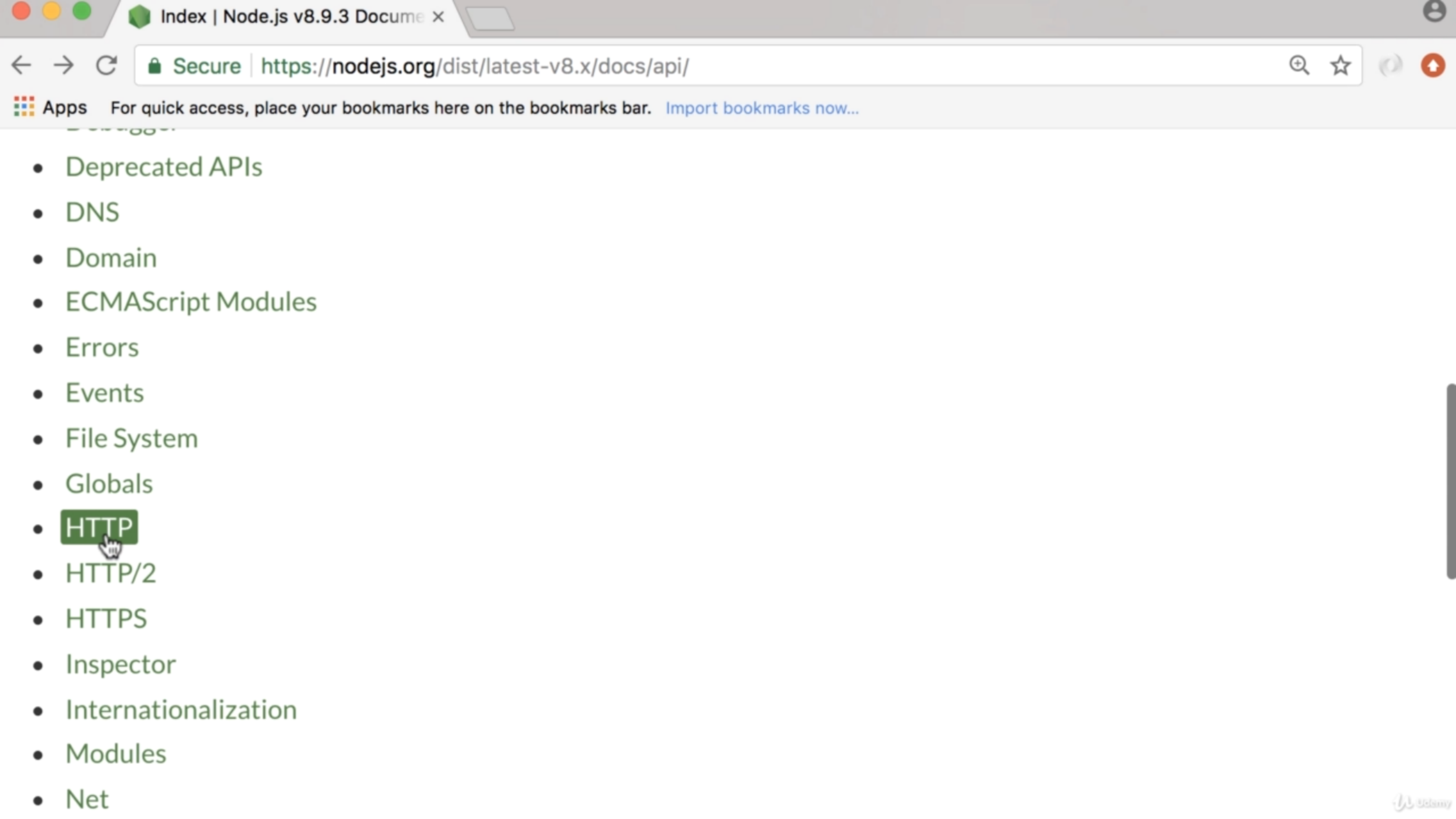
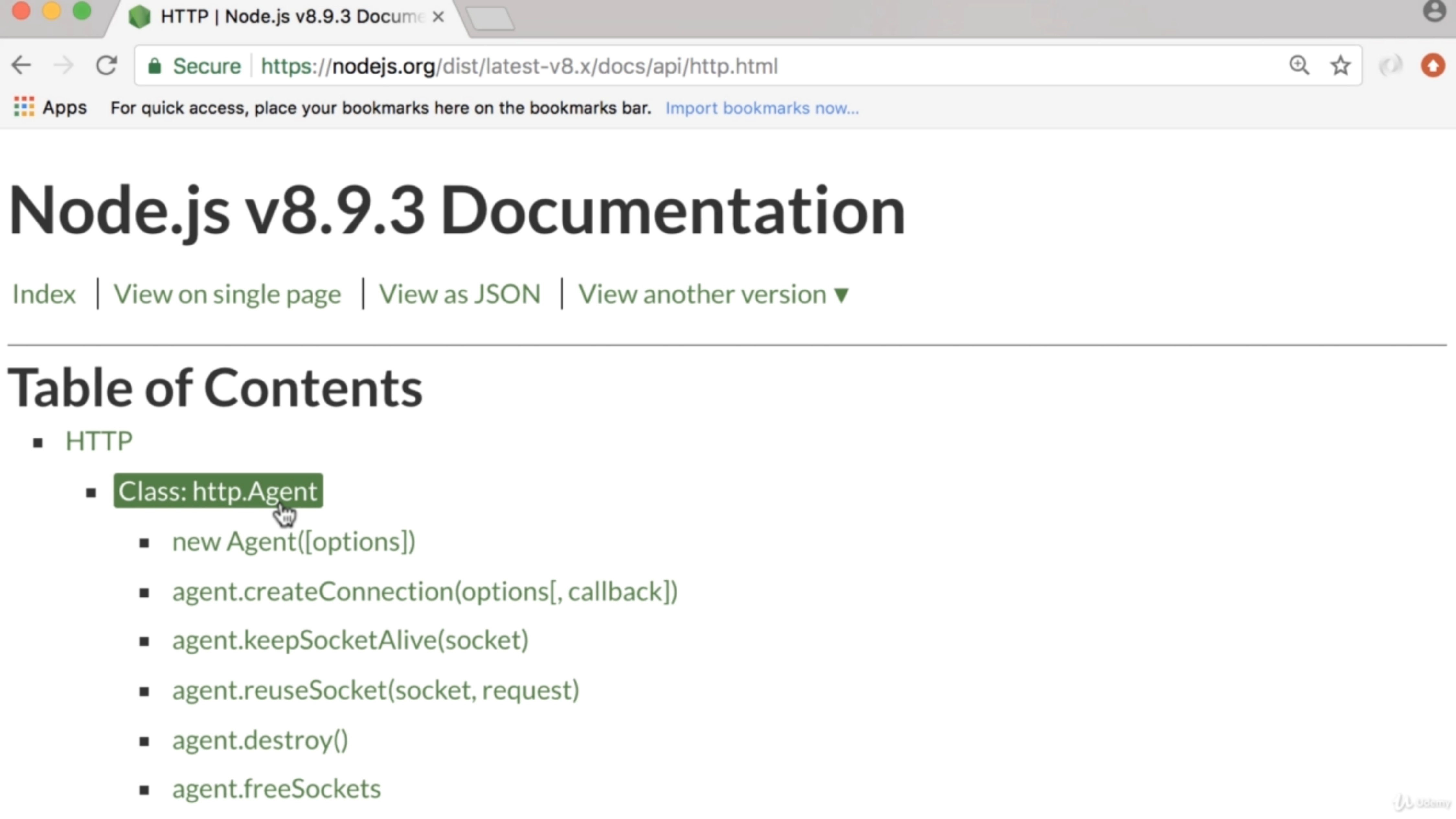
HTTP Module

Mosh Hamedani

programmingwithmosh.com


Index | Node.js v8.9.3 Docume
Secure
https://nodejs.org/dist/latest-v8.x/docs/api/
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

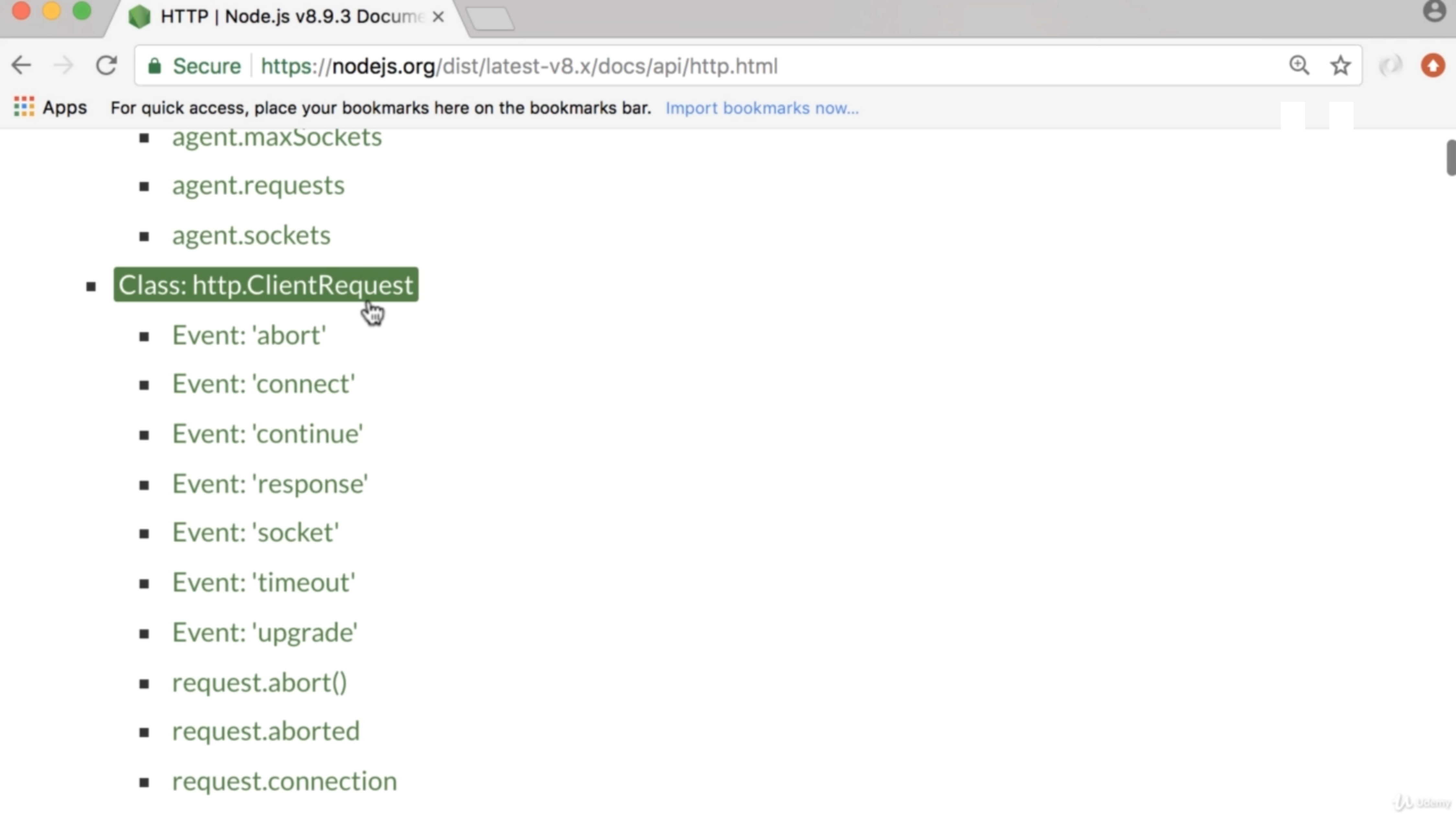
- Debugger
- Deprecated APIs
- DNS
- Domain
- ECMAScript Modules
- Errors
- Events
- File System
- Globals
- HTTP
- HTTP/2
- HTTPS
- Inspector
- Internationalization
- Modules
- Net


HTTP | Node.js v8.9.3 Docume
Secure
https://nodejs.org/dist/latest-v8.x/docs/api/http.html
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

Node.js v8.9.3 Documentation

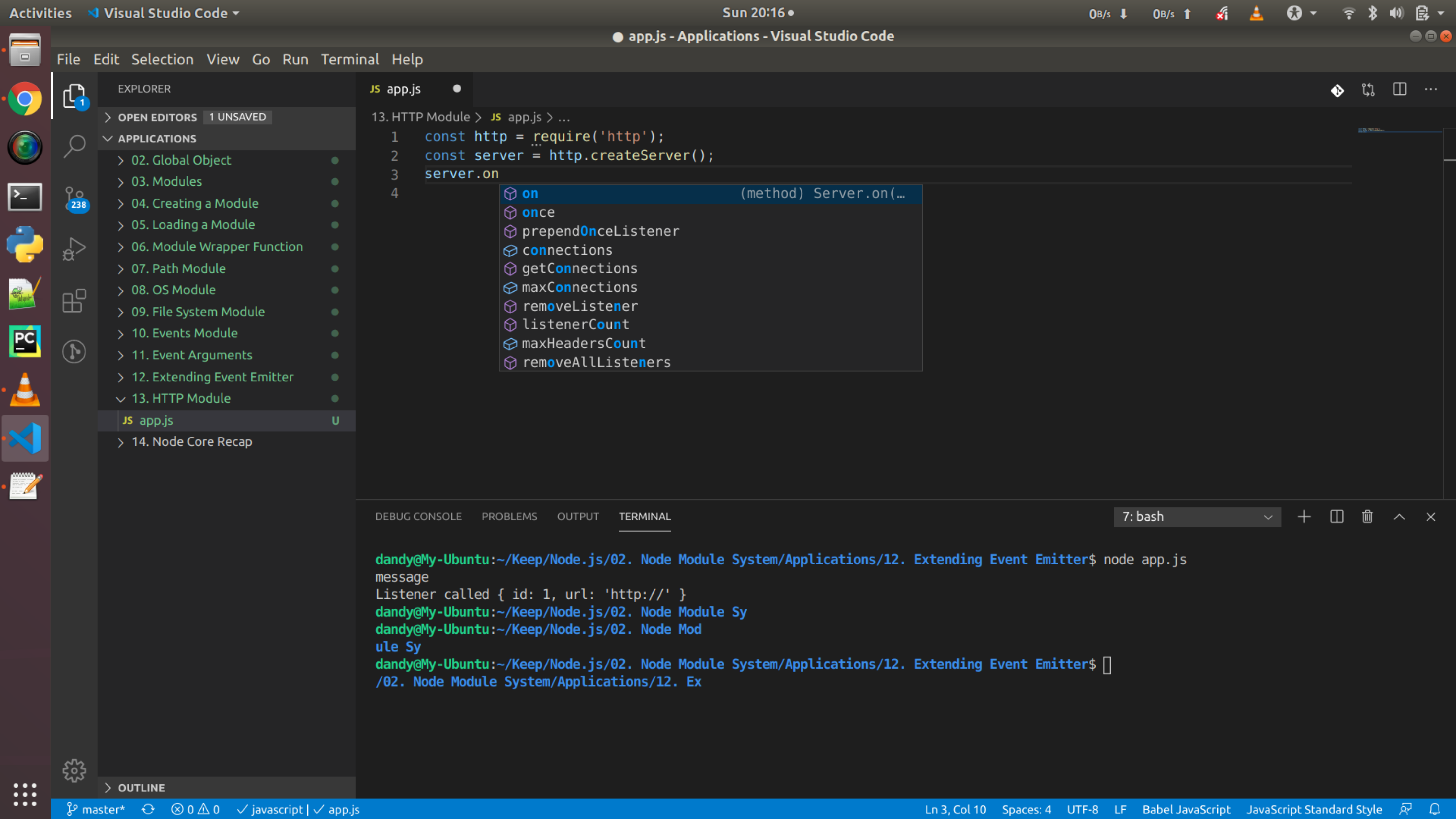
Index | View on single page | View as JSON | View another version ▼

Table of Contents


HTTP | Node.js v8.9.3 Docume
Secure
https://nodejs.org/dist/latest-v8.x/docs/api/http.html
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

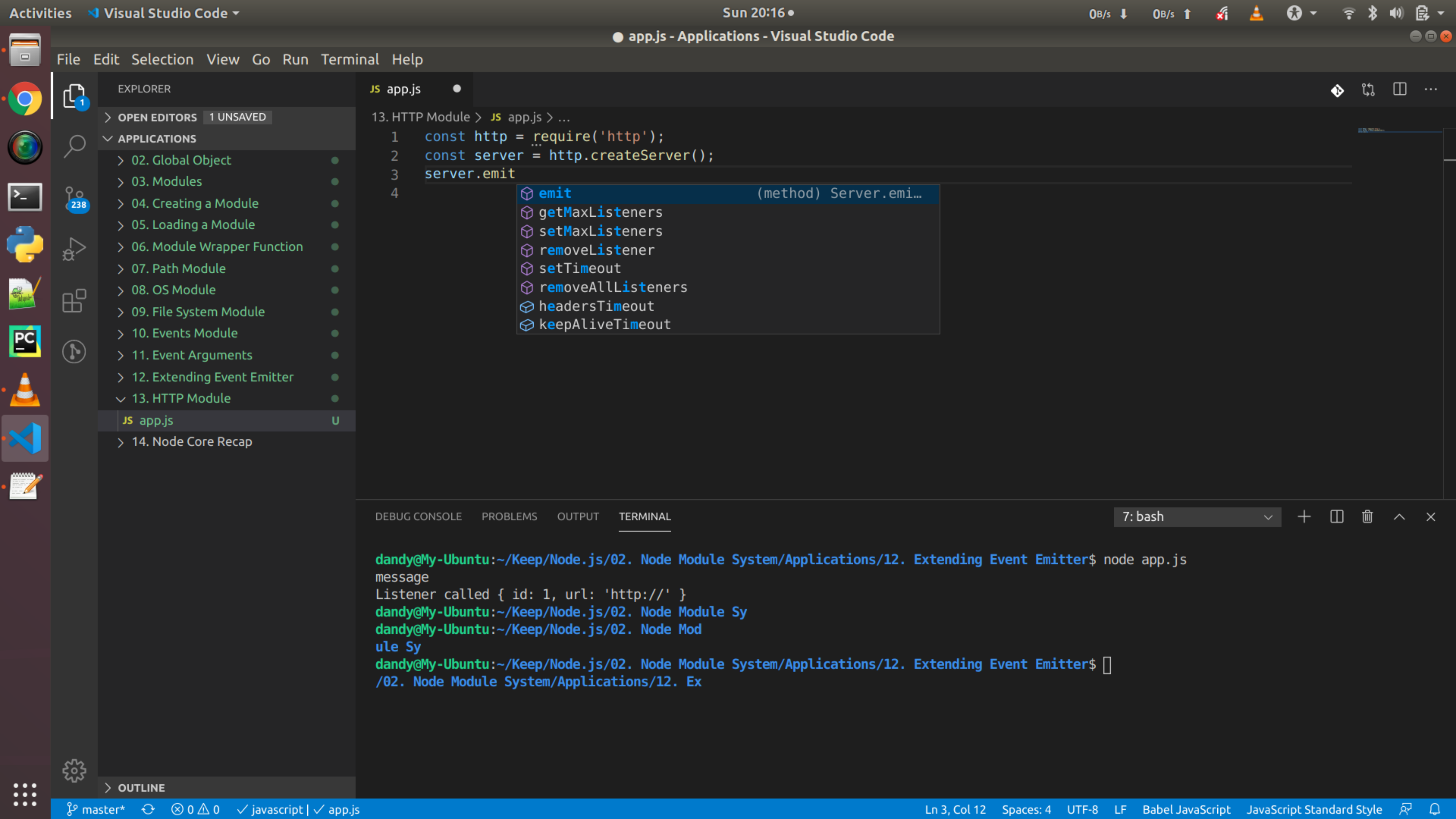
- agent.maxSockets
- agent.requests
- agent.sockets

- Class: http.ClientRequest
  - Event: 'abort'
  - Event: 'connect'
  - Event: 'continue'
  - Event: 'response'
  - Event: 'socket'
  - Event: 'timeout'
  - Event: 'upgrade'
  - request.abort()
  - request.aborted
  - request.connection



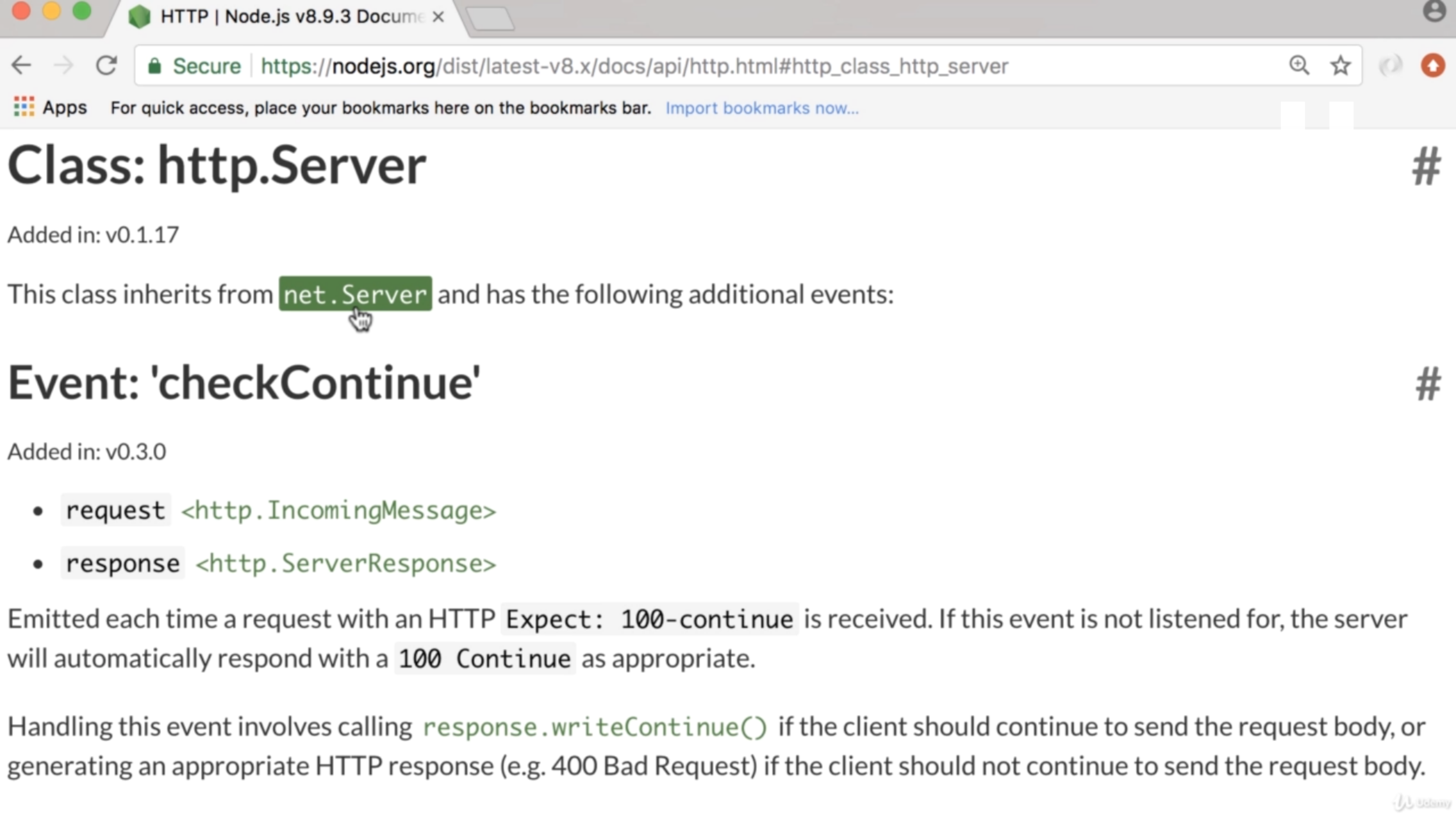
```
const http = require('http');
const server = http.createServer();
server.on
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/12. Extending Event Emitter$ node app.js
message
Listener called { id: 1, url: 'http://' }
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module Sy
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Mod
ule Sy
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/12. Extending Event Emitter$ 
/02. Node Module System/Applications/12. Ex
```



```
const http = require('http');
const server = http.createServer();
server.emit
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/12. Extending Event Emitter$ node app.js
message
Listener called { id: 1, url: 'http://' }
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module Sy
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Mod
ule Sy
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/12. Extending Event Emitter$ 
/02. Node Module System/Applications/12. Ex
```


HTTP | Node.js v8.9.3 Docume
Secure
https://nodejs.org/dist/latest-v8.x/docs/api/http.html#http_class_http_server
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

Class: http.Server

Added in: v0.1.17

This class inherits from `net.Server` and has the following additional events:

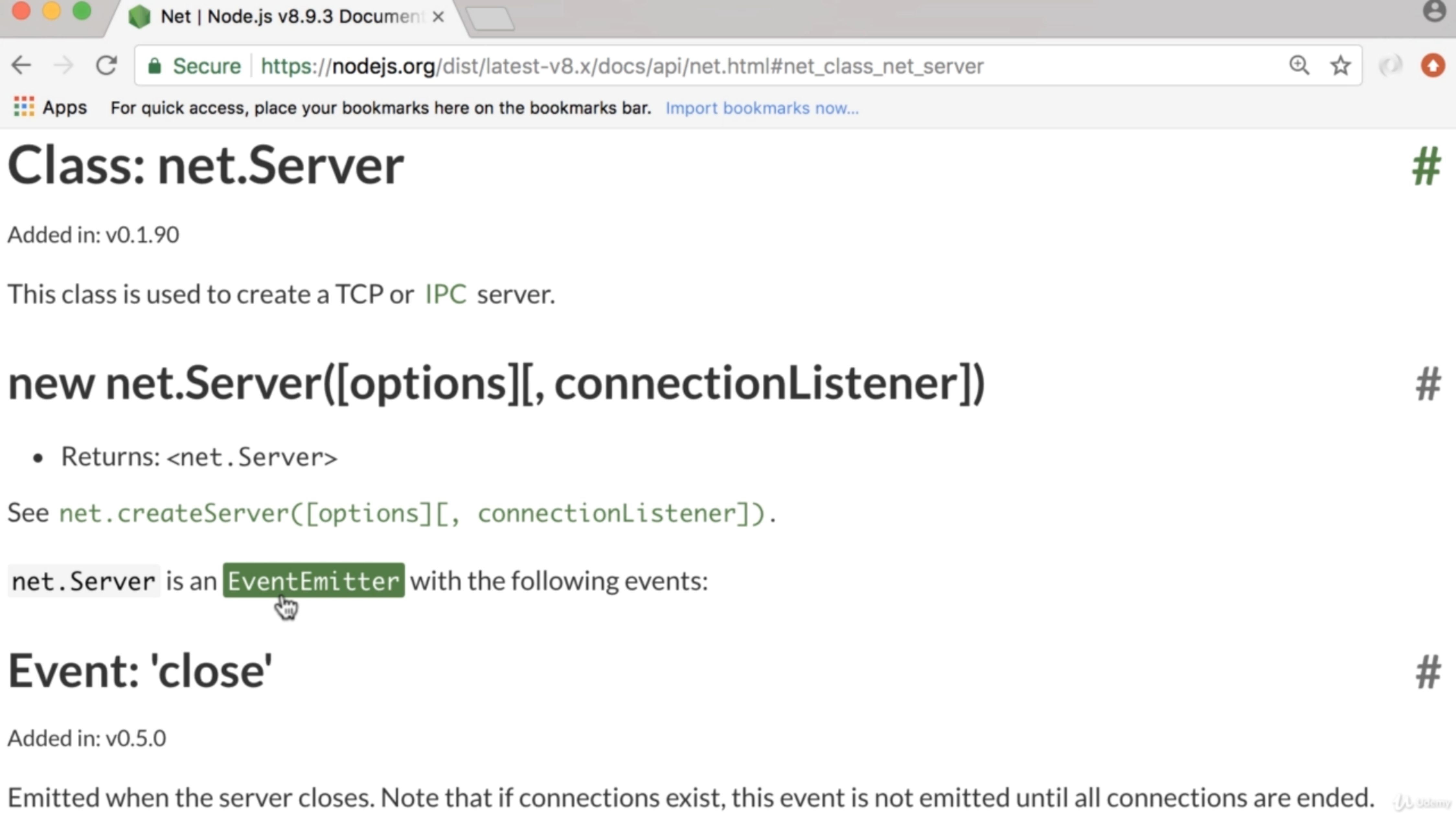
Event: 'checkContinue'

Added in: v0.3.0

- `request` <http.IncomingMessage>
- `response` <http.ServerResponse>

Emitted each time a request with an HTTP `Expect: 100-continue` is received. If this event is not listened for, the server will automatically respond with a `100 Continue` as appropriate.

Handling this event involves calling `response.writeContinue()` if the client should continue to send the request body, or generating an appropriate HTTP response (e.g. 400 Bad Request) if the client should not continue to send the request body.



Net | Node.js v8.9.3 Documen
Secure
https://nodejs.org/dist/latest-v8.x/docs/api/net.html#net_class_net_server
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...

Class: net.Server

Added in: v0.1.90

This class is used to create a TCP or IPC server.

new net.Server([options][, connectionListener])

- Returns: <net.Server>

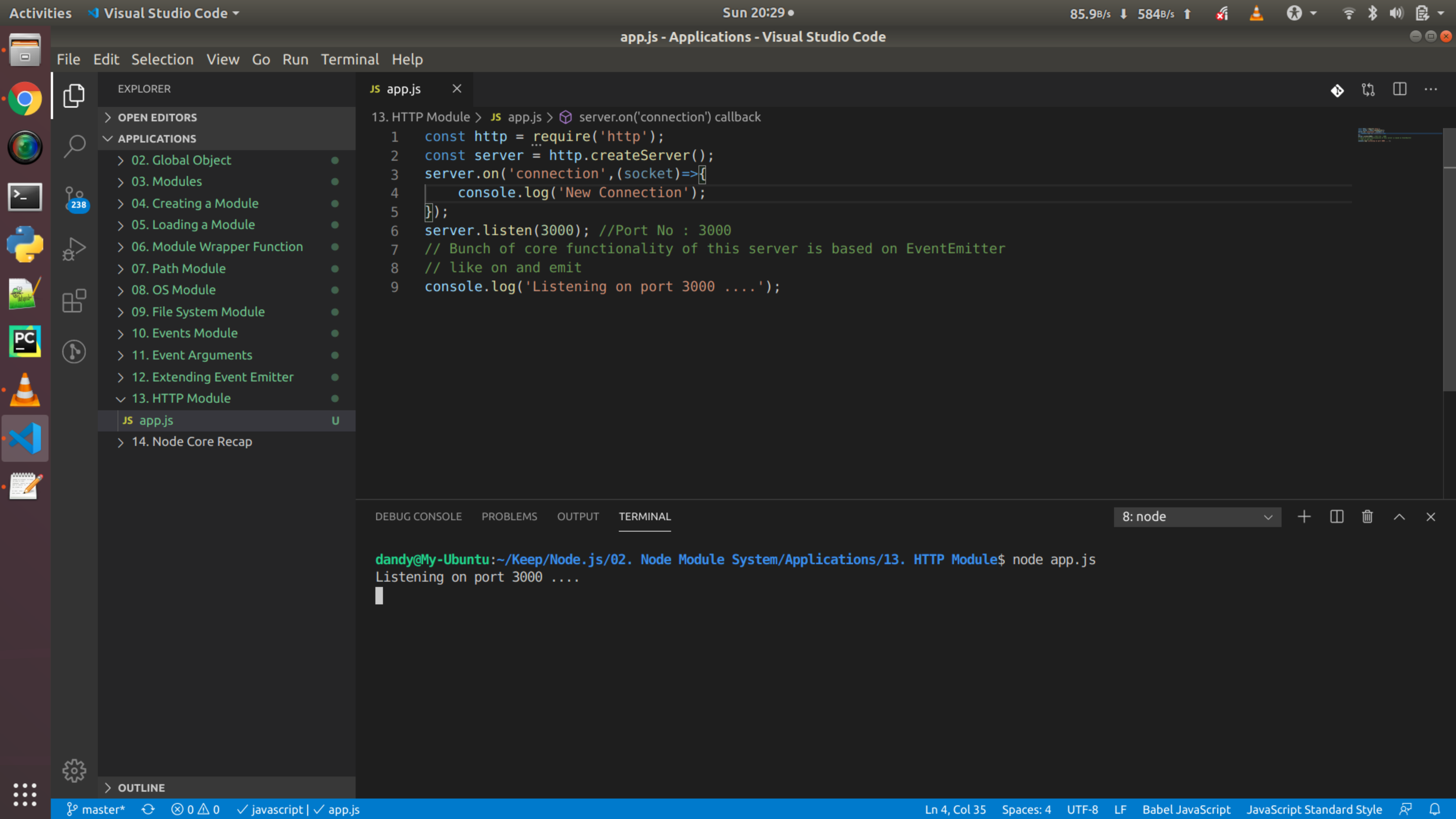
See `net.createServer([options][, connectionListener])`.

`net.Server` is an `EventEmitter` with the following events:

Event: 'close'

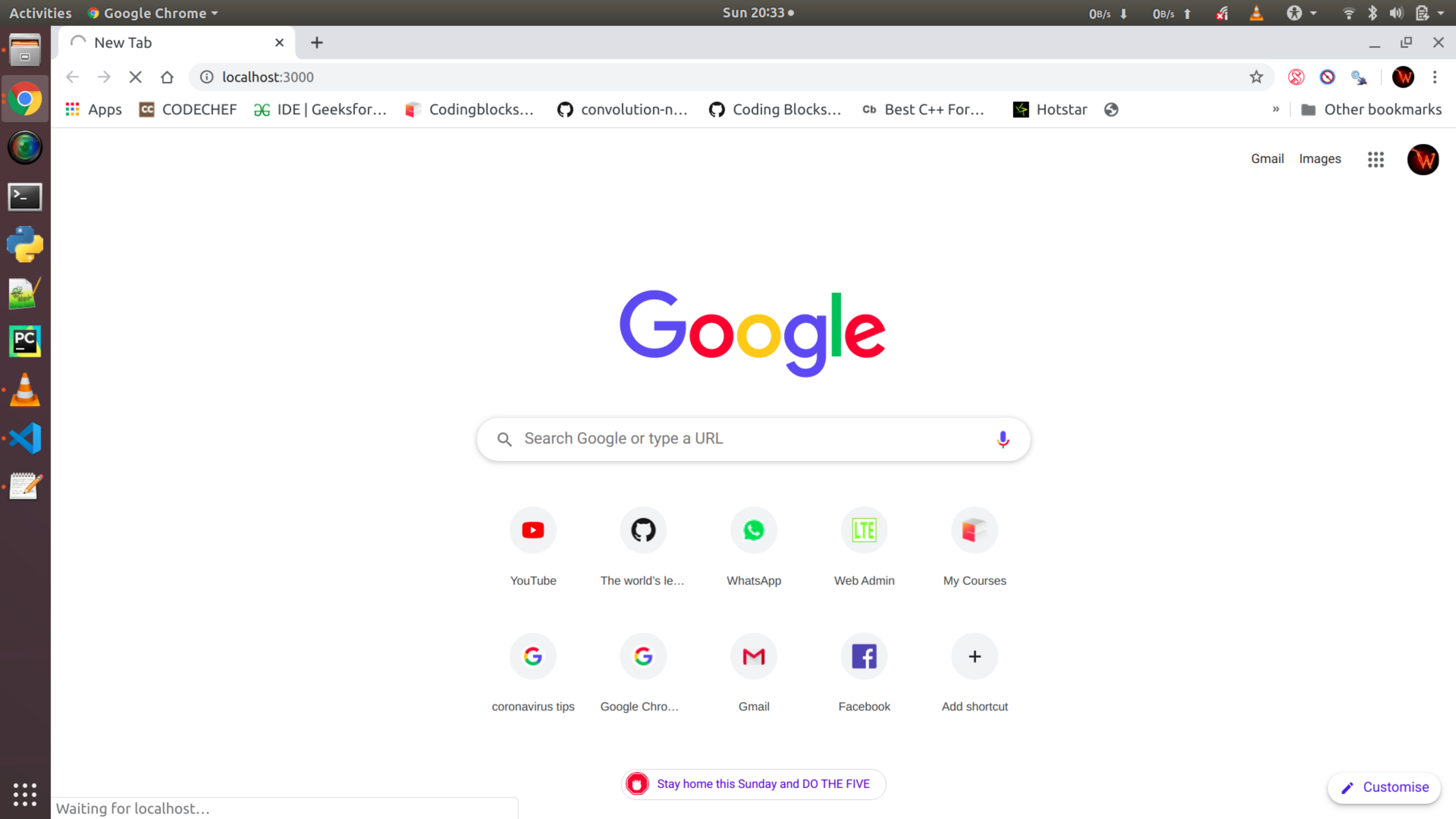
Added in: v0.5.0

Emitted when the server closes. Note that if connections exist, this event is not emitted until all connections are ended.

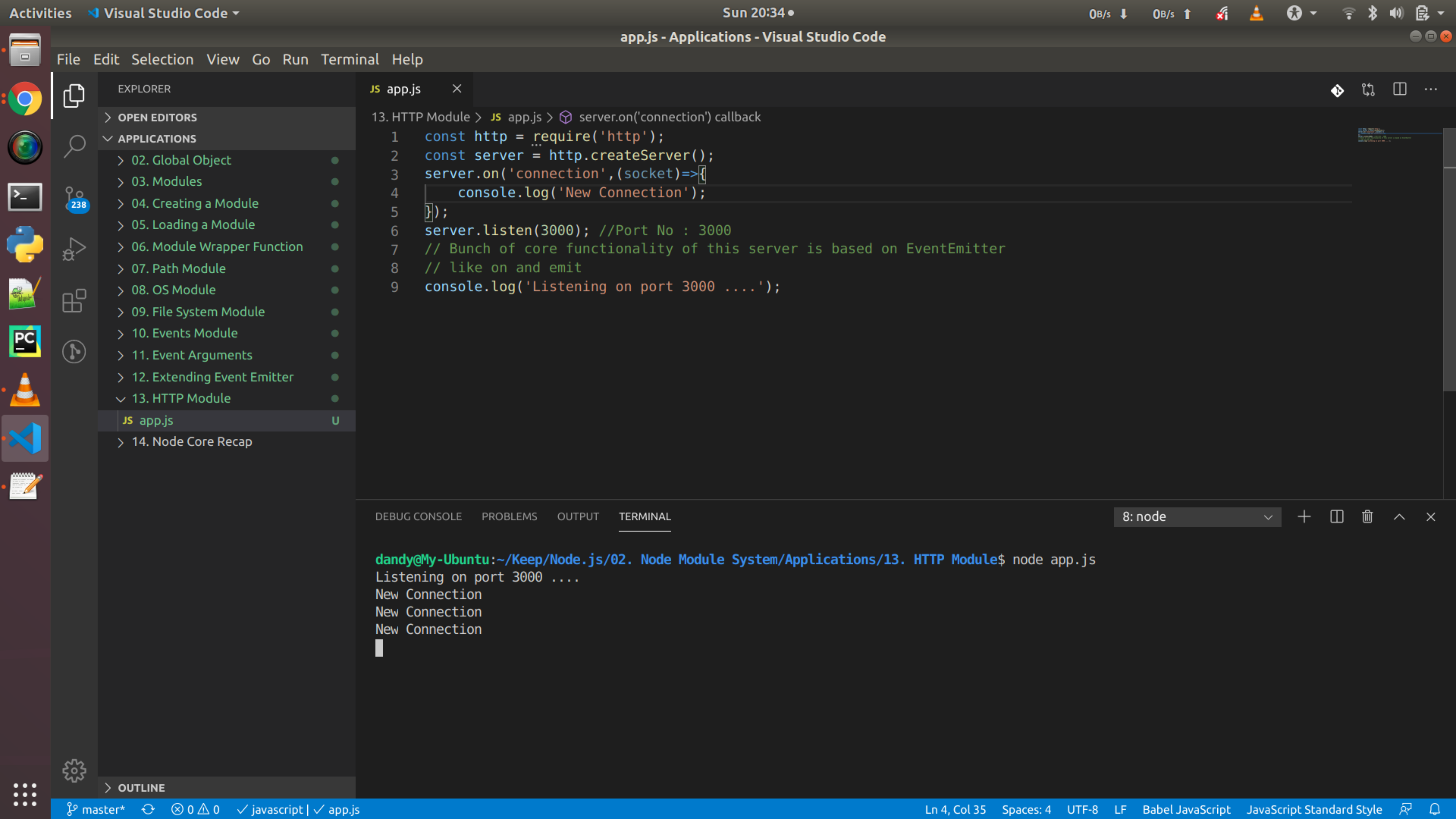


```javascript
const http = require('http');
const server = http.createServer();
server.on('connection',(socket)=>{
    console.log('New Connection');
});
server.listen(3000); //Port No : 3000
// Bunch of core functionality of this server is based on EventEmitter
// like on and emit
console.log('Listening on port 3000 ....');
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
```

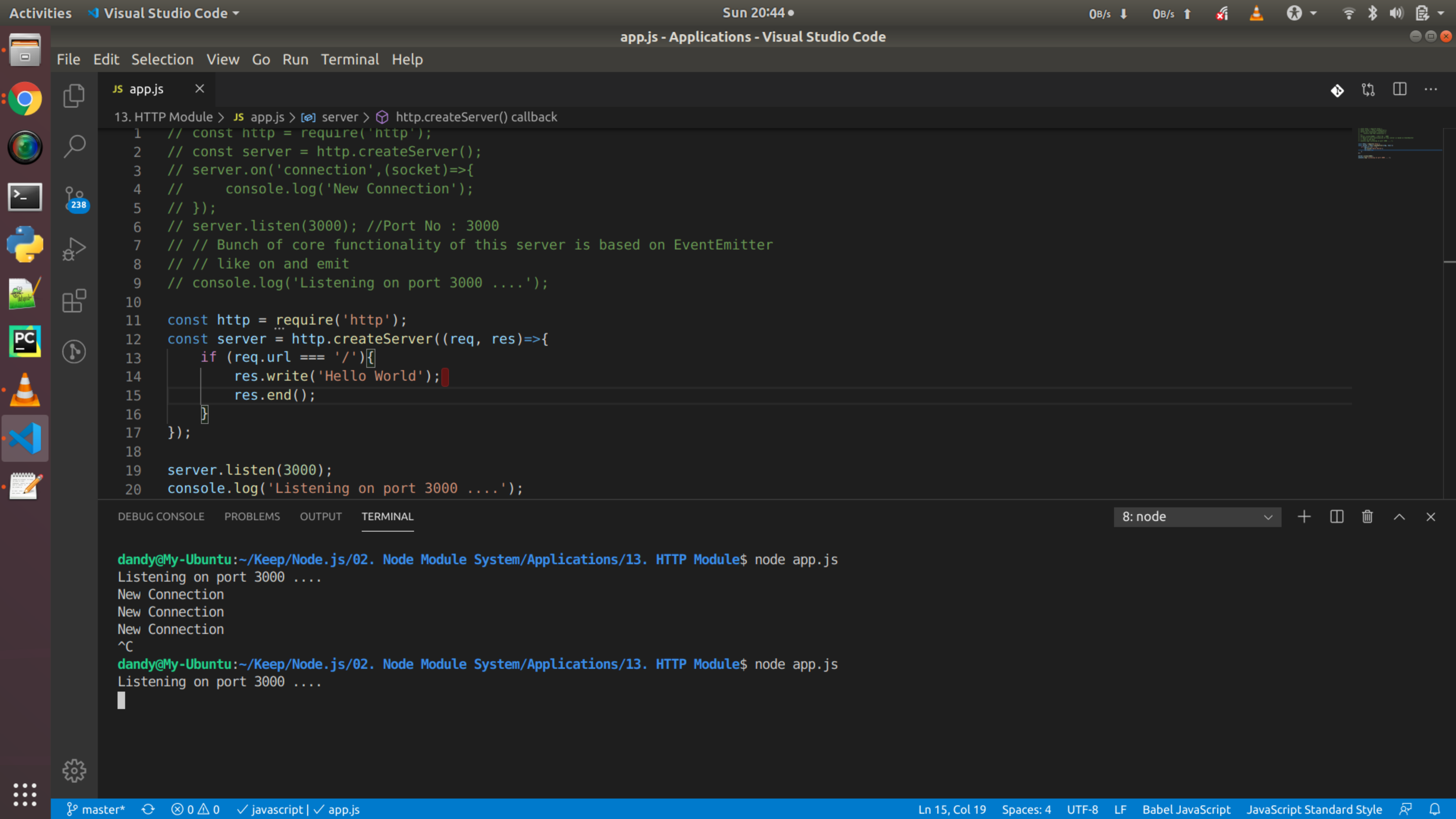


Activities
Google Chrome
Sun 20:33
0B/s
0B/s
New Tab
localhost:3000
Apps
CODECHEF
IDE | Geeksfor...
Codingblocks...
convolution-n...
Coding Blocks...
Best C++ For...
Hotstar
Other bookmarks
Gmail
Images
Google
Search Google or type a URL
YouTube
The world's le...
WhatsApp
Web Admin
My Courses
coronavirus tips
Google Chro...
Gmail
Facebook
Add shortcut
Stay home this Sunday and DO THE FIVE
Customise
Waiting for localhost...



```
const http = require('http');
const server = http.createServer();
server.on('connection',(socket)=>{
    console.log('New Connection');
});
server.listen(3000); //Port No : 3000
// Bunch of core functionality of this server is based on EventEmitter
// like on and emit
console.log('Listening on port 3000 ....');
```

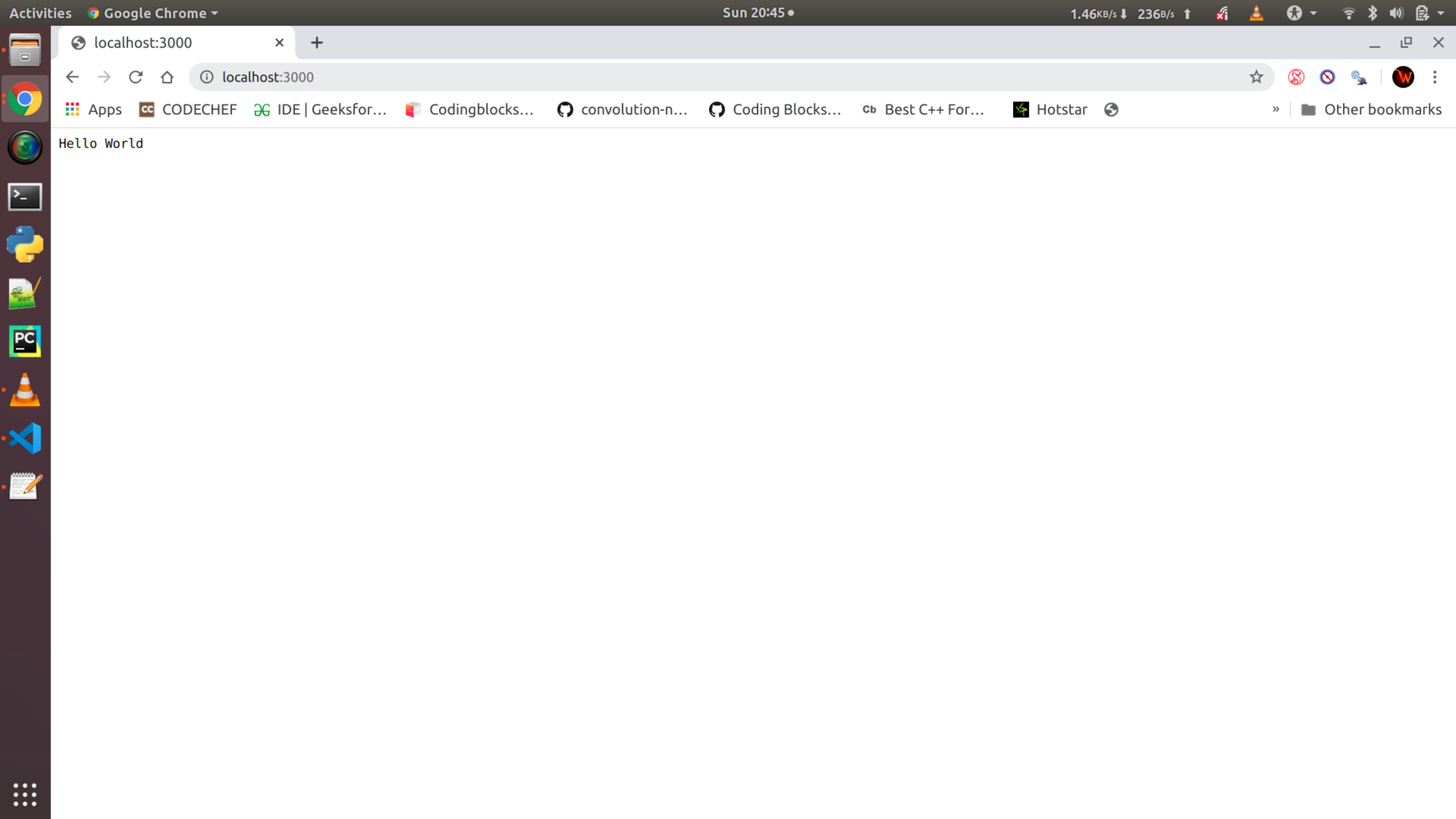
```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
New Connection
New Connection
New Connection
```



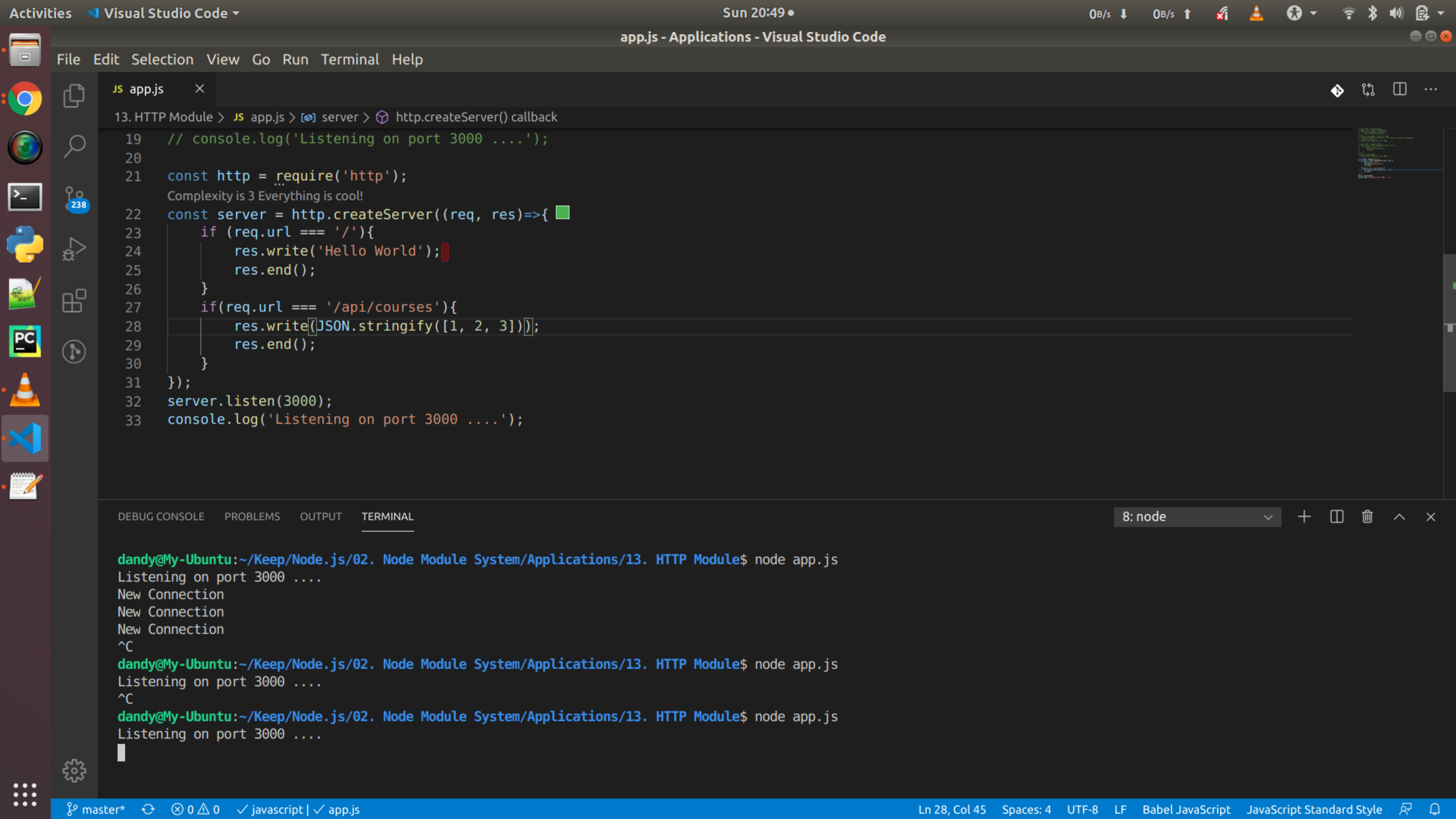
Activities
Visual Studio Code
Sun 20:44
app.js - Applications - Visual Studio Code
File Edit Selection View Go Run Terminal Help
app.js
13. HTTP Module > app.js > server > http.createServer() callback
238
DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL
8: node
master* ⊗ 0 ⚠ 0 ✓ javascript | ✓ app.js
Ln 15, Col 19 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style

```
// const http = require('http');
// const server = http.createServer();
// server.on('connection',(socket)=>{
//     console.log('New Connection');
// });
// server.listen(3000); //Port No : 3000
// // Bunch of core functionality of this server is based on EventEmitter
// // like on and emit
// console.log('Listening on port 3000 ....');

const http = require('http');
const server = http.createServer((req, res)=>{
    if (req.url === '/'){
        res.write('Hello World');
        res.end();
    }
});

server.listen(3000);
console.log('Listening on port 3000 ....');
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
New Connection
New Connection
New Connection
^C
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
```

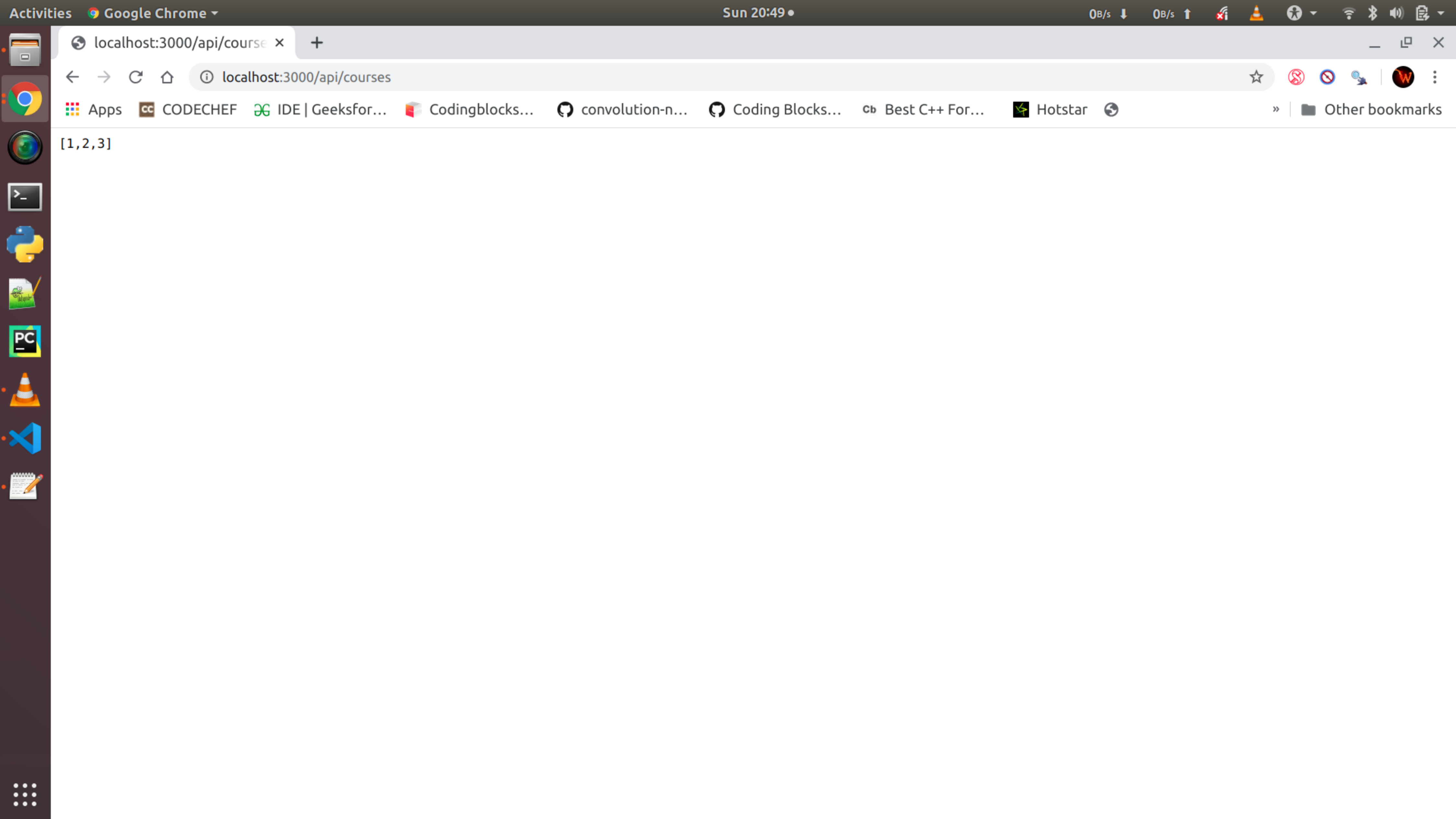

Activities
Google Chrome
Sun 20:45
1.46KB/s
236B/s
localhost:3000
localhost:3000
Apps
CODECHEF
IDE | Geeksfor...
Codingblocks...
convolution-n...
Coding Blocks...
Best C++ For...
Hotstar
Other bookmarks

```
Hello World
```



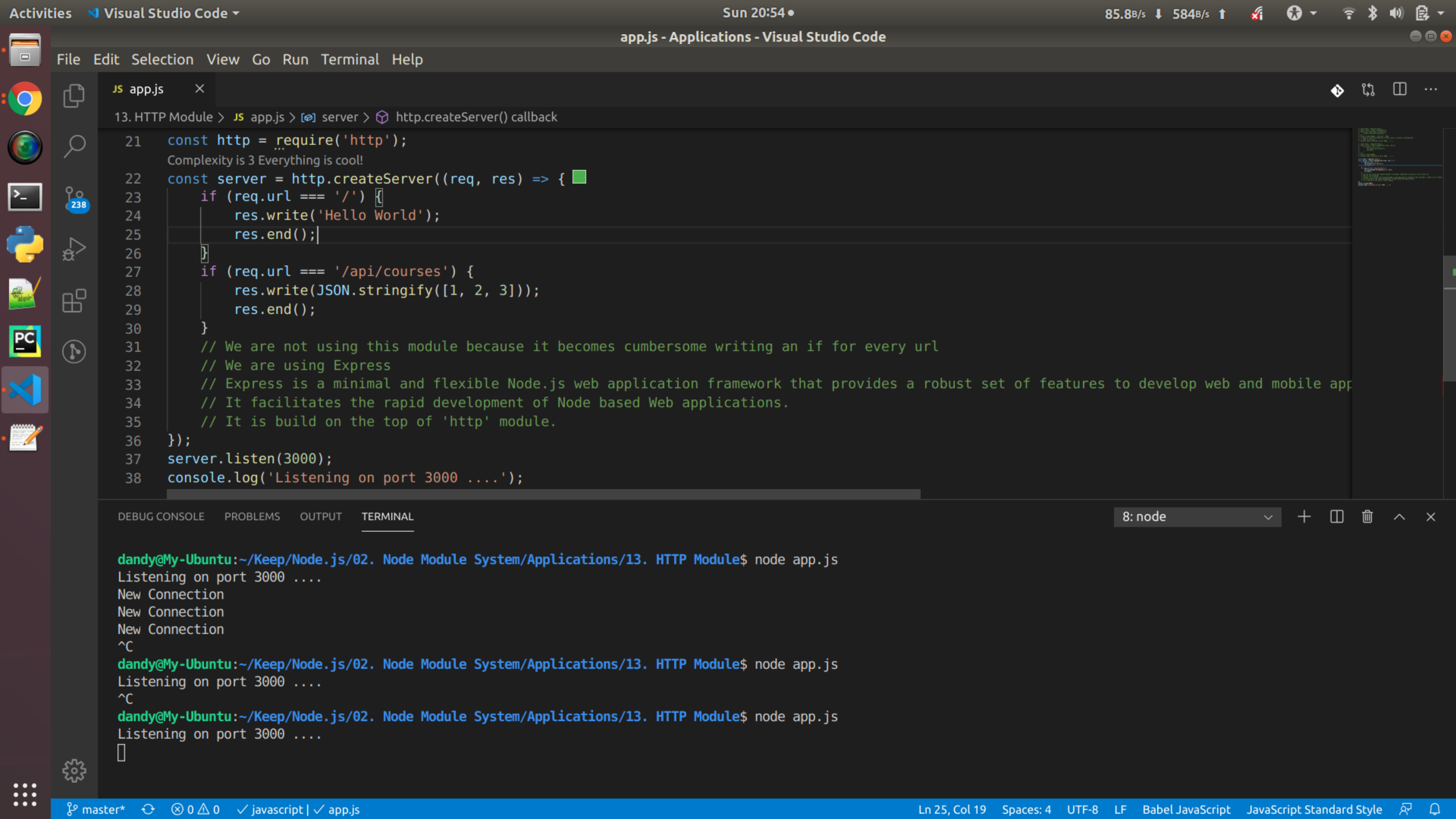
```
// console.log('Listening on port 3000 ....');

const http = require('http');
const server = http.createServer((req, res)=>{
    if (req.url === '/'){
        res.write('Hello World');
        res.end();
    }
    if(req.url === '/api/courses'){
        res.write(JSON.stringify([1, 2, 3]));
        res.end();
    }
});
server.listen(3000);
console.log('Listening on port 3000 ....');
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
New Connection
New Connection
New Connection
^C
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
^C
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
```



Activities
Google Chrome
Sun 20:49
0B/s
0B/s
localhost:3000/api/course
localhost:3000/api/courses
Apps
CODECHEF
IDE | Geeksfor...
Codingblocks...
convolution-n...
Coding Blocks...
Best C++ For...
Hotstar
Other bookmarks

```
[1,2,3]
```



```
const http = require('http');
const server = http.createServer((req, res) => {
    if (req.url === '/') {
        res.write('Hello World');
        res.end();
    }
    if (req.url === '/api/courses') {
        res.write(JSON.stringify([1, 2, 3]));
        res.end();
    }
    // We are not using this module because it becomes cumbersome writing an if for every url
    // We are using Express
    // Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile app
    // It facilitates the rapid development of Node based Web applications.
    // It is build on the top of 'http' module.
});
server.listen(3000);
console.log('Listening on port 3000 ....');
```

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
New Connection
New Connection
New Connection
^C
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
^C
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/13. HTTP Module$ node app.js
Listening on port 3000 ....
```