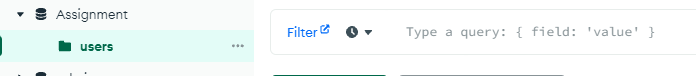
Part 1.

Develop the MongoDB database to store documents for different users who may login to the system.

MongoDb Database/Collection:



NodeJs Application:

Server.js – Application Index

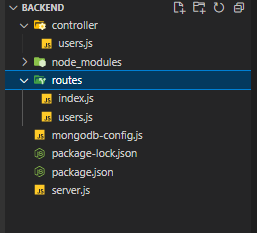
routes/index.js – call all router

routes/users.js – router for user

controller – directory for all contollers

controller/users.js – controller for user

mongodb-config.js – connection to mongo database and collection



server.js:

const express = require('express')

const bodyParser = require('body-parser')

const app = express()

const main = async () => {

  app.use(bodyParser.json())

  // all routers is called here

  app.use(require('./routes/index'))

  // default not found response

  app.use((req, res) => {

    res.status(404).send('Resource not found')

  })

  // default error response

  app.use((err, req, res, next) => {

    return res

      .status(500)

      .json({ statusCode: 422, message: err.message, stack: err.stack })

  })

  // application is running in localhost port 8000

  app.listen(8000, () => {

    console.info(`Server is running on Port:8000`)

  })

  return app

}

module.exports = main()

router/index.js

const express = require('express')

const router = express.Router()

const userRouter = require('./users')

router.use('/user', userRouter)

module.exports = router

router/users.js

const express = require("express")

const userRouter = express.Router()

const Users = require("../controller/users")

// assign controller to routes

userRouter.post("/login", Users.login)

userRouter.post("/register", Users.register)

module.exports = userRouter;

controller/users.js

const mongodb = require("../mongodb-config")

let Users = {}

Users.login = async (req, res) => {

  try {

    // call mongodb

    const db = await mongodb();

    // destructure data from request body

    const { username, password } = req.body

    // query data

    const [user] = await db.find({ username: username, password: password}).toArray();

    if (!user) {

      throw {message: "user not found!"}

    }

    delete user.password

    res.status(200).json({statusCode: 200, message: "successfully login!", data: user});

  } catch (error) {

    res.status(200).json({statusCode: 422, message: error.message});

  }

}

Users.register = async (req, res) => {

  try {

    // call mongodb

    const db = await mongodb();

    // destructure data from request body

    const {username, password, first\_name, last\_name, age, gender, birth\_date} = req. body;

    let collectionDataRegister = {

      username,

      password,

      first\_name,

      last\_name,

      age,

      gender,

      birth\_date

    }

    // query

    await db.insertOne(collectionDataRegister);

    res.status(200).json({statusCode: 200, message: "successfully registered!"});

  } catch (error) {

    res.status(200).json({statusCode: 422, message: error.message});

  }

}

module.exports = Users

mongodb-config.js

const { MongoClient } = require('mongodb');

// connection in mongodb server

const url = 'mongodb://localhost:27017';

// database name

const dbName = 'Assignment';

const initMongoose = async () => {

  const client = new MongoClient(url)

  await client.connect()

  const db = client.db(dbName)

  // collection name

  const collection = db.collection('users')

  return collection

}

module.exports = initMongoose

TEST DATA:

Register:

url: localhost:8000/user/register

Body raw

{

    "username": "test",

    "password": "test\_pass",

    "first\_name": "test\_firstname",

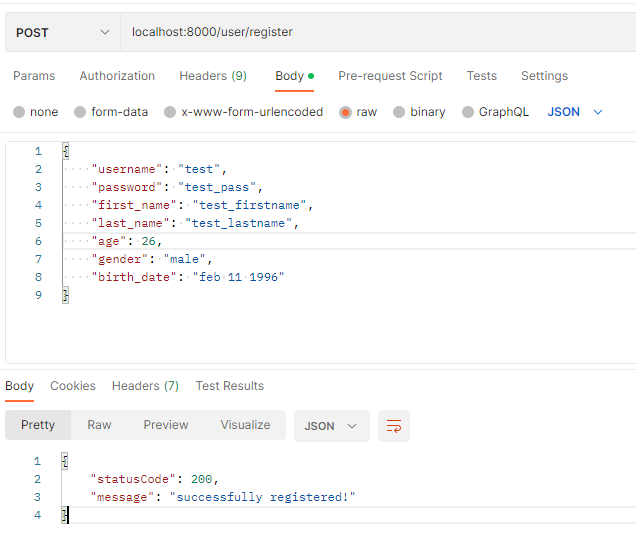
    "last\_name": "test\_lastname",

    "age": 26,

    "gender": "male",

    "birth\_date": "feb 11 1996"

}



LOGIN:

url: localhost:8000/user/login

Body raw

{

    "username": "test",

    "password": "test\_pass"

}

