

Ex. No: 6	Web Server Creation using NodeJS
21.09.2023	

Aim: To Create a Web Server offering basic web service(s) to the front-end.

Algorithm:

1. Ensure you have Node.js installed on your system.
2. Develop a JavaScript file (e.g., `server.js`) for your web server.
3. In `server.js`, require Node.js's built-in `http` module using `require('http')`.
4. Use the `http.createServer()` method to create an HTTP server, specifying a request handling function.
5. Inside the request handling function, use the `request` and `response` objects to define how your server should respond to different routes and HTTP methods.
6. Test your web server using tools like cURL or Postman. Debug and refine your route handling as needed.
7. Optionally, configure the web server to serve static HTML, CSS, and JavaScript files if your front-end includes them, using the `fs` (file system) module.

Program:

Server.js

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

const server = http.createServer((req, res) => {
  const reqUrl = url.parse(req.url, true);

  if (reqUrl.pathname === "/" || reqUrl.pathname === "/index.html") {
    // Serve the HTML page
    fs.readFile(path.join(__dirname, "public", "index.html"), (err, data) => {
      if (err) {
        res.writeHead(500, { "Content-Type": "text/plain" });
        res.end("Internal Server Error");
      } else {
        res.writeHead(200, { "Content-Type": "text/html" });
        res.end(data);
      }
    });
  } else if (reqUrl.pathname === "/styles.css") {
    // Serve the CSS file
    fs.readFile(path.join(__dirname, "public", "styles.css"), (err, data) => {
      if (err) {
        res.writeHead(500, { "Content-Type": "text/plain" });
        res.end("Internal Server Error");
      }
    });
  }
});
```

```

    } else {
      res.writeHead(200, { "Content-Type": "text/css" });
      res.end(data);
    }
  });
} else if (reqUrl.pathname === "/script.js") {
  // Serve the JavaScript file
  fs.readFile(path.join(__dirname, "public", "script.js"), (err, data) => {
    if (err) {
      res.writeHead(500, { "Content-Type": "text/plain" });
      res.end("Internal Server Error");
    } else {
      res.writeHead(200, { "Content-Type": "text/javascript" });
      res.end(data);
    }
  });
} else if (reqUrl.pathname === "/notes" && req.method === "GET") {
  // Handle GET request to retrieve notes (simulated in-memory storage)
  const notes = [
    { id: 1, text: "Buy groceries" },
    { id: 2, text: "Call John" },
  ];
  res.writeHead(200, { "Content-Type": "application/json" });
  res.end(JSON.stringify(notes));
} else {
  // Handle other routes with a 404 Not Found response
  res.writeHead(404, { "Content-Type": "text/plain" });
  res.end("Not Found");
}
});

const port = process.env.PORT || 3000;
server.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});

```

Index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Server-Side Notes</title>
<link rel="stylesheet" href="styles.css">
</head>

```

```

<body>
<h1>Server-Side Notes</h1>
<div class="notes-container">
<textarea id="noteInput" placeholder="Add a new note">
</textarea>
<button id="addNote">Add Note</button>
</div>
<div class="notes-list">
</div>
<script</body>
</html>
src="script.js"></script>

```

Script.js

```

document.addEventListener("DOMContentLoaded", () => {
const noteInput = document.getElementById("noteInput");
const addNoteButton = document.getElementById("addNote");
const notesList = document.querySelector(".notes-list");
// Fetch and display notes
fetch("/notes")
.then((response) => response.json())
.then((notes) => {
notes.forEach((note) => {
displayNote(note);
});
})
.catch((error) => {
console.error("Error fetching notes:", error);
});
// Add a new note
addNoteButton.addEventListener("click", () => {
const text = noteInput.value.trim();
if (text) {
fetch("/notes", {
method: "POST",
headers: {
"Content-Type": "application/json",
},
body: JSON.stringify({ text }),
})
.then((response) => response.json())
.then((newNote) => {
displayNote(newNote);
noteInput.value = "";
})
.catch((error) => {

```

```


console.error("Error adding note:", error);
});
}
});
// Display a note
function displayNote(note) {
const noteElement = document.createElement("div");
noteElement.className = "note";
noteElement.textContent = note.text;
notesList.appendChild(noteElement);
}
});

```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023

Server-Side Notes



Server Side output:



Result:

Therefore, we've successfully implemented a web server backend using NodeJS .