

Day 4: Creating a Simple Web Server with Node.js

What is a server?

A server is just a computer that waits for requests and responds to them. It's like a restaurant waiter: you ask for something, and it brings it to you.

What does `http.createServer()` do?


It creates a basic server that listens for requests and sends back responses.

Example:

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

```
const server = http.createServer((req, res) => {  
  res.end('Hello from the server!');  
});
```

```
server.listen(3000, () => {  
  console.log('Server is running on http://localhost:3000');  
});
```

Open your browser and go to `http://localhost:3000` — you'll see “Hello from the server!” 

4

What is a port?

A port is like a door number for your app on your computer. You can have many apps running, and each

one listens on a different port.

What is a request and response?

- Request = What the user (browser) asks for
- Response = What the server sends back

What if I want to do calculations on the backend? Do the calculation in Node and return the result!

Example:

```
const server = http.createServer((req, res) => {  
  if (req.url === '/sum') {  
    const result = 40 + 2;  
    res.end(`The sum is ${result}`);  
  } else {  
    res.end("Hello from the server");  
  }  
});
```

How does frontend talk to backend?

- Frontend makes an HTTP request (using fetch, axios, etc.)
- Backend sends a response with data
- Frontend updates the UI with it

It's like a waiter bringing food from the kitchen!

Task

1. Create a file called server.js
2. Create different routes (/ , /about , /sum)
3. Do a calculation on /sum
4. Run and test in browser

🎯 Stretch goal: Connect frontend fetch() with your backend and show data in the UI You've officially made your own server today — not bad for Day 4!