Develop a backend server using Express JS and create at least 3 endpoints or routes.

Code:-

const express=require('express');

const users=require('./Users');

const uuid=require('uuid');

const path=require('path');

const app=express();

const idFilter = req => member => member.id === parseInt(req.params.id);

//app.use('/api/users',require('./routes/api/members'));

// Body Parser Middleware

app.use(express.json());

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

app.use(express.static(path.join(\_\_dirname,'public')));

//GET All USERS

app.get('/api/users',(req,res)=>res.json(users));

//GET Specific USER Based on ID

app.get('/api/users/:id', (req, res) => {

const found = users.some(idFilter(req));

if (found) {

res.json(users.filter(idFilter(req)));

} else {

res.status(400).json({ msg:' No member with the id of ${req.params.id}' });

}

});

//CREATE A NEW USER

app.post('/api/users',(req,res)=>{

const newMember={

id: uuid.v4(),

name: req.body.name,

email: req.body.email,

status: 'guest'

};

if(!newMember.name){

return res.status(400).json({msg:'NAME Must be provided'});

}

users.push(newMember);

res.json(users);

}

);

//DELETE Specific USER Based on ID

app.delete('/api/users/:id', (req, res) => {

const found = users.some(idFilter(req));

if (found) {

res.json({msg:'Deleted',

members:users.filter(

member=>member.id!==parseInt(req.params.id))})

} else {

res.status(400).json({ msg: 'No member with the id of ${req.params.id}' });

}

});

//UPDATE Specific USER Based on ID

app.put('/api/users/:id',(req,res)=>

{

const found = users.some(member=>member.id===parseInt(req.params.id));

if(found)

{

const updMember=req.body;

users.forEach(

member=>{

if(member.id===parseInt(req.params.id))

{

member.name=updMember ? updMember.name : member.name;

member.email=updMember.email ? updMember.email : member.email;

res.json({msg:'Updated Details',member})

}

}

);

}

else{

res.status(400).json({msg:'No User found with ${req.params.id}'});

}

});

const PORT=3001;

app.listen(PORT, () => console.log('Server is Running ${PORT}'));

output:-

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated