

Lesson:

How to Create Server in Nodejs



Topics

- HTTP modules
- Server with HTTP modules
- Features of HTTP method

HTTP modules

In Nodejs, the HTTP (which stands for Hyper Transfer protocol) module is a core module that provides functionality for creating and interacting with HTTP servers and clients. The HTTP modules in Nodejs are designed to support many features of the protocol which have been traditionally difficult to use. The module's interface is careful design never to buffer entire requests or responses, so the user is able to stream data.

To use the HTTP server client must require("http") which is a standard way to import the module and is widely supported.

An HTTP (Hypertext Transfer Protocol) request is a message sent by a client to a server, requesting for some information or action to be performed. It is the primary means by which web browsers, search engines, and other web clients communicate with web servers.

Server with HTTP modules

There are several ways to create a server in Nodejs, one of the simplest ways is to create it using HTTP modules.

Firstly create a folder called **node** and inside it, make a file called **server.js**

To create the server with built-in HTTP modules firstly import the built-in HTTP module, **node/server.js**

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

The **http** module is a built-in module in Node.js that provides functionality for creating and interacting with HTTP servers and clients

The next is to define or specify the hostname and the port number.
node/server.js

```
const port = 3000
const host = 'localhost' // 127.0.0.1
```

Port – a port number is a communication endpoint in an operating system that identifies a specific process to which message or data package is to be forwarded.

Host(localhost) – Localhost is a hostname that refers to the local computer or the computer that a user is currently using. It is commonly used in network communications and refers to the loopback network interface (virtual network interface that allows a computer to communicate with itself using internet protocol Protocol), which allows a computer to communicate with itself using the IP address 127.0.0.1.

In Node.js, the **http.createServer()** method is used to create an HTTP server. The createServer() method takes a callback function as its argument, which will be called whenever a new request is received by the server.

node/server.js

```
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-type", "text/plain");
  res.end("hello world");
});
```

Inside the callback function, request (req) and response(res) are the two arguments passed which represent the incoming requests and outgoing responses, respectively, the response status code is set to 200 using **res.statusCode = 200**, this indicates that the request has been successfully processed. And also set the content-type, header to text/plain using **res.setHeader("Content-type", "text/plain")**, This tells the client that the response contains plain text.

Finally, the response is ended by calling **res.end("hello world")**, this sends a response back to the client with the message "hello world"

server.js

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

const port = 3000;
const host = "localhost";
//creating server with http instance
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader("Content-type", "text/plain");
  res.end("hello world");
});

server.listen(port, () => {
  console.log(`Server running at ${host}:${port}`);
});

// 1. Console Output
// Server running at localhost:3000

// 2. Browser Output
// hello world
```

The **listen** method is called on the **server** object to start the server and make it listen on port 3000 using **server.listen()**. When the server is ready to receive requests, the callback function passed to listen is called, and the message "**Server running at localhost:3000**" is logged to the console.

The server can be tested by visiting <http://localhost:3000> in the browser or postman.

The Server in Nodejs can also be created using third parties library like express and koa which are popular web frameworks for Nodejs

Features of HTTP method

The HTTP module is a built-in module in Node.js that provides functionality for creating and interacting with HTTP servers and clients. It provides a simple way to create web servers and handle HTTP requests and responses.

Some of the key features of HTTP module are as follows:

1. Creating an HTTP server - create an HTTP server using the **http.createServer()** method, which takes a request listener function as an argument. This function will be called every time a request is made to the server.
2. Handling HTTP requests - The request listener function can be used to handle incoming HTTP requests. It can read the request headers and body, and send a response back to the client.
3. Sending HTTP request - The **http.request()** method can be used to send HTTP requests from a Node.js application. This can be useful for interacting with other web servers or APIs.

Example of **http.request**

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

//API - https://fakestoreapi.com/products/1
const options = {
  hostname: "fakestoreapi.com",
  path: "/products/1",
  methods: "GET",
};

// http.request with two arguments
const req = http.request(options, (res) => {
  res.on("data", (d) => {
    process.stdout.write(d); // write data directly to standard output stream
  });
});
req.on("error", (e) => {
  console.error(e);
});
req.end();

// output - logs a single product details
{
  "id": 1,
  "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops",
  "price": 109.95,
  "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday",
  "category": "men's clothing",
  "image": "https://fakestoreapi.com/img/81fPKd-2AYL._AC_SL1500_.jpg",
  "rating": {
    "rate": 3.9,
    "count": 120
  }
}
```