json({message: error.message})
    }
  })

  //Update by ID API
  router.patch('/update/:id', (req, res) => {
    try {
      const id = req.params.id;
      const updatedData = req.body;
      const
options = { new: true };
      const result = await
Model.findByIdAndUpdate(
      id,
      updatedData, options
    )

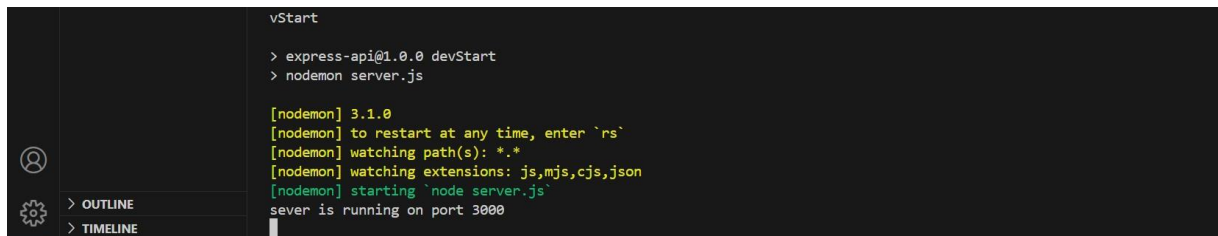
    res.send(result)
    } catch (error) {
res.status(400).json({ message: error.message })
    }
  })

  //Delete by ID API
  router.delete('/delete/:id', (req, res) => {
    try {
      const id = req.params.id;
      const data = await
Model.findByIdAndDelete(id)
      res.send(`Document with
${data.name} has been deleted..`)
    }
    catch (error) {
      res.status(400).json({ message: error.message })
    }
  })
})

```

Model.js:

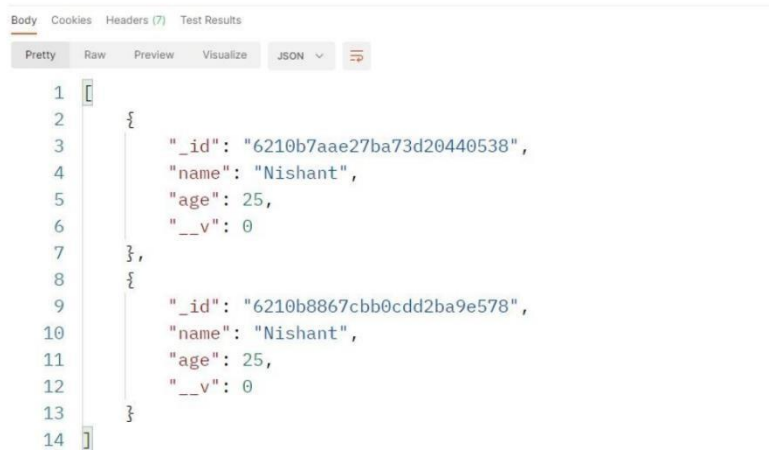
```
const mongoose = require('mongoose');
const dataSchema = new
mongoose.Schema({      name: {
required: true,          type: String
},      age: {
required: true,
type: Number
}
}) module.exports = mongoose.model('Data',
dataSchema)
```



```
vStart
> express-api@1.0.0 devStart
> nodemon server.js

[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
server is running on port 3000
```

post data to database:



Get data to database:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 [
2   {
3     "_id": "6210b7aae27ba73d20440538",
4     "name": "Nishant",
5     "age": 25,
6     "__v": 0
7   },
8   {
9     "_id": "6210b8867cbb0cdd2ba9e578",
10    "name": "Nishant",
11    "age": 25,
12    "__v": 0
13  }
14 ]
```

get data based on the ID:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "_id": "6210b7aae27ba73d20440538",
3   "name": "Nishant",
4   "age": 25,
5   "__v": 0
6 }
```

update data based on ID:

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   ... "name": "Nishant 9000",
3   ... "age": 25
4 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **JSON** ▾ ⌵

```
1 {
2   "_id": "6210b7aae27ba73d20440538",
3   "name": "Nishant 9000",
4   "age": 25,
5   "__v": 0
6 }
```

delete data based on ID:

DELETE ▾ localhost:3000/api/delete/6210c0b3d9e00e1e3d8bbf5a

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY
Key

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **HTML** ▾ ⌵

```
1 Document with Nishant has been deleted..
```