Node.js - Day 5: HTTP Module → Build Server + Routing System

■ Learning Objectives

- Understand what the HTTP module is and how it works.
- Create a web server using Node.js.
- Implement basic routing for different URLs.
- Serve HTML and JSON responses dynamically.

■ Session 1 (45 min) — Understanding the HTTP Module

1. What is the HTTP Module?
The HTTP module allows Node.js to create web servers and handle client requests.
It uses TCP protocol to communicate between browser and server.
const http = require('http');

2. Create a Basic HTTP Server
const http = require('http'); const server = http.createServer((req, res) => { res.writeHead(200, {'Content-Type': 'text/plain'}); res.end('Hello from Node.js Server!'); }); server.listen(3000, () => { console.log('Server running at http://localhost:3000'); }); Visit http://localhost:3000 to test.

■ 3. Explanation

- http.createServer() — creates the server.
- (req, res) — callback that handles each request.
- res.writeHead() — defines status code and headers.
- res.end() — ends the response.
- server.listen(3000) — starts the server on port 3000.

4. Request & Response Objects
const server = http.createServer((req, res) => { console.log(req.url, req.method); res.end('Logging request details...'); });

■ Mini Exercise

Create a server that logs every request URL and displays 'Welcome to my Node.js server!'.

■ Session 2 (45 min) — Building a Routing System

1. What is Routing?
Routing defines how a server responds to client requests for different URLs.
/ → Home Page /about → About Page /contact → Contact Page

2. Implement Basic Routing
const http = require('http'); const server = http.createServer((req, res) => {
res.writeHead(200, {'Content-Type': 'text/html'}); if (req.url === '/') { res.end('Home PageWelcome!'); } else if (req.url === '/about') { res.end('About UsThis is the about page.'); } else if (req.url === '/contact') { res.end('ContactContact us anytime.'); } else { res.writeHead(404); res.end('404 Not Found'); } }); server.listen(3000, () => console.log('Server running at http://localhost:3000'));

3. Serve External HTML Files
const http = require('http'); const fs = require('fs'); const server = http.createServer((req, res) => { if (req.url === '/') { fs.readFile('index.html', (err, data) => { res.writeHead(200, {'Content-Type': 'text/html'}); res.end(data); }); } else if (req.url === '/about') { fs.readFile('about.html', (err, data) => { res.writeHead(200, {'Content-Type': 'text/html'}); res.end(data); }); } else { res.writeHead(404); res.end('404 Not Found'); } }); server.listen(3000, () => console.log('Server running...'));

4. Add JSON Response
if (req.url === '/api') { const users = [{ name: 'Alice', age: 25 }, { name: 'Bob', age: 30 }]; res.writeHead(200, {'Content-Type': 'application/json'}); res.end(JSON.stringify(users)); }

■ Mini Project — Simple Router System

Build a Node.js server that:
• Has routes: '/', '/about', '/api', and '/404'
• Serves HTML and JSON responses
• Logs requests with timestamps
Bonus: Use **fs** and **path** to read HTML from a /pages folder.

■ Recap Discussion

- What is the purpose of the HTTP module?
- Difference between inline HTML and file-based responses?
- How to serve JSON data from Node.js?
- What happens when a route doesn’t exist?

■ Homework

Create a routing system that:
• Serves pages from a /pages folder
• Handles routes: /home, /services, /about, /contact

• Logs each request URL with timestamp in log.txt
• Returns a custom 404 page for unknown routes
Example log:
2025-10-15 14:22:10 - /about 2025-10-15 14:23:01 - /home