Lesson 05 Demo 05

Creating HTTP Parser

Objective: To create an HTTP parser in a Node.js app for handling both HTTP requests and responses

Tools required: Node Package Manager and Visual Studio Code

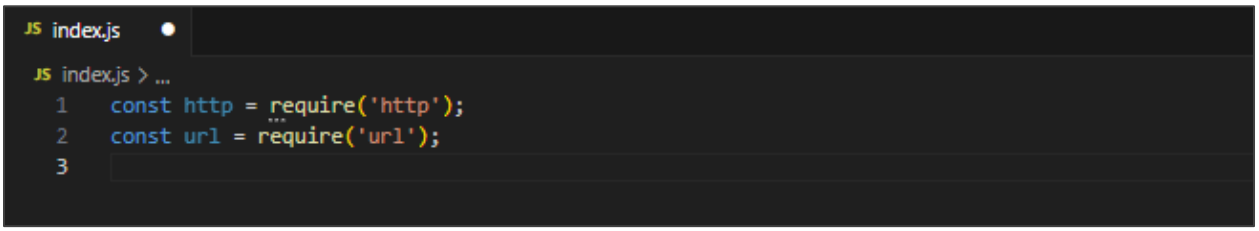
Prerequisites: Basic Linux Commands, NPM commands, JavaScript, and HTTP module

Steps to be followed:

1. Create an HTTP parser for an HTTP request

Step 1: Create an HTTP parser for an HTTP request

1.1 Import the HTTP and URL modules:



1.2 Create the server port and hostname:
const SERVER_PORT = 3000;
const SERVER_HOSTNAME = "127.0.0.1";

1.3 Use the **createServer** function to create the HTTP server:
const server = http.createServer();

```
index.js
index.js > ...
1   const http = require('http');
2   const url = require('url');
3
4   const SERVER_PORT = 3000;
5   const SERVER_HOSTNAME = "127.0.0.1";
6
7   const server = http.createServer();
```

1.4 Create the listener on the server to handle each request, parse through the query string, and convert it into a key-value JavaScript object:
server.on("request", (req, res)=> {

```
 const query = url.parse(req.url).query
 const queryObj = query.split("&").reduce((prev, next) => {
  let [key, value] = next.split("=");
  return { ...prev, [key]: value }
 }, {});
}
);
```

```
index.js > ...
1   const http = require('http');
2   const url = require('url');
3
4   const SERVER_PORT = 3000;
5   const SERVER_HOSTNAME = "127.0.0.1";
6
7   const server = http.createServer();
8
9   server.on("request", (req, res)=> {
10
11
12      const query = url.parse(req.url).query
13      const queryObj = query.split("&").reduce((prev, next) => {
14        let [key, value] = next.split("=");
15        return { ...prev, [key]: value }
16      }, {});
17  }
18  );
19
20
```

1.5 Set the **statusCode** as 200 and send the JSON response:

res.statusCode = 200;
res.setHeader("Content-Type", "application/json")
return res.end(JSON.stringify(queryObj));

1.6 Specify the port number and hostname to run the server:

server.listen(SERVER_PORT, SERVER_HOSTNAME, () => {
console.log(`Server is up and listening on port ${SERVER_PORT}`);
})

1.7 Open the terminal and execute the command:

node index.js

1.8 Navigate to the Chrome browser and open the following URL to view the output:

127.0.0.1:3000/name=Fionna&email=fionna@example.com&age=26

By following these steps, you have successfully created an HTTP parser in a Node.js app for handling both HTTP requests and responses.