Exercise (Instructions): Node and the HTTP Module | Coursera

Objectives and Outcomes

In this exercise, you will explore three core Node modules: HTTP, fs and path. At the end of this exercise, you will be able to:

- Implement a simple HTTP Server
- Implement a server that returns html files from a folder

A Simple HTTP Server

- Create a folder named *node-http* at a convenient location and move into the folder.
- In the *node-http* folder, create a subfolder named *public*.
- Create a file named *server-1.js* and add the following code to it:

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

var hostname = 'localhost';
var port = 3000;

var server = http.createServer(function(req, res){
  console.log(req.headers);
    res.writeHead(200, { 'Content-Type': 'text/html' });
  res.end('<html><body><h1>Hello World</h1></body></html>');
  })
server.listen(port, hostname, function(){
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- Start the server by typing the following at the prompt:

```
     node server-1
```

- Then you can type http://localhost:3000 in your browser address bar and see the result.
- You can also use postman chrome extension to send requests to the server and see the response.

Serving HTML Files

- In the *public* folder, create a file named *index.html* and add the following code to it:

```
<html>
<title>This is index.html</title>
<body>
<h1>Index.html</h1>
<p>This is the contents of this file</p>
</body>
</html>
```

- Similarly create an aboutus.html file and add the following code to it:

```
<html>
<title>This is aboutus.html</title><body>
<h1>Aboutus.html</h1><p>This is the contents of the aboutus.html file</p>
</body>
</html>
```

- Then create a file named *server-2.js* and add the following code to it:

```
var http = require('http');
var fs = require('fs');
var path = require('path');

var hostname = 'localhost';
var port = 3000;

var server = http.createServer(function(req, res){
  console.log('Request for ' + req.url + ' by method ' + req.method);
  if (req.method == 'GET') {
    var fileUrl;
        if (req.url == '/') fileUrl = '/index.html';
    else fileUrl = req.url;
        var filePath = path.resolve('./public'+fileUrl);
        var fileExt = path.extname(filePath);
        if (fileExt == '.html') {
       fs.exists(filePath, function(exists) {                 if (!exists) {
                res.writeHead(404, { 'Content-Type': 'text/html' });
                res.end('<html><body><h1>Error 404: ' + fileUrl +
                        ' not found</h1></body></html>');
                return;
```

```
          }
                    res.writeHead(200, { 'Content-Type': 'text/html' });
          fs.createReadStream(filePath).pipe(res);
              });
    }
    else {

        res.writeHead(404, { 'Content-Type': 'text/html' });
        res.end('<html><body><h1>Error 404: ' + fileUrl +
                ' not a HTML file</h1></body></html>');
    }
  }
  else {
        res.writeHead(404, { 'Content-Type': 'text/html' });
        res.end('<html><body><h1>Error 404: ' + req.method +
                ' not supported</h1></body></html>');
  }
})

server.listen(port, hostname, function(){
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- Start the server, and send various requests to it and see the corresponding response.

Conclusions

In this exercise you learnt about using the Node HTTP module to implement a HTTP server.