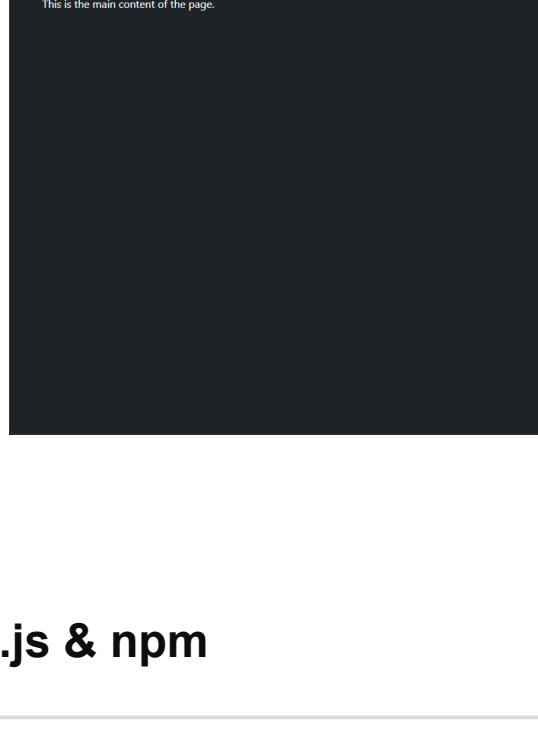


1. Setup & Hello World App



1. Install Node.js & npm

```
md

# Setting Up a Basic Node.js Environment to Serve `index.html`

Below are step-by-step instructions to help you set up a new Node.js project,
serve the `index.html` file locally, and test it in your browser. You can
copy/paste these steps directly into your notes or project documentation.

## 1. Install Node.js and npm
1. **Download and install Node.js** (includes npm) from the official [Node.js
website](https://nodejs.org).
2. Verify installation by opening a terminal (Command Prompt, PowerShell, or VS
Code Terminal) and running:

```bash
node -v
npm -v
```

You should see version numbers displayed for both commands.

## 2. Create a New Folder for Your Project

1. Open **Visual Studio Code**.
2. Create or open a folder where you want to store your Node.js project. For example, you might name it `my-node-setup`.

## 3. Initialize npm

1. In the **VS Code terminal**, navigate (`cd`) to your project folder:

```
bash

cd path/to/my-node-setup
```

2. Run the following command to initialize a new npm project and automatically create a `package.json` file:

```
bash

npm init -y
```

This command adds some default settings to your `package.json`.

## 4. Create the Project Structure

Inside your project folder ( `my-node-setup` ), create the following structure:

```
pgsql

my-node-setup
├── public
│ ├── index.html
│ └── partials
│ └── navbar1.html
└── server.js
```

- `public/index.html` : Place your **Hello World** HTML code here (the code you provided).
- `public/partials/navbar1.html` : Place your navbar partial here.
- `server.js` : Will contain the Node.js/Express server logic.

### 4.1. Copy Your `index.html`

`public/index.html` (Replace the file content with the code you shared):

```
html

<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <title>MyCompany.com</title>
 <!-- Load Bootstrap CSS (dark nav) -->
 <link
 rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
 />
</head>
<body class="bg-dark text-white">
 <!-- Navbar will be injected here -->
 <div id="navbar-container"></div>

 <!-- Main content -->
 <div class="container mt-4">
 <h1>Hello, World!</h1>
 <p>This is the main content of the page.</p>
 </div>

 <!-- Bootstrap JS + dependencies -->
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

 <!-- Script to Load the navbar partial -->

</script>

document.addEventListener('DOMContentLoaded', () => {
 fetch('/partials/navbar1.html')
 .then(response => response.text())
 .then(navbarHTML => {
 document.getElementById('navbar-container').innerHTML = navbarHTML;
 })
 .catch(error => console.error('Error loading navbar:', error));
});
</script>
</body>
</html>
```

### 4.2. Create the `navbar1.html` Partial

`public/partials/navbar1.html` (Replace or customize as you like):

```
html

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container-fluid">
 MyCompany.com

 <button
 class="navbar-toggler"
 type="button"
 data-bs-toggle="collapse"
 data-bs-target="#navbarSupportedContent"
 aria-controls="navbarSupportedContent"
 aria-expanded="false"
 aria-label="Toggle navigation"
 >

 </button>
 <div class="collapse navbar-collapse" id="navbarSupportedContent">
 <ul class="navbar-nav me-auto">
 <li class="nav-item">
 Home

 <!-- Add more nav items as needed -->

 </div>

</div>
</nav>
```

## 5. Install Express.js

To serve static files easily, we'll use [Express.js](#). In your project folder ( `my-node-setup` ), run:

```
bash

npm install express
```

This will create a `node_modules` folder and add Express to your `package.json` dependencies.

## 6. Create a Simple Server ( `server.js` )

Inside your project folder, create `server.js` with the following content:

```
js

const express = require('express');
const path = require('path');
const app = express();

// Serve all files in the "public" folder
app.use(express.static(path.join(__dirname, 'public')));

// Define a port (default to 3000)
const PORT = process.env.PORT || 3000;

// Start the server
app.listen(PORT, () => {
 console.log(`Server is running at http://localhost:${PORT}`);
});
```

**What this file does:**

- Imports `express`.
- Configures the app to serve static files from the `public` directory.
- Starts a server on port **3000** (or another port if specified in the environment variables).
- Logs a message when the server is up and running.

## 7. Run the Server

In your **VS Code terminal**, ensure you're in the project folder and run:

```
bash

node server.js
```

You should see something like:

```
arduino

Server is running at http://localhost:3000
```

## 8. Test in Your Browser

1. Open **Google Chrome** (or any browser).
2. Go to: <http://localhost:3000>.
3. You should see **"Hello, World!"** and the navbar loaded at the top. The background should be dark, as per the Bootstrap theme.

## 9. Next Steps

- You can edit the HTML, CSS, or partial files in the `public` folder and simply **refresh** the page in your browser to see updates.
- For larger changes, consider stopping and restarting the server by pressing `Ctrl + C` in the terminal and then running `node server.js` again.

**That's it!** You've successfully set up a minimal Node.js server with an `index.html` file and

a partial. Now you can begin building more complex functionality on top of this foundation.

```
Copy Edit
```

Ask anything