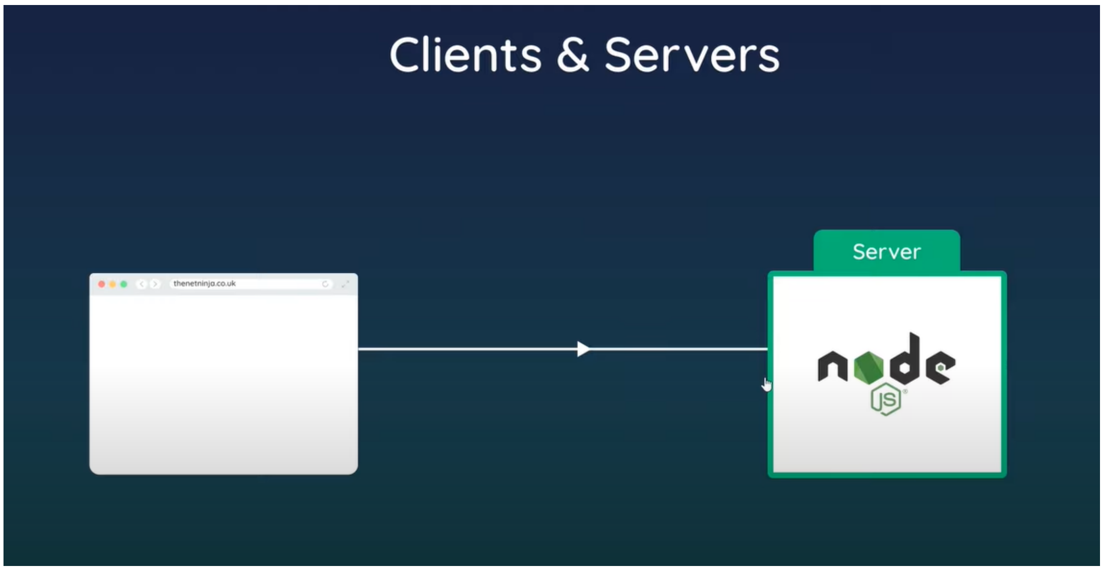
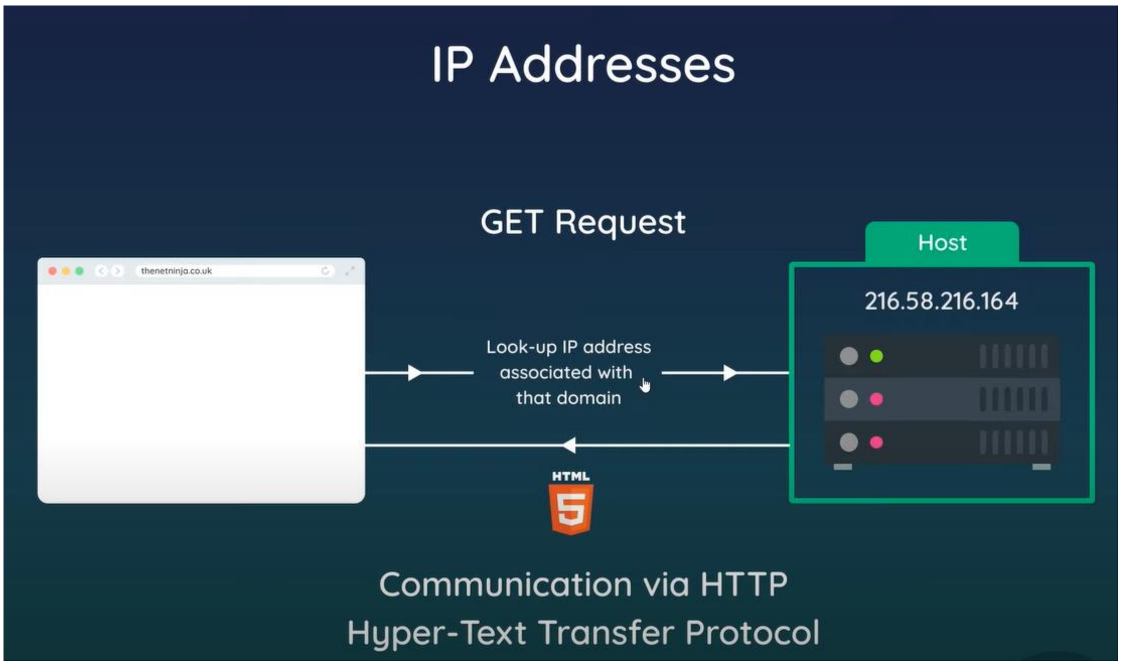
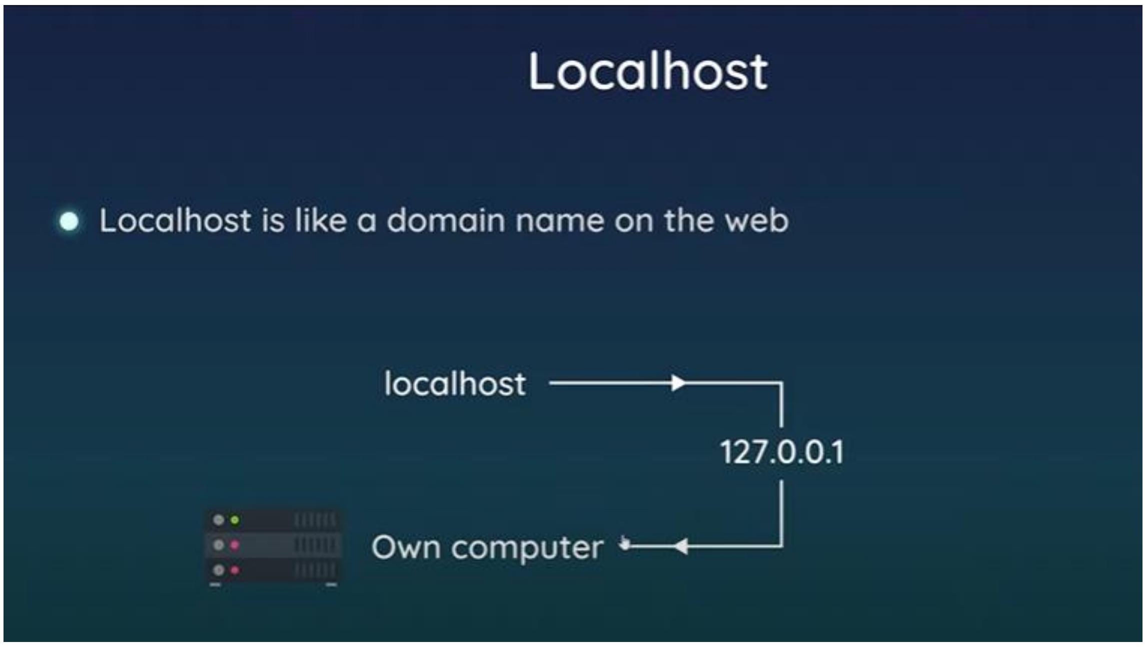
Clients & Servers



IP Addresses

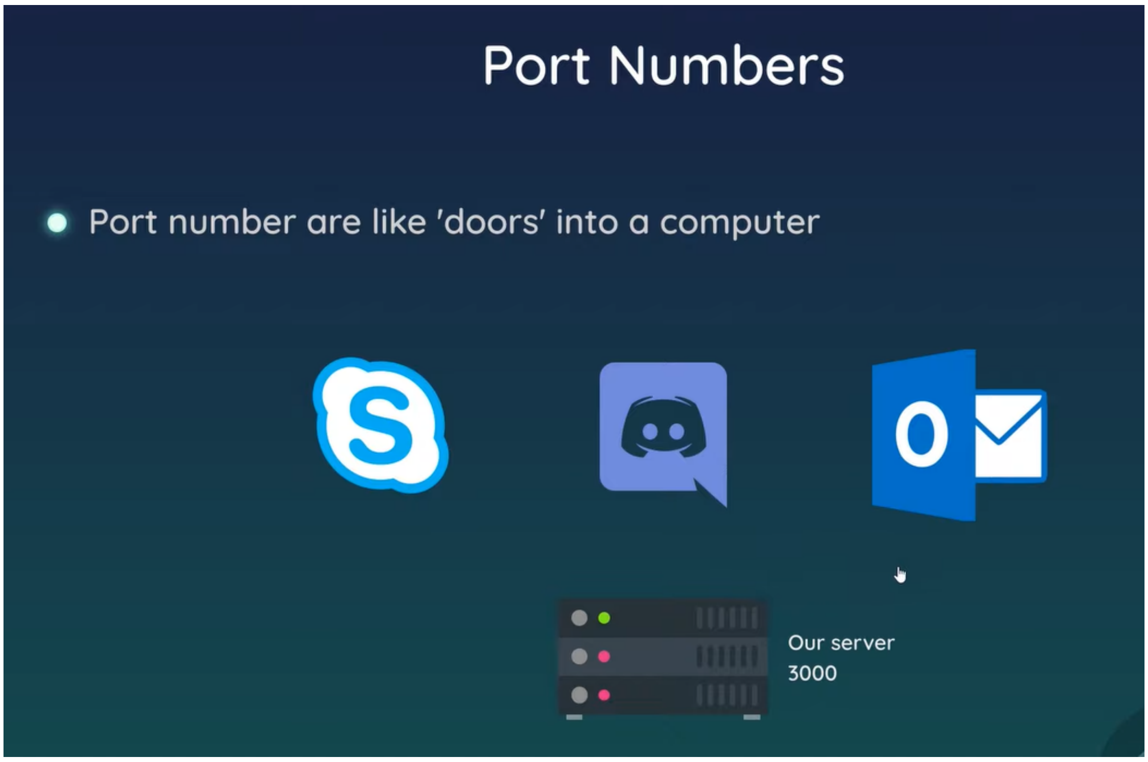


Localhost & Port Number



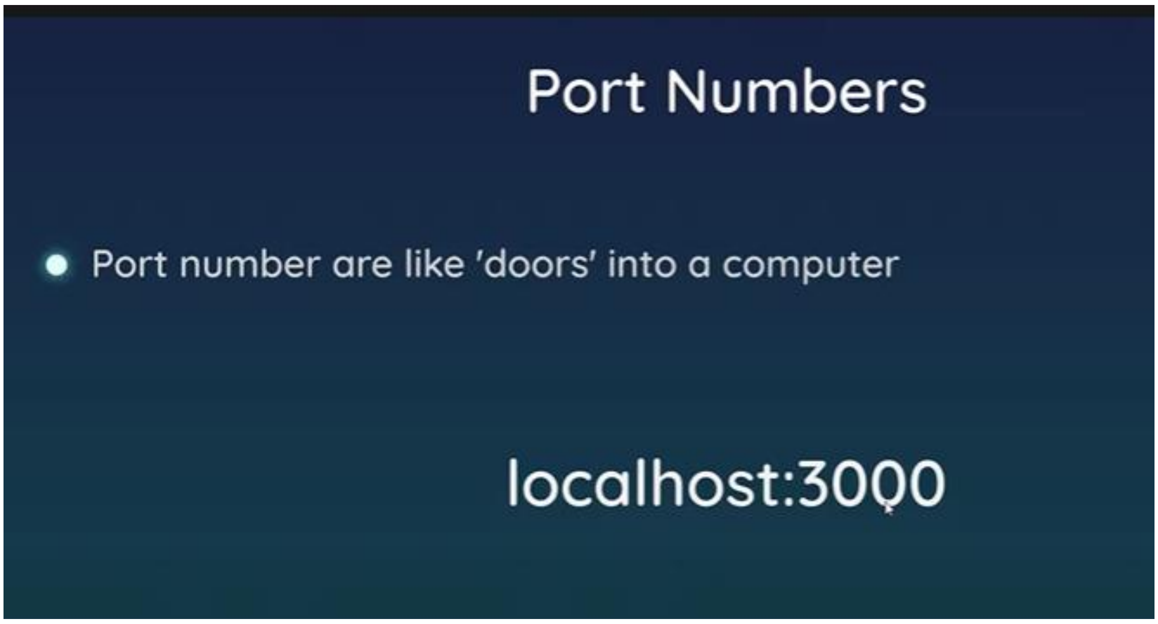
Localhost

- Localhost is like a domain name on the web



Port Numbers

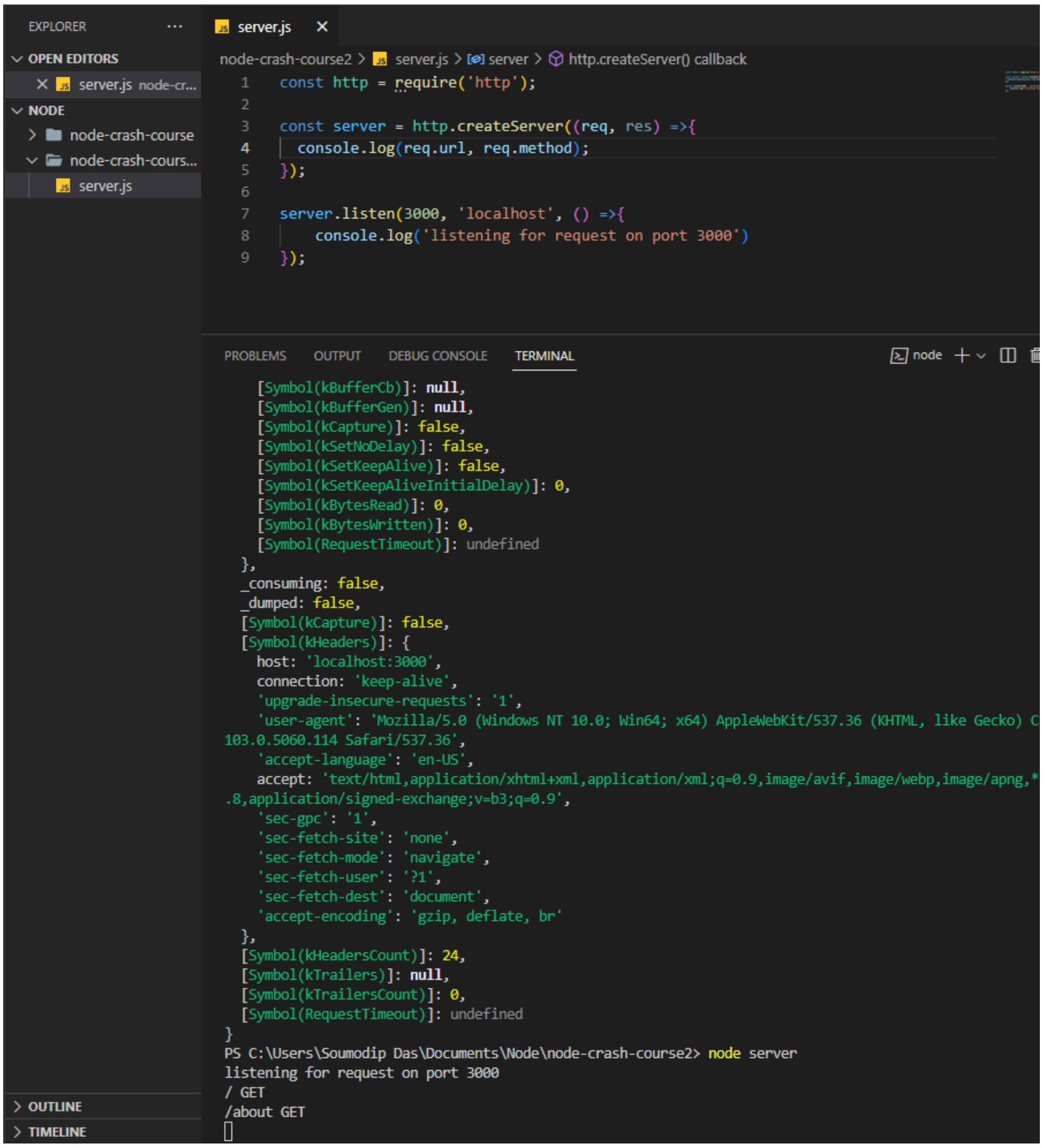
- Port number are like 'doors' into a computer



Port Numbers

- Port number are like 'doors' into a computer

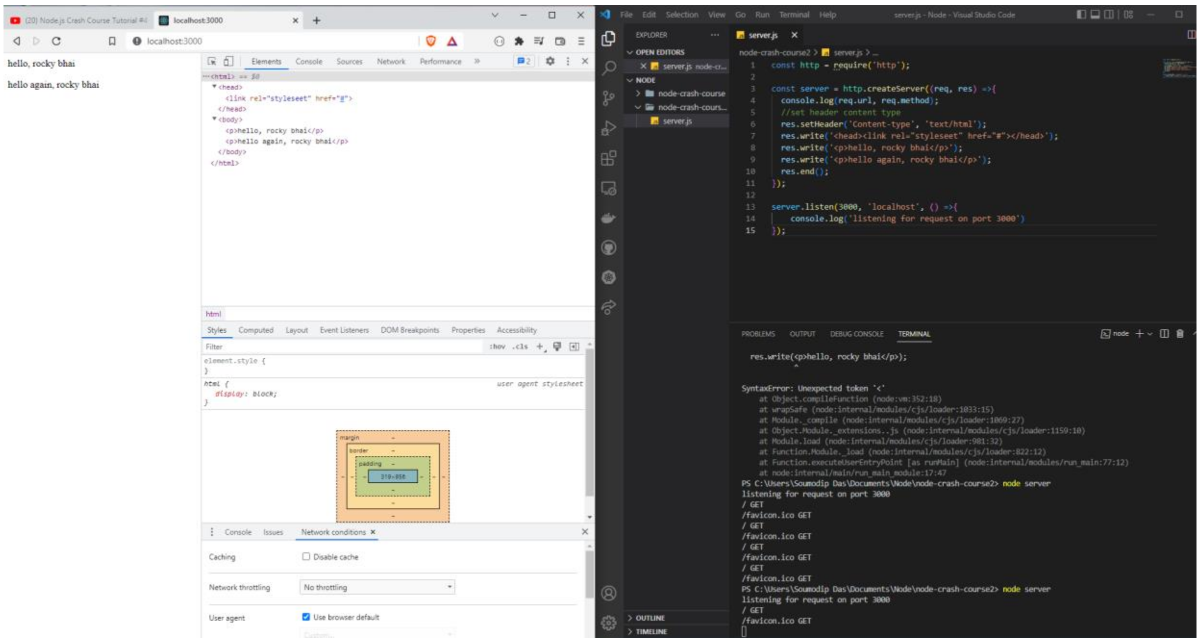
localhost:3000



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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
```

```
    [Symbol(kBufferCb)]: null,
    [Symbol(kBufferGen)]: null,
    [Symbol(kCapture)]: false,
    [Symbol(kSetNoDelay)]: false,
    [Symbol(kSetKeepAlive)]: false,
    [Symbol(kSetKeepAliveInitialDelay)]: 0,
    [Symbol(kBytesRead)]: 0,
    [Symbol(kBytesWritten)]: 0,
    [Symbol(RequestTimeout)]: undefined
  },
  _consuming: false,
  _dumped: false,
  [Symbol(kCapture)]: false,
  [Symbol(kHeaders)]: {
    host: 'localhost:3000',
    connection: 'keep-alive',
    'upgrade-insecure-requests': '1',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
103.0.5060.114 Safari/537.36',
    'accept-language': 'en-US',
    accept: 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
.8,application/signed-exchange;v=b3;q=0.9',
    'sec-gpc': '1',
    'sec-fetch-site': 'none',
    'sec-fetch-mode': 'navigate',
    'sec-fetch-user': '?1',
    'sec-fetch-dest': 'document',
    'accept-encoding': 'gzip, deflate, br'
  },
  [Symbol(kHeadersCount)]: 24,
  [Symbol(kTrailers)]: null,
  [Symbol(kTrailersCount)]: 0,
  [Symbol(RequestTimeout)]: undefined
}
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/about GET
```

The Response Object



hello, rocky bhai

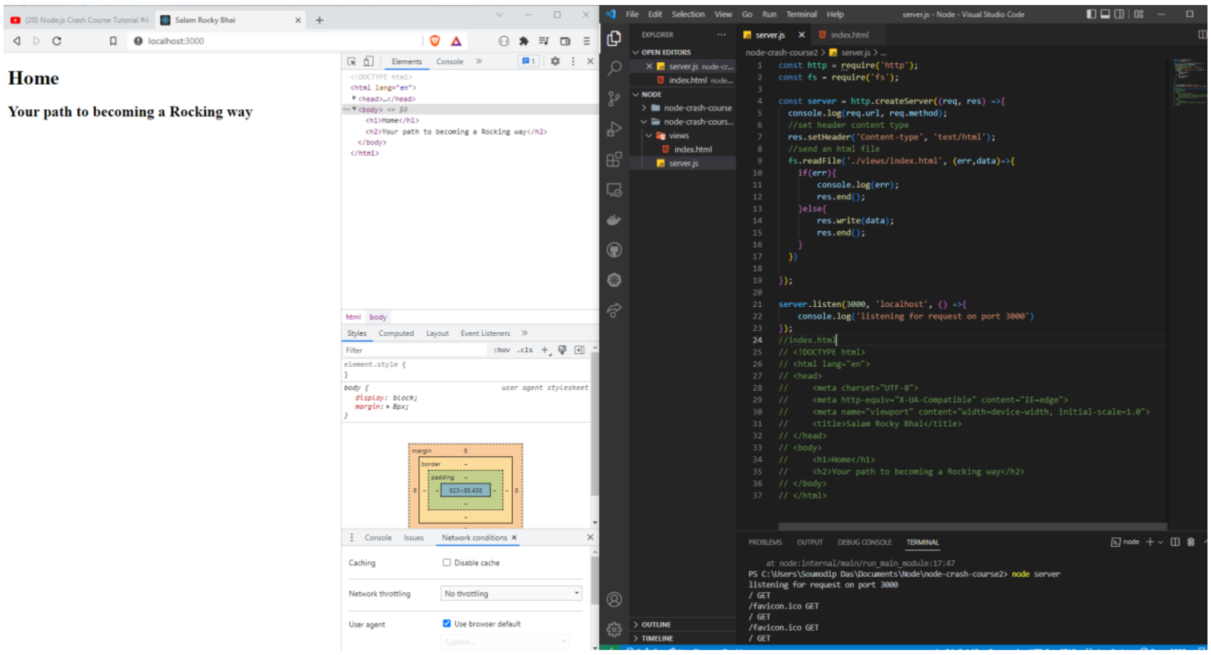
hello again, rocky bhai

```
<html> == $0
  <head>
    <link rel="styleseet" href="#">
  </head>
  <body>
    <p>hello, rocky bhai</p>
    <p>hello again, rocky bhai</p>
  </body>
</html>
```

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  res.write('<head><link rel="styleseet" href="#"></head>');
  res.write('<p>hello, rocky bhai</p>');
  res.write('<p>hello again, rocky bhai</p>');
  res.end();
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
```

```
  res.write(<p>hello, rocky bhai</p>);
            ^

SyntaxError: Unexpected token '<'
    at Object.compileFunction (node:vm:352:18)
    at wrapSafe (node:internal/modules/cjs/loader:1033:15)
    at Module._compile (node:internal/modules/cjs/loader:1069:27)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1159:10)
    at Module.load (node:internal/modules/cjs/loader:981:32)
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/favicon.ico GET
/ GET
/favicon.ico GET
/ GET
/favicon.ico GET
/ GET
/favicon.ico GET
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/favicon.ico GET
```

Returning HTML Pages



Home

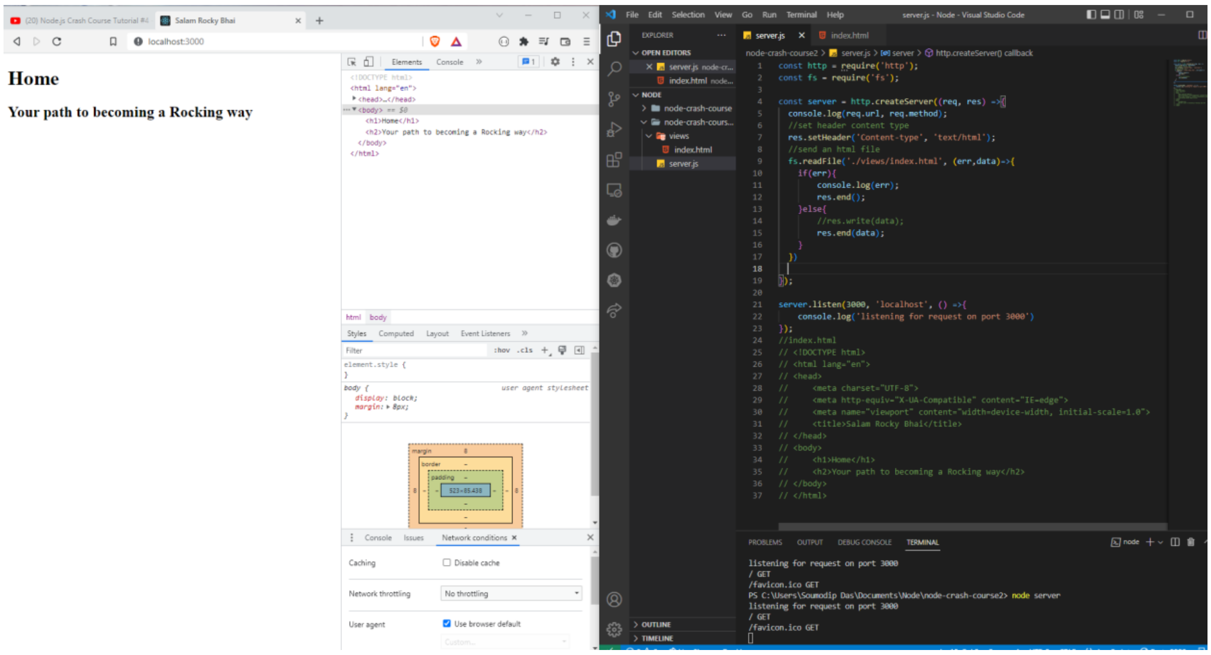
Your path to becoming a Rocking way

```
<!DOCTYPE html>
<html lang="en">
  <head>…</head>
  <body> == $0
    <h1>Home</h1>
    <h2>Your path to becoming a Rocking way</h2>
  </body>
</html>
```

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  //send an html file
  fs.readFile('./views/index.html', (err,data)=>{
    if(err){
        console.log(err);
        res.end();
    }else{
        res.write(data);
        res.end();
    }
  })

});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
//index.html
// <!DOCTYPE html>
// <html lang="en">
// <head>
//     <meta charset="UTF-8">
//     <meta http-equiv="X-UA-Compatible" content="IE=edge">
//     <meta name="viewport" content="width=device-width, initial-scale=1.0">
//     <title>Salam Rocky Bhai</title>
// </head>
// <body>
//     <h1>Home</h1>
//     <h2>Your path to becoming a Rocking way</h2>
// </body>
// </html>
```

```
    at node:internal/main/run_main_module:17:47
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/favicon.ico GET
/ GET
/favicon.ico GET
/ GET
```

OR



Home

Your path to becoming a Rocking way

```html
<!DOCTYPE html>
<html lang="en">
▸ <head>…</head>
▾ <body> == $0
    <h1>Home</h1>
    <h2>Your path to becoming a Rocking way</h2>
  </body>
</html>
```

```css
element.style {
}
body {                                   user agent stylesheet
  display: block;
  margin: ▸ 8px;
}
```

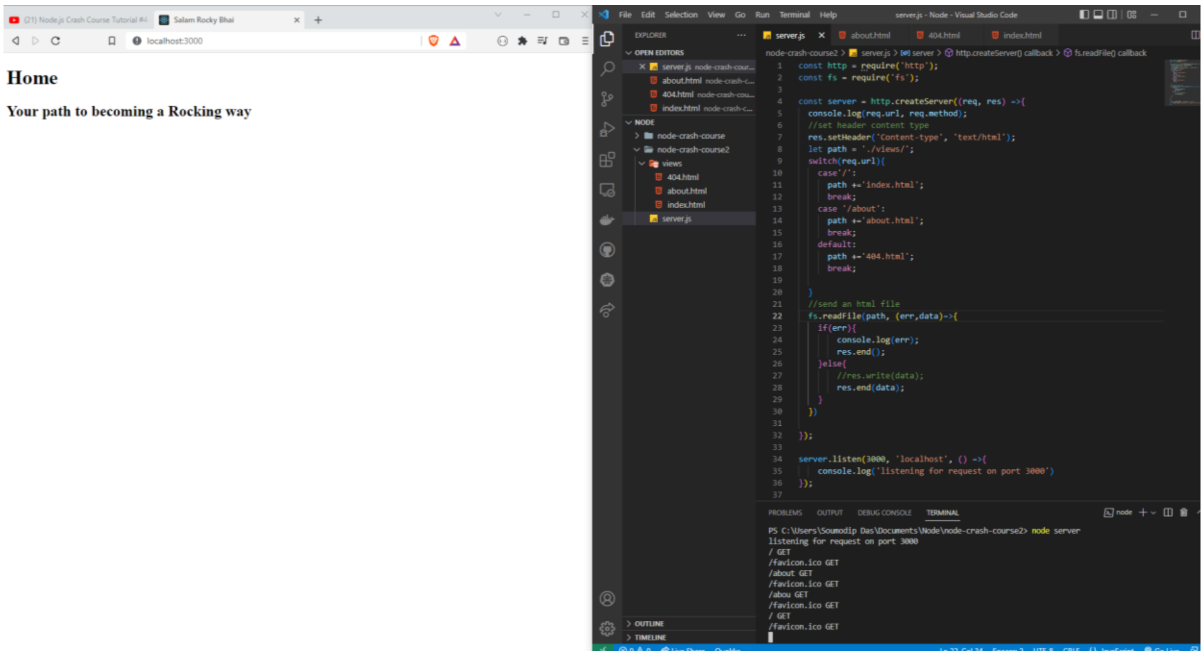
Caching: Disable cache
Network throttling: No throttling
User agent: [✓ Use browser default]

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  //send an html file
  fs.readFile('./views/index.html', (err,data)=>{
    if(err){
        console.log(err);
        res.end();
    }else{
        //res.write(data);
        res.end(data);
    }
  })
  
});

server.listen(3000, 'localhost', () =>{
   console.log('listening for request on port 3000')
});
//index.html
// <!DOCTYPE html>
// <html lang="en">
// <head>
//     <meta charset="UTF-8">
//     <meta http-equiv="X-UA-Compatible" content="IE=edge">
//     <meta name="viewport" content="width=device-width, initial-scale=1.0">
//     <title>Salam Rocky Bhai</title>
// </head>
// <body>
//     <h1>Home</h1>
//     <h2>Your path to becoming a Rocking way</h2>
// </body>
// </html>
```

```
listening for request on port 3000
/ GET
/favicon.ico GET
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/favicon.ico GET
```

Basic Routing

Home.html



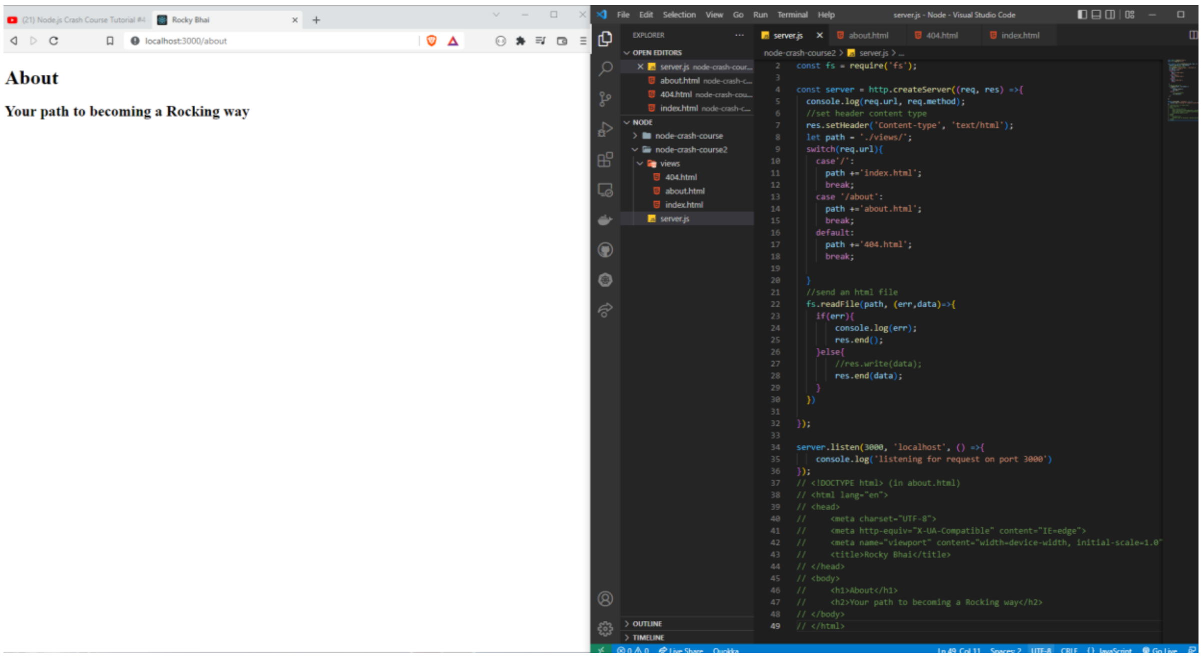
Home

Your path to becoming a Rocking way

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  let path = './views/';
  switch(req.url){
    case'/':
      path +='index.html';
      break;
    case '/about':
      path +='about.html';
      break;
    default:
      path +='404.html';
      break;

  }
  //send an html file
  fs.readFile(path, (err,data)=>{
    if(err){
        console.log(err);
        res.end();
    }else{
        //res.write(data);
        res.end(data);
    }
  })

});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
```

```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> node server
listening for request on port 3000
/ GET
/favicon.ico GET
/about GET
/favicon.ico GET
/abou GET
/favicon.ico GET
/ GET
/favicon.ico GET
```

About.html



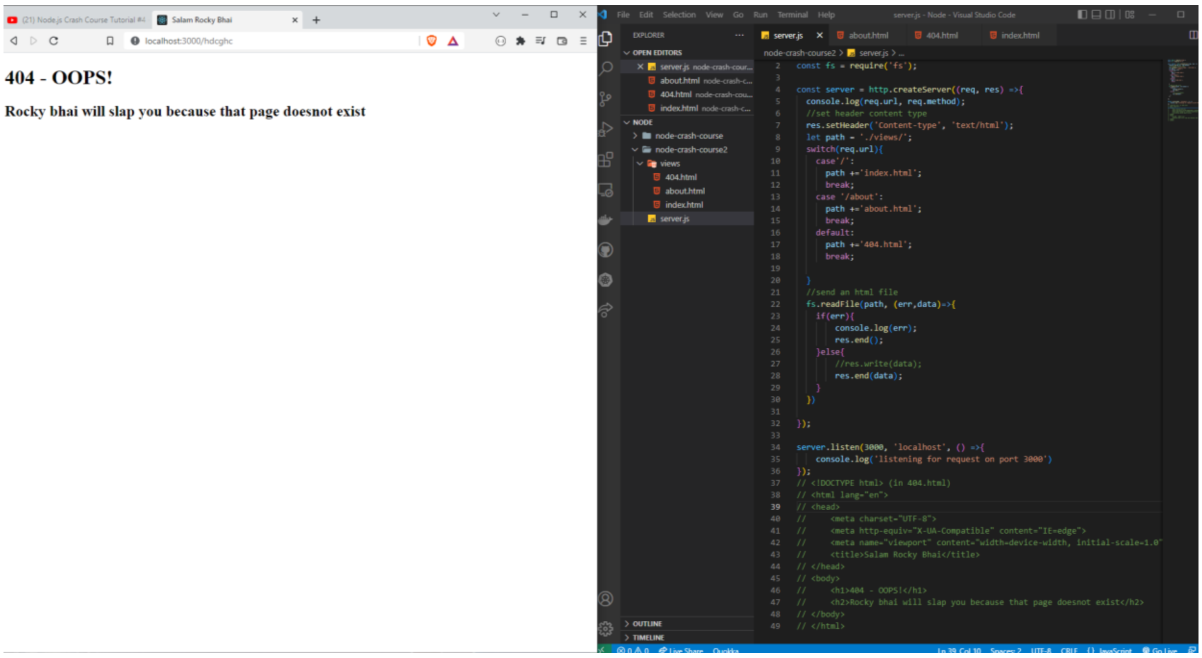
localhost:3000/about

About

Your path to becoming a Rocking way

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  let path = './views/';
  switch(req.url){
    case'/':
      path +='index.html';
      break;
    case '/about':
      path +='about.html';
      break;
    default:
      path +='404.html';
      break;

  }
  //send an html file
  fs.readFile(path, (err,data)=>{
    if(err){
        console.log(err);
        res.end();
    }else{
        //res.write(data);
        res.end(data);
    }
  })

});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
// <!DOCTYPE html> (in about.html)
// <html lang="en">
// <head>
//     <meta charset="UTF-8">
//     <meta http-equiv="X-UA-Compatible" content="IE=edge">
//     <meta name="viewport" content="width=device-width, initial-scale=1.0"
//     <title>Rocky Bhai</title>
// </head>
// <body>
//     <h1>About</h1>
//     <h2>Your path to becoming a Rocking way</h2>
// </body>
// </html>
```

404.html



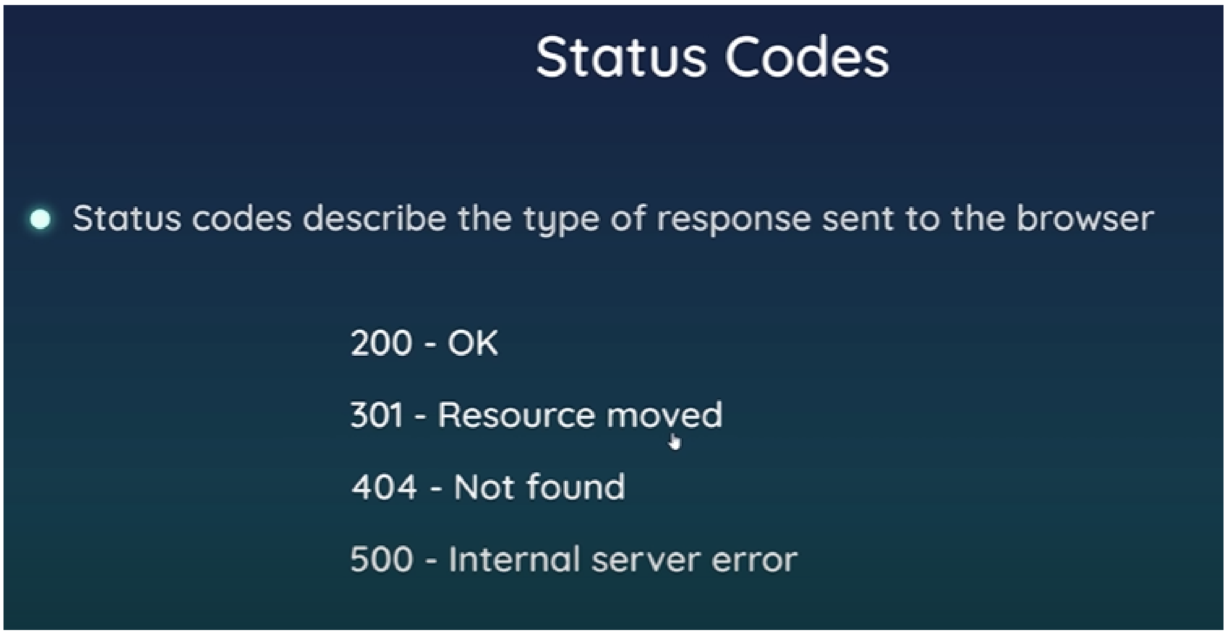
localhost:3000/hdcghc

404 - OOPS!

Rocky bhai will slap you because that page doesnot exist

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  let path = './views/';
  switch(req.url){
    case'/':
      path +='index.html';
      break;
    case '/about':
      path +='about.html';
      break;
    default:
      path +='404.html';
      break;

  }
  //send an html file
  fs.readFile(path, (err,data)=>{
    if(err){
        console.log(err);
        res.end();
    }else{
        //res.write(data);
        res.end(data);
    }
  })

});

server.listen(3000, 'localhost', () =>{
    console.log('listening for request on port 3000')
});
// <!DOCTYPE html> (in 404.html)
// <html lang="en">
// <head>
//     <meta charset="UTF-8">
//     <meta http-equiv="X-UA-Compatible" content="IE=edge">
//     <meta name="viewport" content="width=device-width, initial-scale=1.0"
//     <title>Salam Rocky Bhai</title>
// </head>
// <body>
//     <h1>404 - OOPS!</h1>
//     <h2>Rocky bhai will slap you because that page doesnot exist</h2>
// </body>
// </html>
```

Status Codes



Status Codes

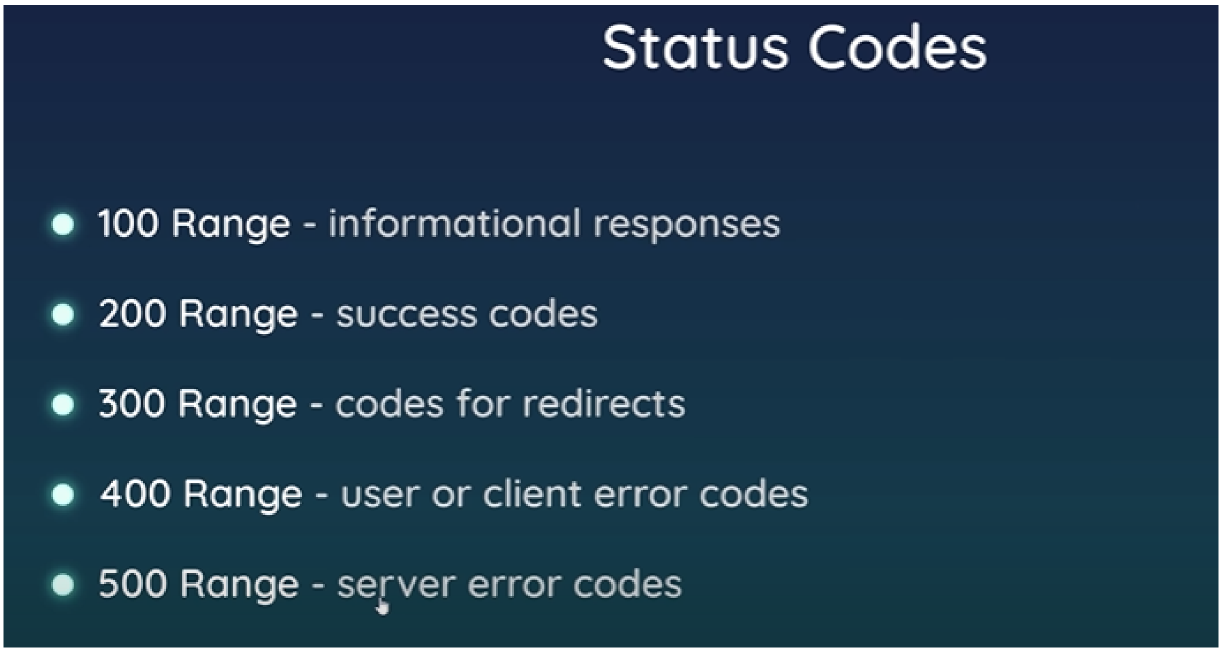
- Status codes describe the type of response sent to the browser

200 - OK

301 - Resource moved

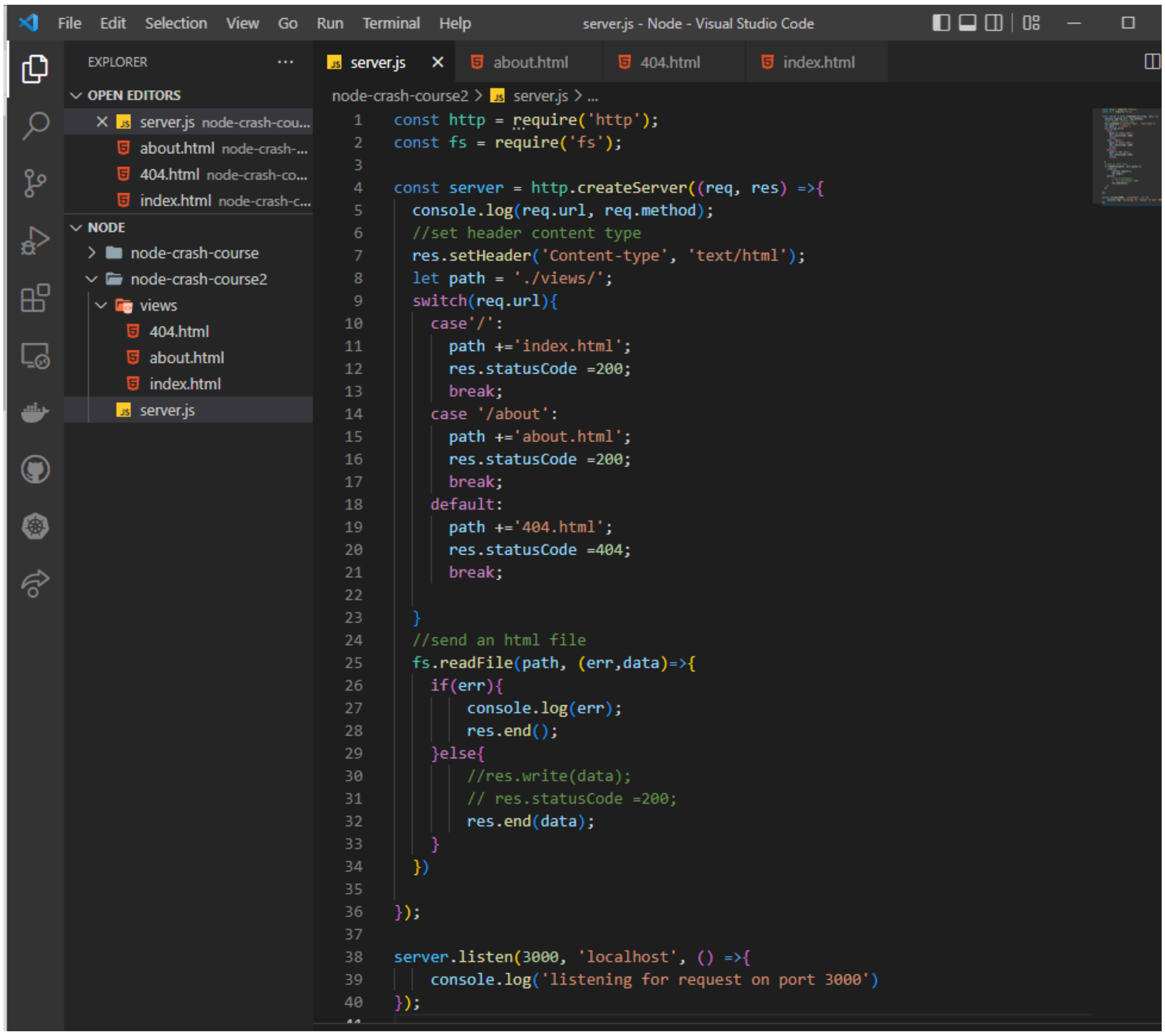
404 - Not found

500 - Internal server error



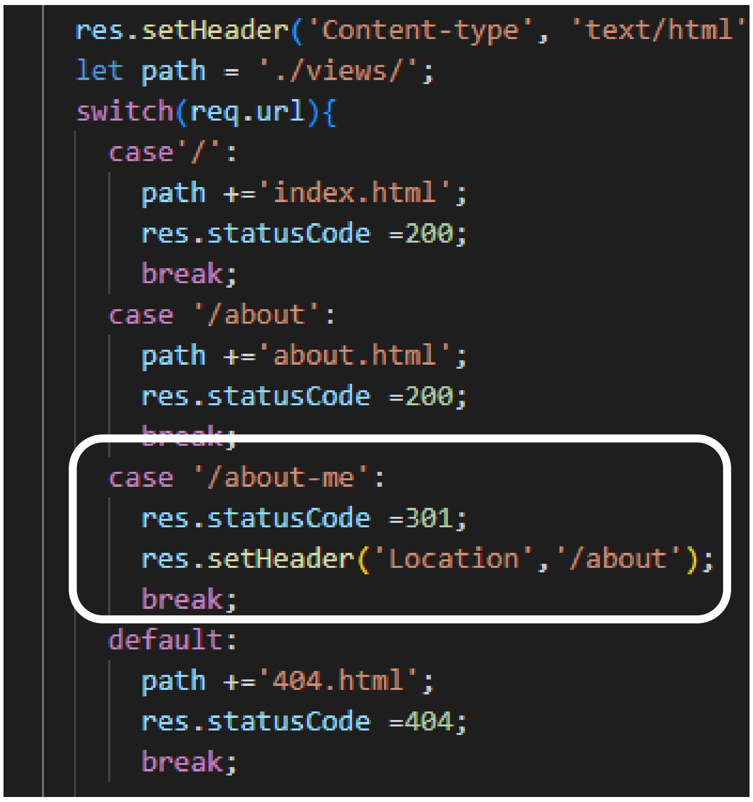
Status Codes

- 100 Range - informational responses
- 200 Range - success codes
- 300 Range - codes for redirects
- 400 Range - user or client error codes
- 500 Range - server error codes


File Edit Selection View Go Run Terminal Help
server.js - Node - Visual Studio Code
EXPLORER
OPEN EDITORS
server.js node-crash-cou...
about.html node-crash-...
404.html node-crash-co...
index.html node-crash-c...
NODE
node-crash-course
node-crash-course2
views
404.html
about.html
index.html
server.js
server.js
about.html
404.html
index.html
node-crash-course2 > server.js > ...

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

const server = http.createServer((req, res) =>{
  console.log(req.url, req.method);
  //set header content type
  res.setHeader('Content-type', 'text/html');
  let path = './views/';
  switch(req.url){
    case'/':
      path +='index.html';
      res.statusCode =200;
      break;
    case '/about':
      path +='about.html';
      res.statusCode =200;
      break;
    default:
      path +='404.html';
      res.statusCode =404;
      break;

  }
  //send an html file
  fs.readFile(path, (err,data)=>{
    if(err){
      console.log(err);
      res.end();
    }else{
      //res.write(data);
      // res.statusCode =200;
      res.end(data);
    }
  })

});

server.listen(3000, 'localhost', () =>{
  console.log('listening for request on port 3000')
});
```

Redirects



```
res.setHeader('Content-type', 'text/html'
let path = './views/';
switch(req.url){
  case'/':
    path +='index.html';
    res.statusCode =200;
    break;
  case '/about':
    path +='about.html';
    res.statusCode =200;
    break;
  case '/about-me':
    res.statusCode =301;
    res.setHeader('Location','/about');
    break;
  default:
    path +='404.html';
    res.statusCode =404;
    break;
```

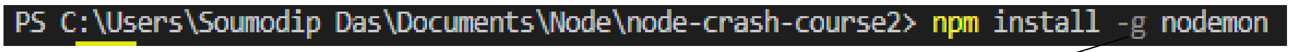
- **It redirects on a better way(about-me = about)**

<u>NPM = Node Package Manager</u>

- **Nodemon**

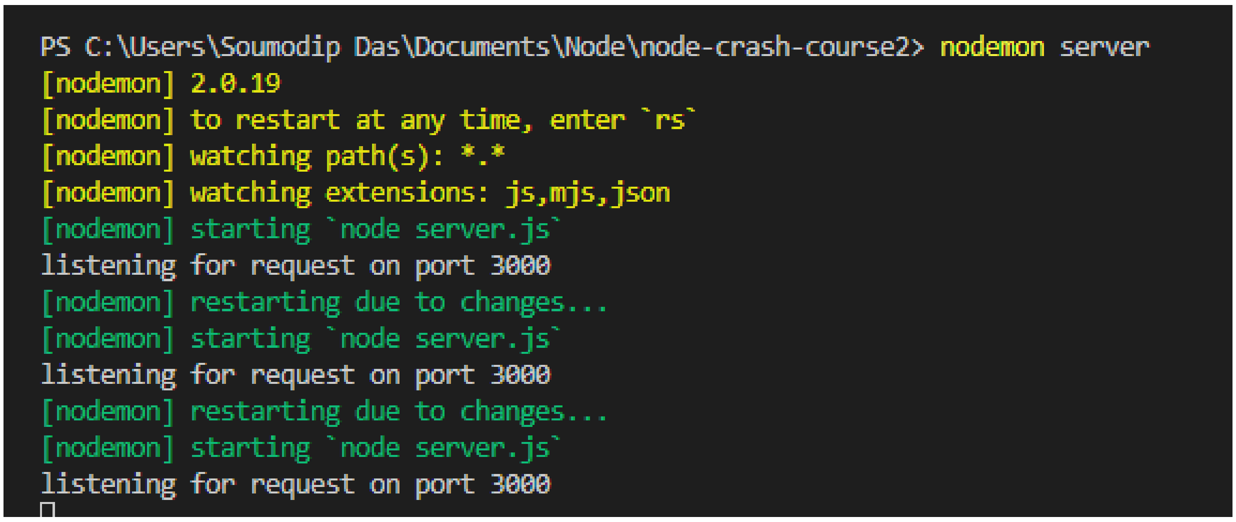
When we are changing anything in code then we need to restart the servers manually, Nodemon help us to combat that by automatically restarting the server whenever changes are made and saved to our file.

So we have to install Nodemon.



```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> npm install -g nodemon
```

For install Nodemon in our system globally



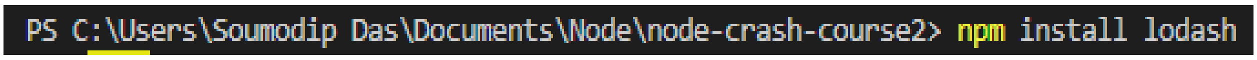
```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> nodemon server
[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
listening for request on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
listening for request on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
listening for request on port 3000
```

It manually restart the server when we change the code.

<u>Package.json File</u>

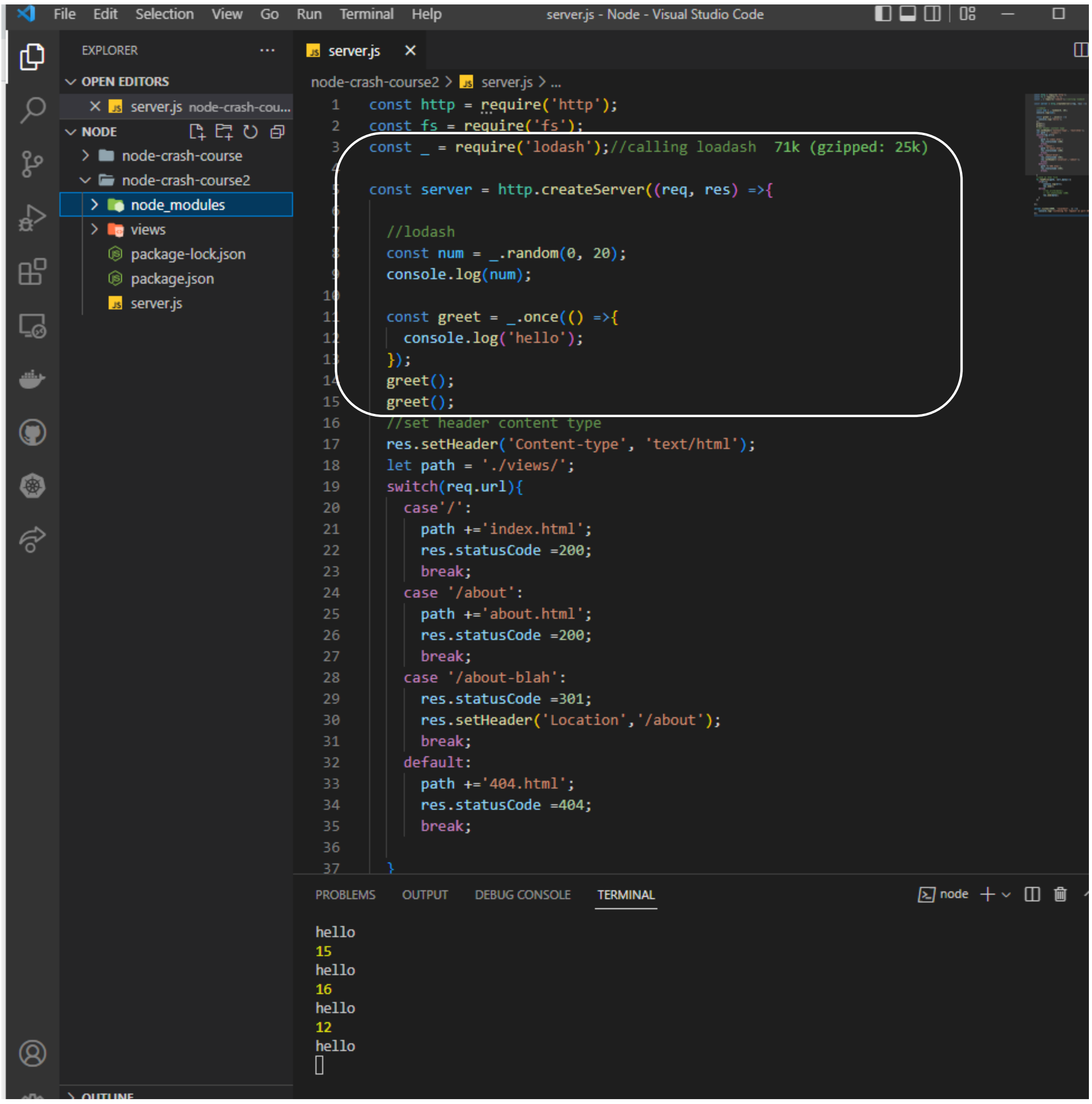
- **npm init for installing package.json file**

<u>Installing package locally</u>



```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course2> npm install lodash
```

- **Installing lodash on system**



```javascript
const http = require('http');
const fs = require('fs');
const _ = require('lodash');//calling loadash  71k (gzipped: 25k)

const server = http.createServer((req, res) =>{

  //lodash
  const num = _.random(0, 20);
  console.log(num);

  const greet = _.once(() =>{
    console.log('hello');
  });
  greet();
  greet();
  //set header content type
  res.setHeader('Content-type', 'text/html');
  let path = './views/';
  switch(req.url){
    case'/':
      path +='index.html';
      res.statusCode =200;
      break;
    case '/about':
      path +='about.html';
      res.statusCode =200;
      break;
    case '/about-blah':
      res.statusCode =301;
      res.setHeader('Location','/about');
      break;
    default:
      path +='404.html';
      res.statusCode =404;
      break;

  }
```

```
hello
15
hello
16
hello
12
hello
```

Dependencies

- **npm install – for installing node modules**
- **npm install -g nodemon – for installing nodemon**
- **npm init – for installing package.json file**