

## LAB EXERCISE 08 - REST API WITH EXPRESS, NODE & MongoDB

*Program:-*

Server.js

```
require('dotenv').config()
const express = require('express')
const app = express()
const mongoose = require('mongoose')

mongoose.connect(process.env.DATABASE_URL, { 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'))
```

Subscriber.js – Routes

```
const express = require('express')
const router = express.Router()
const Subscriber = require('../models/subscriber')
```

```
const subscribers = require('../models/subscribers')
```

```
// Getting all
```

```
router.get('/', async(req, res) => {  
  try {  
    const subscriber = await Subscriber.find()  
    res.json(subscribers)  
  } catch (error) {  
    res.status(500).json({message: error.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 subscribersave()  
  res.status(201).json(newSubscriber)  
} catch (err) {  
  res.status(400).json({message: err.message })  
}
```

// Updating One

```
router.patch('/', getSubscriber, async (req,res) => {  
  if (req.body.name != null) {  
    res.subscriber.name = req.body.name  
  }  
  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()  
  
  }catch (err) {  
    res.status(500).json({message: err.message})  
  }  
})
```

function getSubscriber (req,res,next) {

```
  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
}

module.exports = router

```

## Subscribers.js – Model

```

const mongoose = require('mongoose')

const subscriberSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true

  },
  subscriberToChannel: {

  },

  subsscribeDate: {
    type: Date,

```

```
    required: true,  
    default: Date.now  
  }  
})
```

```
module.exports = mongoose.model('Subscribe', subscriberSchema)
```

```
Route.rest
```

```
GET http://localhost:300/subscribers
```

```
###
```

```
GET http://localhost:300/subscribers/15
```

```
###
```

```
POST http://localhost3000/subscribers
```

```
{  
  "name" : "Amazing Person",  
  "subscribedtoChannel" : "Web Dev Simplified"  
}
```

```
###
```

```
DELETE http://localhost3000/subscribers
```

```
###
```

```
PATCH http://localhost3000/subscribers
```

Content-Type: application/json

```
{  
  "name": "New Name"  
}
```