

# Send HTML File in Response using FS Module

Node JS Tutorial in Hindi #14 - Send HTML File in Response using FS Mo...



## Why load an HTML file with fs?

Instead of writing HTML as a string in `resp.write()`, it's more scalable and readable to keep HTML in a separate `.html` file.

This approach is cleaner, especially for large or structured HTML.

## Server File (`web.js`)

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

```
http.createServer((req, resp) => {}).listen(3200);
```

Creates a valid HTTP server.

Listens on port 3200.

## HTML File (web.html)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Web page</title>
  </head>
  <body>
    <h1>Web Page Heading</h1>
    <input type="text" />
    <button>Click</button>
  </body>
</html>
```

Well-formed HTML structure.

Proper title, meta tags, and UI elements.

## Read and Load HTML File Using fs Module

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

http
  .createServer((req, resp) => {
```

```
fs.readFile("html/web.html", "utf-8", (err, data) => {
  if (err) {
    resp.writeHead(500, {
      "Content-Type": "text/plain",
    });
    resp.write("internal server error");
    resp.end();
    return;
  }

  resp.writeHead(200, {
    "Content-Type": "text/html",
  });
  resp.write(data);
  resp.end();
});
}).listen(3200);
```

Correctly reads the file `html/web.html` (assuming the folder structure is correct).

Sets appropriate status codes (500 for error, 200 for success).

Handles errors gracefully.

Sends the HTML content as response.

## Check Output

When you run `node web.js` or `nodemon web.js`, and visit `http://localhost:3200`, the browser will correctly show the HTML content from `web.html`.

[Home](#) [About](#) [Contact](#)