const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

app.use(cors());

app.use(bodyParser.json());

let clients = [];

let messages = [];

// ส่งข้อความใหม่ให้ทุก client

function sendToClients(message) {

clients.forEach((client) => client.res.write(`data: ${JSON.stringify(message)}\n\n`));

}

// Endpoint สำหรับ SSE

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

res.setHeader('Content-Type', 'text/event-stream');

res.setHeader('Cache-Control', 'no-cache');

res.setHeader('Connection', 'keep-alive');

clients.push({ id: Date.now(), res });

// ส่งข้อความที่มีอยู่ทั้งหมดให้ client ใหม่

res.write(`data: ${JSON.stringify(messages)}\n\n`);

req.on('close', () => {

clients = clients.filter((client) => client.res !== res);

});

});

// Endpoint สำหรับเพิ่มข้อความ

app.post('/add-message', (req, res) => {

const { message, name } = req.body;

if (!message || !name) {

return res.status(400).json({ error: 'กรุณาใส่ชื่อและข้อความ' });

}

const newMessage = { id: Date.now(), message, name };

messages.push(newMessage);

sendToClients(newMessage); // ส่งข้อความใหม่ให้ทุก client

res.status(201).json({ success: true });

});

//Endpoint สำหรับดูข้อความ

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

res.json(messages); // ส่งข้อความทั้งหมดกลับในรูปแบบ JSON

});

//Endpoint สำหรับลบข้อความ

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

const messageId = parseInt(req.params.id, 10); // แปลง ID ที่ส่งมาเป็นตัวเลข

const index = messages.findIndex(msg => msg.id === messageId);

if (index !== -1) {

messages.splice(index, 1); // ลบข้อความออกจาก array

res.status(200).send('ข้อความถูกลบแล้ว');

} else {

res.status(404).send('ไม่พบข้อความ');

}

});

// เริ่มเซิร์ฟเวอร์

const PORT = 3000;

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));