

Lecture 4. Handling Url's in Nodejs

08 April 2025 23:21

```
// Creating Http web server
const http = require("http");
const fs = require("fs");
const url = require("url");

const myserver = http.createServer((req, res) => {
  if (req.url === '/favicon.ico') return res.end(); // Ignore favicon

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

  const myUrl = url.parse(req.url, true); // Parse the full URL
  console.log(myUrl); // For debugging: shows pathname and query

  fs.appendFile('log.txt', log, (err, data) => {
    if (err) {
      res.writeHead(500);
      return res.end("Internal Server Error");
    }
  });

  switch (myUrl.pathname) {
    case '/':
      res.end("Home Page");
      break;

    case '/about':
      const username = myUrl.query.myname;
      res.end(`Hiii, ${username}`);
      break;

    case '/search':
      const search = myUrl.query.search_query;
      res.end("Here are your results for " + search);
      break;

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

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

Approach:-

What Happens Step-by-Step?

1. **Server starts** on port 8000.
2. On each request:
 - Ignores /favicon.ico.
 - Logs the URL in log.txt with a timestamp.
 - Parses the URL and its query string.
 - Switches on the pathname to serve different responses.

URL Patterns to Test:

- / → Returns: Home Page
- /about?myname=Krishan → Returns: Hiii, Krishan
- /search?search_query=nodejs → Returns: Here are your results for nodejs
- /xyz → Returns: 404 Not Found

Key Learnings:

- `url.parse(req.url, true)` helps break the URL into:
 - `pathname` → `/about`
 - `query` → `{ myname: 'Krishan' }`
- Never use `res.end()` more than once per request.
- Always check for fs errors to prevent server crashes.