

HTTP server with core http

Although Node.js can be used for a wide variety of tasks, it's used primarily for building web applications. Node.js thrives in networking as a result of its asynchronous nature and built-in modules such as `net` and `http`. Node is great for building fast and efficient web servers.

Here's a quintessential Hello World example in which we create a server object, define the request handler (function with `req` and `res` arguments), pass some data back to the recipient, and start up the whole thing (`server.js`):

```
const http = require('http')
const port = 3000
http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'})
  res.end('Hello World\n')
}).listen(port)

console.log(`Server running at http://localhost:${port}/`)
```

Let's break it down a bit. The following loads the core `http` module for the server:

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

This snippet below creates a server with a callback function which contains the response handler code:

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

To set the right header and status code, use the following:

```
res.writeHead(200, {'Content-Type': 'text/plain'})
```

To output Hello World with the line end symbol, use: