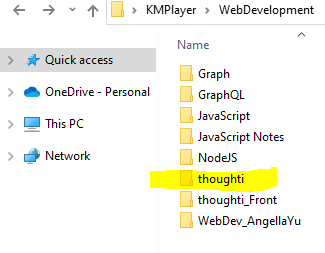
The Application Programming Interface (API) has been created using Nodejs as the server and ExpressJs as the Middle ware.

To Launch and test the API , follow the instructions below:

1. Create a folder with the name thoughti :



1. Open VS Code editor , and open the thoughti folder.
2. Open the Terminal in VS code and type : npm init –y.

This would create a package.json file.

1. Then type npm i express in the terminal to install express.
2. To make the process smoother , install nodemon after you have created a dev dependency.

Type npm i –save-dev nodemon.

1. Create 2 new files server.js and routes.js.
2. Install REST CLIENT from VS Code extensions.

Copy and paste the codes:

In server.js :

const express=require('express');

const app=express();

const routeMethod=require('./routes');

//To prevent Cross Origin Resource Sharing Error.

app.use((req, res, next) => {

res.setHeader('Access-Control-Allow-Origin', '\*');

res.setHeader(

'Access-Control-Allow-Headers',

'Origin, X-Requested-With, Content-Type, Accept, Authorization'

);

res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PATCH, DELETE');

next();

});

app.use('/api/managed-records',routeMethod);

app.listen(3000,()=>{

console.log("The server started on port 3000");

});

In routes.js :

const express=require('express');

const router=express.Router();

const users=require('./data.js');

const data=users.data;

router.get('/records',(req,res)=>{

const page=parseInt(req.query.page);

const limit=req.query.limit;

const startIndex=(page-1)\*limit;

const endIndex=page\*limit;

const results={};

if(endIndex<data.length){

results.next={

page:page+1

}

}

if(startIndex>0){

results.previous={

page:page-1,

}

}

results.results=data.slice(startIndex,endIndex)

res.json(results);

})

module.exports=router;

Now create a file request.rest which could be read by the REST CLIENT Extention.

Type:

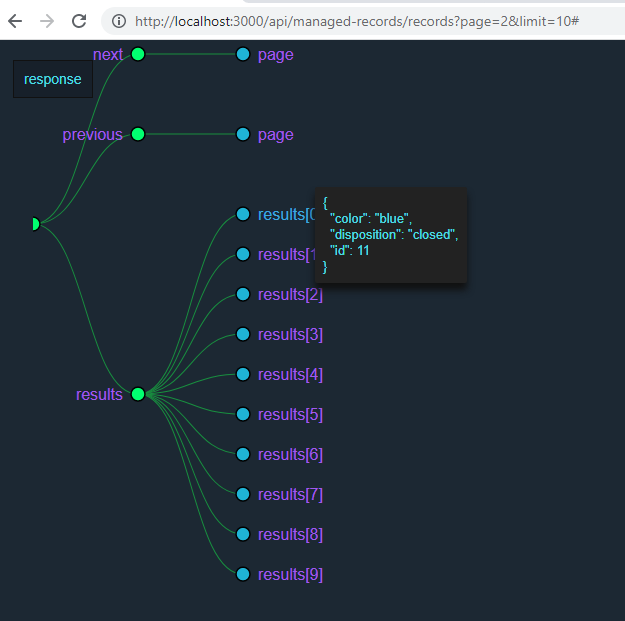
<http://localhost:3000/api/managed-records/records?page=10&limit=10>

In the above URL , you have to specify the page number and the number of items you would want to have in the page.

You get the output as :



The above thing could also be tested in the browser :

(JSON Viewer pro has been used):

A bird’s eye view of the project would look like:

