//using middleware and handler to create http server

import 'dotenv/config';

import  http  from 'http';

const users =[

    {id:1,user:'basanta'},

    {id:2,user:'2basanta'},

    {id:3,user:'3basanta'},

]

// Logger middleware

const logger = ( req, res, next) =>{

    console.log(` ${ req.url } , ${ req.method } `);

    next(); // Pass control to the next middleware or route handler

}

// json ContentType middleware

const jsonContentTypeMIddleWare = ( req, res, next) =>{

    res.writeHead(200, { 'Content-Type': 'application/json' } );

    next();

}

//handle all user

const getAllUsersHandler = ( req, res ) =>{

    res.write(JSON.stringify(users));

    res.end();

}

//handle  user By Id

const getUserHandlerById = (req, res ) =>{

    const urlId = parseInt(req.url.split('/')[3]);

    const user = users.find( (user) => user.id === urlId);

    if(user){

        res.write(JSON.stringify( {id:1,name:"dynamic data"} ));

    }else{

        res.write(JSON.stringify({message:'no such user found'}));

    }

    res.end();

}

//no Page Found Handler

const noPageFoundHandler = ( req, res ) =>{

    res.write(JSON.stringify({message:'No page  found'}));

    res.end();

}

const server = http.createServer( async (req,res)=>{

    logger( req, res, ()=>{

        if ( req.method === 'GET' && req.url === '/api/users' ){

            jsonContentTypeMIddleWare(req,res, ()=>{

                getAllUsersHandler(req, res);

            });

        } else if ( req.method === 'GET' && req.url.match(/\/api\/users\/([0-9]+)/) ) {

            getUserHandlerById(req,res);

        } else{

            noPageFoundHandler(req, res);

        }

    });

});

const PORT = process.env.PORT;

server.listen(PORT, () =>{

    console.log(`port running in ${PORT}`);

})

//proper and handle post

// pusing POST json data to aray

{"id":4,"user":"3basanta"}//json data

import 'dotenv/config';

import  http  from 'http';

const users =[

    {id:1,user:'basanta'},

    {id:2,user:'2basanta'},

    {id:3,user:'3basanta'},

]

// Logger middleware

const logger = ( req, res, next) =>{

    console.log(` ${ req.url } , ${ req.method } `);

    next(); // Pass control to the next middleware or route handler

}

// json ContentType middleware

const jsonContentTypeMIddleWare = ( req, res, next) =>{

    res.writeHead(200, { 'Content-Type': 'application/json' } );

    next();

}

//handle all user

const getAllUsersHandler = ( req, res ) =>{

    res.write(JSON.stringify(users));

    res.end();

}

//post user handler

//for now in array not in database

const postUserHandler = (req, res) =>{

    let body ='';

    //lisen for data whihch comes in

    //data event and end event

    req.on('data', (chunk) =>{

        body += chunk;

        body.toString();

    })

    req.on('end', () =>{

        //converting into js string and pushing to array

        const data = JSON.parse(body);

        users.push(data);

        //for display

        res.statusCode = 201 ;

        res.write(JSON.stringify(data));

        res.end();

    })

}

//handle  user By Id

const getUserHandlerById = (req, res ) =>{

    const urlId = parseInt(req.url.split('/')[3]);

    const user = users.find( (user) => user.id === urlId);

    if(user){

        res.write(JSON.stringify( {id:1,name:"dynamic data"} ));

    }else{

        res.write(JSON.stringify({message:'no such user found'}));

    }

    res.end();

}

//no Page Found Handler

const noPageFoundHandler = ( req, res ) =>{

    res.write(JSON.stringify({message:'No page  found'}));

    res.end();

}

const server = http.createServer( async (req,res)=>{

    logger( req, res, ()=>{

        if ( req.method === 'GET' && req.url === '/api/users' ){

            jsonContentTypeMIddleWare(req,res, ()=>{

                getAllUsersHandler(req, res);

            });

        } else if ( req.method === 'GET' && req.url.match(/\/api\/users\/([0-9]+)/) ) {

            getUserHandlerById(req,res);

        }else if ( req.method === 'POST' && req.url === '/api/users' ){

            postUserHandler(req,res);

        }else {

            noPageFoundHandler(req, res);

        }

    });

});

const PORT = process.env.PORT;

server.listen(PORT, () =>{

    console.log(`port running in ${PORT}`);

})