

Lecture 3. Creating HTTP web server in Node js

08 April 2025 19:44

Node.js HTTP Server with Logging – Summary

This script creates a basic **HTTP web server** in Node.js that:

- Listens for incoming requests.
- Logs each request to a log.txt file with a timestamp.
- Responds with different content based on the route (/ , /about, or others).

Modules Used

js

CopyEdit

```
const http = require("http");
```

```
const fs = require("fs");
```

- http: Core Node.js module to create a web server.
- fs: Used to append request logs to a file (log.txt).

Server Creation and Routing

js

CopyEdit

```
const myserver = http.createServer((req, res) => {
```

```
  const log = `${Date.now()}: ${req.url} New Request Recieved\n`;
```

- Creates the server using http.createServer().
- Logs request time and URL using Date.now() and req.url.

Logging Each Request

js

CopyEdit

```
fs.appendFile('log.txt', log, (err, data) => {
```

```
  switch(req.url){
```

```
    case '/': res.end("Home Page");
```

```
    break;
```

```
    case '/about': res.end("I am krishan");
```

```
    break;
```

```
    default: res.end("404 Not Found");
```

```
  }
```

```
});
```

- Appends log entries to log.txt **asynchronously**.
- Inside the callback:
 - Uses switch to handle simple routing:
 - / → "Home Page"
 - /about → "I am krishan"
 - Others → "404 Not Found"

 **Issue in Code:**

js

CopyEdit

```
res.end("Hello from Server Again");
```

This line **comes after another res.end()** inside the switch, which causes an error because you can

only call `res.end()` **once per response**. You should **remove** or relocate it properly.

Starting the Server

js

CopyEdit

```
myserver.listen(8000, () => console.log("server started"));
```

- Starts the server on port 8000.
- Logs "server started" in the console.

Optimized & Fixed Version

Here's the **corrected version** of the code:

js

CopyEdit

```
const http = require("http");
const fs = require("fs");
const myserver = http.createServer((req, res) => {
  const log = `${Date.now()}: ${req.url} New Request Recieved\n`;
  fs.appendFile('log.txt', log, (err) => {
    if (err) {
      res.end("Error logging request");
      return;
    }
  });
  switch (req.url) {
    case '/':
      res.end("Home Page");
      break;
    case '/about':
      res.end("I am krishan");
      break;
    default:
      res.end("404 Not Found");
  }
});
myserver.listen(8000, () => console.log("server started"));
```

Key Concepts Recap

Feature	Purpose
<code>http.createServer</code>	Creates a Node.js web server
<code>req.url</code>	Captures the requested route
<code>fs.appendFile</code>	Asynchronously logs request data to a file
<code>res.end()</code>	Sends a response back to the client and ends the request
<code>server.listen(8000)</code>	Starts server on port 8000