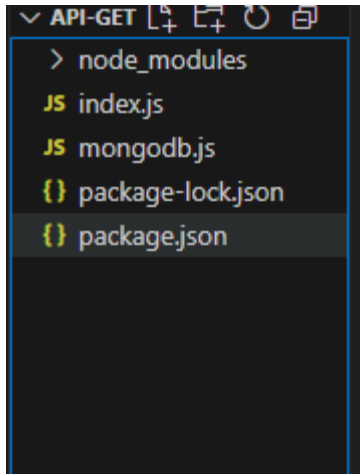


- MONGO && EXPRESS

File –structure



in mongo db

```
const { MongoClient } = require('mongodb');
// or as an es module:
// import { MongoClient } from 'mongodb'

// Connection URL
const url = 'mongodb://localhost:27017';
const client = new MongoClient(url);

// Database Name
const dbName = 'employee';

async function dbConnect() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  return db
}

module.exports = dbConnect
```

index.js

```
import { createRequire } from 'module';
const require = createRequire(import.meta.url);
import multer from 'multer';
const express = require('express');
const dbConnect = require('./mongodb');
const app = express()
const upload = multer();
let db = await dbConnect()
app.use(express.json())

const collection = db.collection('positions');

app.get('/', async (req, res) => {

    const findResult = await collection.find({}).toArray();
    res.send(findResult)
})
app.post('/add', upload.none(), async (req, res) => {
    try {
        const { name, post } = req.body;
        console.log(req.body)
        if (!name || !post) {
            return res.status(400).send({ error: 'Name and Post are required
fields' });
        }

        const obj = { name, post };
        console.log('Form Data:', obj);

        const result = await collection.insertOne(obj);
        res.status(201).send(result);
    } catch (err) {
        console.error('Insert error:', err);
        res.status(500).send({ error: 'Internal Server Error' });
    }
})

app.listen(3000)
```

UPDATE & DELETE

```
import { createRequire } from 'module';
import { ObjectId } from 'mongodb';
const require = createRequire(import.meta.url);
import multer from 'multer';
const express = require('express');
const dbConnect = require('./mongodb');
const app = express();
const upload = multer();
let db = await dbConnect();
app.use(express.json());

const collection = db.collection('positions');

app.put('/update', upload.none(), async (req, res) => {
  try {
    const { id, post } = req.body;
    console.log(req.body);

    const filter = { _id: new ObjectId(id) };
    const updateDoc = { $set: { post } };

    const result = await collection.updateOne(filter, updateDoc);

    if (result.matchedCount === 0) {
      return res.status(404).send({ error: 'Document not found' });
    }

    res.status(200).send({ message: 'Document updated successfully', result });
  } catch (err) {
    console.error('Update error:', err);
    res.status(500).send({ error: 'Internal Server Error' });
  }
});
```

```
app.delete('/delete', upload.none(), async (req, res) => {
  try {
    const { id } = req.body;
    console.log(req.body);

    const filter = { _id: new ObjectId(id) };
    const result = await collection.deleteOne(filter);

    if (result.deletedCount === 0) {
      return res.status(404).send({ error: 'Document not found' });
    }

    res.status(200).send({ message: 'Document deleted successfully', result
  });
} catch (err) {
  console.error('Delete error:', err);
  res.status(500).send({ error: 'Internal Server Error' });
}
});

app.listen(3000)
```