https://shorturl.at/kAHNV

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sequelize**

**Mongoose**

**Project impl**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

==============================================================

Sequelize

==============================================================

- it is a promise based Node.JS ORM(Object relational Mapping).

- it is used to interact with Postgres, MySQL, MariaDB, SQLite

and Microsoft SQL Server.

- It features solid transaction support, relational, egar and

Lazy Loading, read applications and more.

- A great thing about sequelize is no need to take care of

underlying database.

- We can easily switch databases by adjusting the configuration

file, and the code remains the same.

1. Create a file server.js

2. >npm init -y

3. install following modules

express

sequelize

sequelize-cli

mysql2

>yarn add express sequelize@6.31.0 sequelize-cli@6.6.0 mysql2@3.2.2 --save

4. Initialyse sequelize

>npx sequelize init

5. Currently the folders 'migrations' and 'seeders' are not required,

so we can delete those.

6. Create database 'sqldem'

7. Update config.json with password and name of database in development.

<>

config

-config.json

"development": {

"username": "root",

"password": "root",

"database": "sqldem",

"host": "127.0.0.1",

"dialect": "mysql"

},

<>

config

- config.json

models

- todo.js

routes

- apiRoutes.js

fetch

- fetch.js

insert

- insert.js

update

- update.js

delete

- delete.js

- server.js

8. Create server.js for testing with sequelize

\*\*\*server.js\*\*\*

//import express module

const express = require('express')

//create rest object

const app = express()

//import db

const db = require("./models")

//set JSON as MIME type

app.use(express.json())

//client parameters are encoded as JSON

app.use(express.urlencoded({ extended: false }))

//create port

const port = process.env.PORT || 8080

//syncronize with sequelize and assign port no

db.sequelize.sync().then(() => {

app.listen(port, () => {

console.log('Server listening port no ' + port)

})

}, (errMsg) => {

console.log(errMsg)

})

9. test this server as

>node server

<>

models

- todo.js

\*\*\*todo.js\*\*\*

module.exports = (sequelize, DataType) => {

const Todo = sequelize.define("product", {

p\_id: {

type: 'int',

allowNull: false

},

p\_name: {

type: 'varchar(20)',

allowNull: false

},

p\_cost: {

type: 'int',

allowNull: false

},

})

return Todo

}

/\*

where define function has two arguments

- First argument is name of table

- name of the table -> 's' auto appended with the name

- i.e. 'product' table will be as 'products'

- Second argument is json of fields

- key is name of field value is json of type.

- in json of type key is type value of data.

\*/

<>

routes

fetch

- fetch.js

\*\*\*fetch.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import db

const db = require('../../models')

//get all records

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

db.product.findAll().then(result => {

res.send(result)

})

})

//get single record by id

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

db.product.findAll({

where: {

id: req.params.id

}

}).then(result => {

if (result.length == 0)

res.send('Record not found')

else

res.send(result)

})

})

//export router

module.exports = router

<>

routes

insert

- insert.js

\*\*\*insert.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import db

const db = require("../../models")

//insert a new record

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

db.product.create(req.body).then(submitted => {

res.send(submitted)

})

})

//export router

module.exports = router

<>

routes

update

- update.js

\*\*\*update.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import db

const db = require("../../models")

//update a record

router.put("/", (req, res) => {

let obj = {

p\_id: req.body.p\_id,

p\_name: req.body.p\_name,

p\_cost: req.body.p\_cost

}

db.product.update(obj, { where: { id: req.body.id } })

.then((result) => {

if (result == 0)

res.send({ 'record': 'not found' })

else

res.send({ 'record': 'upadated' })

})

})

//export router

module.exports = router

<>

routes

delete

- delete.js

\*\*\*delete.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import db

const db = require("../../models")

//update a record

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

db.product.destroy({ where: { id: req.params.id } })

.then((result) => {

if (result == 0)

res.send({ 'record': 'not found' })

else

res.send({ 'record': 'deleted' })

})

})

//export router

module.exports = router

<>

routes

- apiRoutes.js

\*\*\*apiRoutes.js\*\*\*

//create router instance

let router = require('express').Router()

router.use('/fetch', require('./fetch/fetch'))

router.use('/insert', require('./insert/insert'))

router.use("/update", require('./update/update'))

router.use('/delete', require('./delete/delete'))

//export router

module.exports = router

\*\*\*server.js\*\*\*

//import express module

const express = require('express')

//create rest object

const app = express()

//import db

const db = require("./models")

//set JSON as MIME type

app.use(express.json())

//client parameters are encoded as JSON

app.use(express.urlencoded({ extended: false }))

//create port

const port = process.env.PORT || 8080

/////////////////////////////////////

//import apiRoutes

const apiRoutes = require("./routes/apiRoutes")

//use apiRoutes

app.use("/", apiRoutes)

/////////////////////////////////////

//syncronize with sequelize and assign port no

db.sequelize.sync().then(() => {

app.listen(port, () => {

console.log('Server listening port no ' + port)

})

}, (errMsg) => {

console.log(errMsg)

})

/\*

Test URLs with postman

GET

http://localhost:8080/fetch/all

http://localhost:8080/fetch/find/2

POST

http://localhost:8080/insert

PUT

http://localhost:8080/update

DELETE

http://localhost:8080/delete/2

\*/

=================================================

Mongoose

=================================================

Create a folder MongooseEg

drag to vscode

create server.js

initialise application

>npm init -y

install express body-parser cors mongoose

>yarn add express body-parser cors mongoose --save

Directory Structure

<>

apis

- productApis.js

model

- Product.js => STRICTLY FIRST LETTER CAPITAL AND NO 'S' AT END

routes

- productRoutes.js

- url.js

- server.js

create url.js file and store mongodb url here

login to mongodb atlas

click on databases -> left side panel

click on connect

choose Compass

copy url

-> replace <password> with admin

-> append nodedb in url

\*\*\*url.js\*\*\*

module.exports = "mongodb+srv://admin:admin@mdb.vtkja.mongodb.net/nodedb"

import url in server.js

connect to mongodb as

mongoose.connect

create server.js and test server for mongodb connection

\*\*\*server.js\*\*\*

//import modules

const express = require('express')

let bodyparser = require('body-parser')

let cors = require('cors')

let mongoose = require('mongoose')

//import url

let url = require('./url')

//creater rest object

let app = express()

//set JSON as MIME type

app.use(bodyparser.json())

//client is not sending form data -> encoding JSON

app.use(bodyparser.urlencoded({ extended: false }))

//enable CORS -> Cross Origine Resource Sharing -> communication among various ports

app.use(cors())

////////////////////////////////////////////////

//connect to mongodb

mongoose.connect(url).then(()=>{

console.log('Connection success')

},()=>{

console.log('Connection Failed')

})

////////////////////////////////////////////////

//create port

let port = 8080

//assign port no

app.listen(port, () => {

console.log('Server listening port no :- ' + port)

})

Now create mongoose schema as

model/Product.js

\*\*\*Product.js\*\*\*

//import mongoose

const mongoose = require('mongoose')

//create schema

const productSchema = new mongoose.Schema({

p\_id: Number,

p\_name: String,

p\_cost: Number

})

module.exports = mongoose.model("Product", productSchema)

Create apis

apis/productApis.js

\*\*\*productApis.js\*\*\*

//import db schema

const Product = require("../model/Product")

//get all products

const products\_all = async (req, res) => {

try {

const products = await Product.find()

res.status(200).send(products)

console.log('Data Sent')

}

catch (error) {

res.status(400).send({ 'message': error })

}

}

module.exports = {

products\_all

}

Define routes in

routes/productRoutes.js

\*\*\*productRoutes.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import productApi

const productApi = require('../apis/productApis')

//fetch all records

router.get("/fetch", productApi.products\_all)

//export router

module.exports = router

import and use routes in server.js and test

fetch api in postman

http://localhost:8080/fetch

after that perform remaining

\*\*\*server.js\*\*\*

//import modules

const express = require('express')

let bodyparser = require('body-parser')

let cors = require('cors')

let mongoose = require('mongoose')

//import url

let url = require('./url')

//creater rest object

let app = express()

//set JSON as MIME type

app.use(bodyparser.json())

//client is not sending form data -> encoding JSON

app.use(bodyparser.urlencoded({ extended: false }))

//enable CORS -> Cross Origine Resource Sharing -> communication among various ports

app.use(cors())

////////////////////////////////////////////////

//connect to mongodb

mongoose.connect(url).then(() => {

console.log('Connection success')

}, () => {

console.log('Connection Failed')

})

////////////////////////////////////////////////

//=================================================//

//import routes

const productRoutes = require('./routes/productRoutes')

//use routes

app.use("/", productRoutes)

//=================================================//

//create port

let port = 8080

//assign port no

app.listen(port, () => {

console.log('Server listening port no :- ' + port)

})

>>>>>>>>>>>>>>>>

now update remaining files

\*\*\*productApis.js\*\*\*

//import db schema

const Product = require("../model/Product")

//get all products

const products\_all = async (req, res) => {

try {

const products = await Product.find()

res.status(200).send(products)

console.log('Data Sent')

}

catch (error) {

res.status(400).send({ 'message': error })

}

}

//insert a product

const insert\_product = async (req, res) => {

const product = new Product({

p\_id: req.body.p\_id,

p\_name: req.body.p\_name,

p\_cost: req.body.p\_cost

})

try {

const savedProduct = await product.save()

console.log('Product Inserted')

res.status(200).send(savedProduct)

}

catch (error) {

res.status(400).send(error)

}

}

//update product

const update\_product = async (req, res) => {

let p\_id = req.body.p\_id

const product = {

p\_name: req.body.p\_name,

p\_cost: req.body.p\_cost

}

try {

const updatedProduct = await Product.updateOne(

{ p\_id }, product

)

if (updatedProduct.modifiedCount != 0) {

console.log('Product Updated', updatedProduct)

res.status(200).send({ 'update': 'success' })

}

else {

console.log('Product Not Updated')

res.status(200).send({ 'update': 'failed' })

}

}

catch (error) {

res.status(400).send(error)

}

}

//delete product

const delete\_product = async (req, res) => {

let p\_id = req.body.p\_id

try {

const deletedProduct = await Product.deleteOne(

{ p\_id: p\_id }

)

if (deletedProduct.deletedCount != 0) {

console.log('Product Deleted')

res.status(200).send({ 'delete': 'success' })

}

else {

console.log('Product Not Deleted')

res.status(200).send({ 'delete': 'failed' })

}

} catch (error) {

res.status(400).send(error)

}

}

module.exports = {

products\_all,

insert\_product,

update\_product,

delete\_product

}

\*\*\*prodctRoutes.js\*\*\*

//import express module

const express = require('express')

//create router instance

const router = express.Router()

//import productApi

const productApi = require('../apis/productApis')

//fetch all records

router.get("/fetch", productApi.products\_all)

//insert a record

router.post("/insert", productApi.insert\_product)

//update a record

router.put("/update", productApi.update\_product)

//delete a record

router.delete("/delete", productApi.delete\_product)

//export router

module.exports = router

Host this on render and cyclic

================================================================

================================================================