

Handling Requests and Responses the GET Request

localhost:3000 is simply the same as the home page of any website like www.google.com

localhost:3000/ is the rit of the home page.

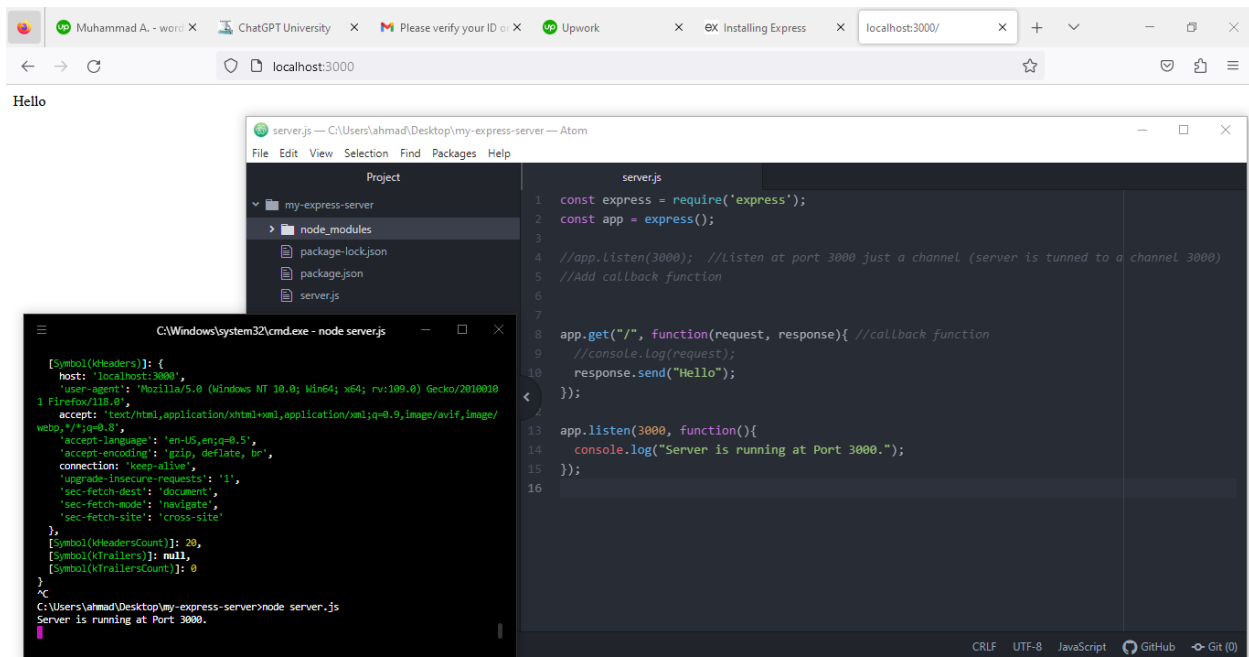
Like when we search www.google.com/ it goes to the home page.

<http://localhost:3000/> in google chrome

see lecture **3. Handling Requests and Responses the GET Request**

Now change send response back to the request, refresh the browser page and see the changes.

Note that the green data shown below comes when we change the js code in server while running it.

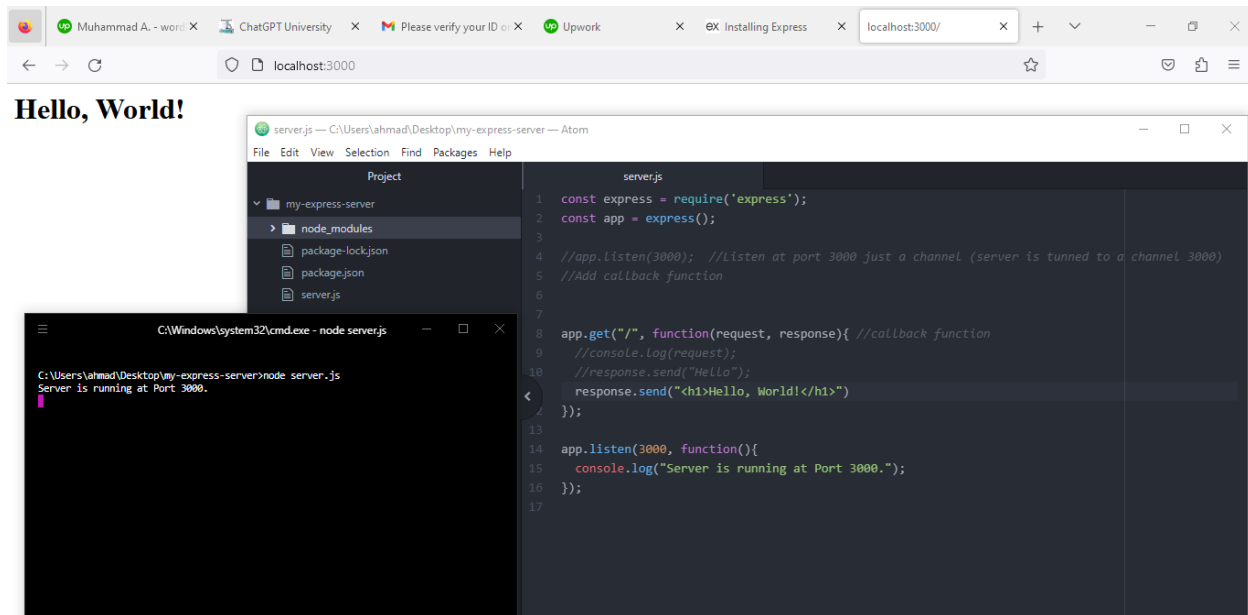


The screenshot shows a web browser window at localhost:3000 displaying the word "Hello". Below the browser, an Atom code editor shows the server.js file with the following code:

```
1 const express = require('express');
2 const app = express();
3
4 //app.listen(3000); //Listen at port 3000 just a channel (server is tunned to a channel 3000)
5 //Add callback function
6
7
8 app.get("/", function(request, response){ //callback function
9   //console.log(request);
10  response.send("Hello");
11 });
12
13 app.listen(3000, function(){
14   console.log("Server is running at Port 3000.");
15 });
16
```

Below the code editor, a terminal window shows the command prompt output:

```
C:\Windows\system32\cmd.exe - node server.js
[Symbol(kHeaders)]: {
  host: 'localhost:3000',
  'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0',
  accept: 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8',
  'accept-language': 'en-US,en;q=0.5',
  'accept-encoding': 'gzip, deflate, br',
  connection: 'keep-alive',
  'upgrade-insecure-requests': '1',
  'sec-fetch-dest': 'document',
  'sec-fetch-mode': 'navigate',
  'sec-fetch-site': 'cross-site'
},
[Symbol(kHeadersCount)]: 20,
[Symbol(kTrailers)]: null,
[Symbol(kTrailersCount)]: 0
}
^C
C:\Users\ahmad\Desktop\my-express-server>node server.js
Server is running at Port 3000.
```



```
//jshint esversion:6
```

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

```
const app = express();
```

```
app.get('/',function(request,response){ //simply write re for request and res for response
```

```
  //console.log(request);
```

```
  response.send("<h1>Hello, World!</h1>");
```

```
}); //this method is present in express and allows what should happen when
```

```
//browser gets touch with the server @nd parameter is call back wunction
```

```
//app.listen(3000); //server is created
```

```
app.listen(3000, function(){
```

```
  console.log("Server started on port 3000");
```

```
});
```