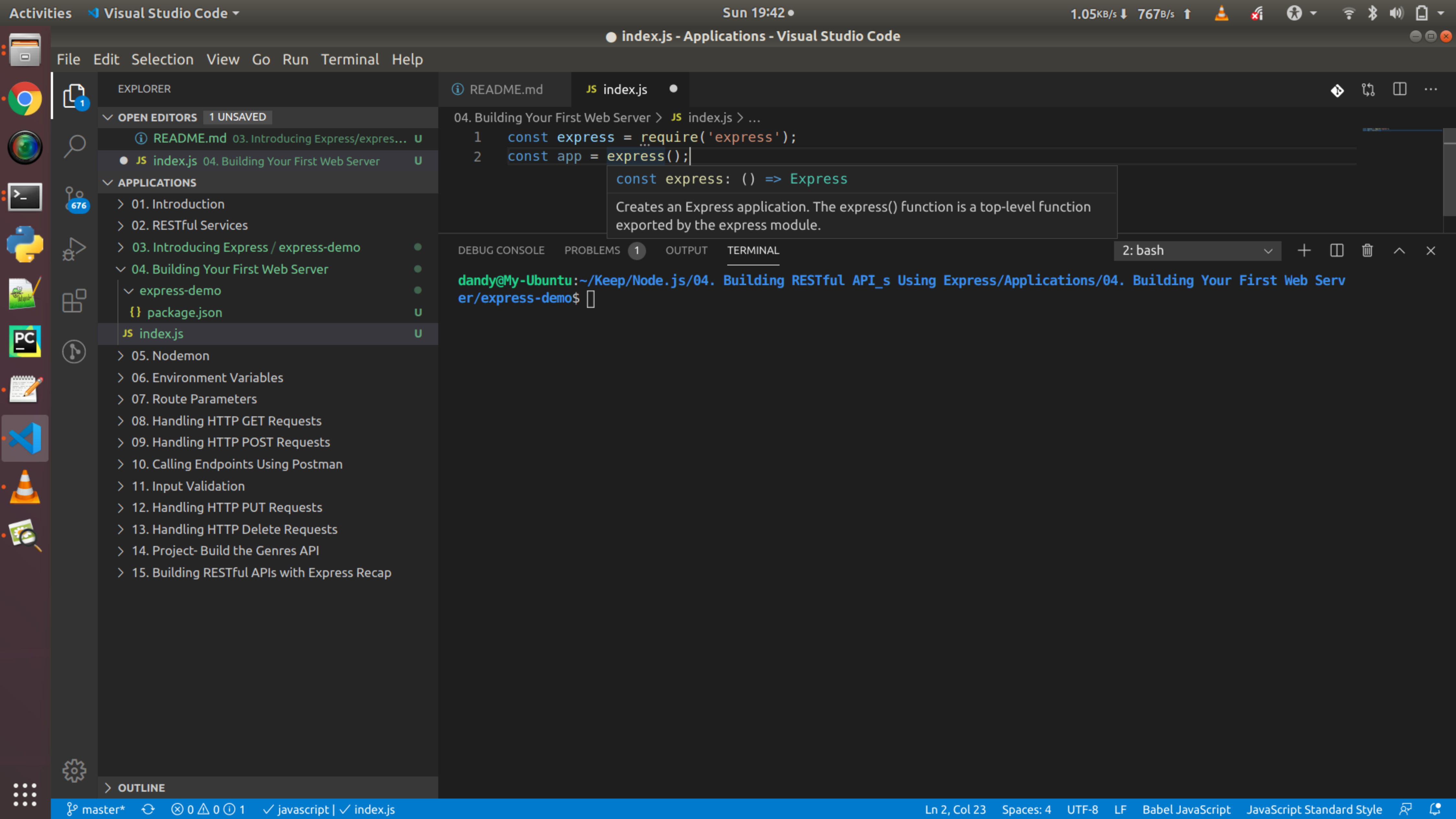
Your First Web Server

Mosh Hamedani
programmingwithmosh.com



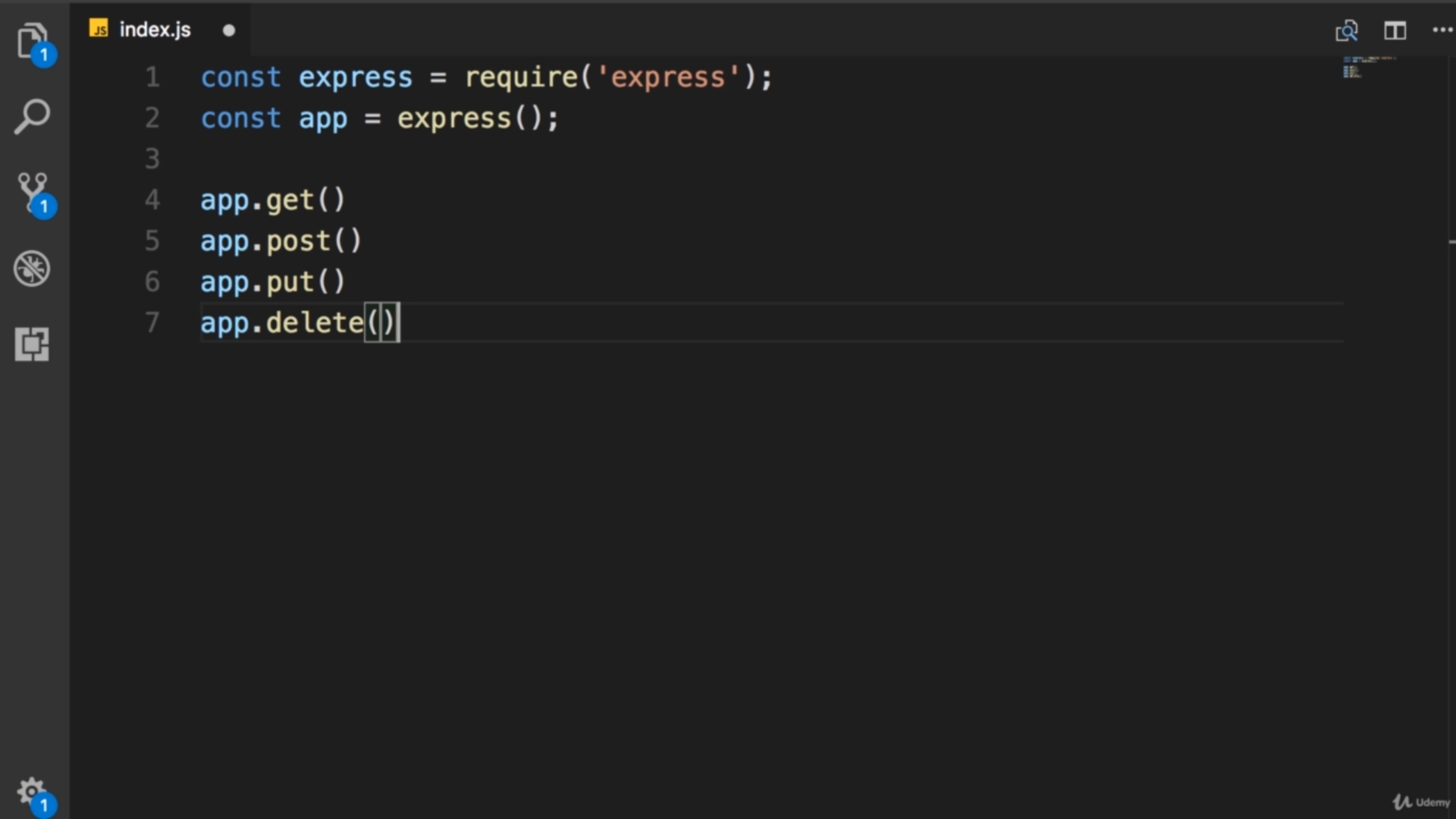
```
const express = require('express');
const app = express();
```

```
const express: () => Express
```

Creates an Express application. The express() function is a top-level function exported by the express module.

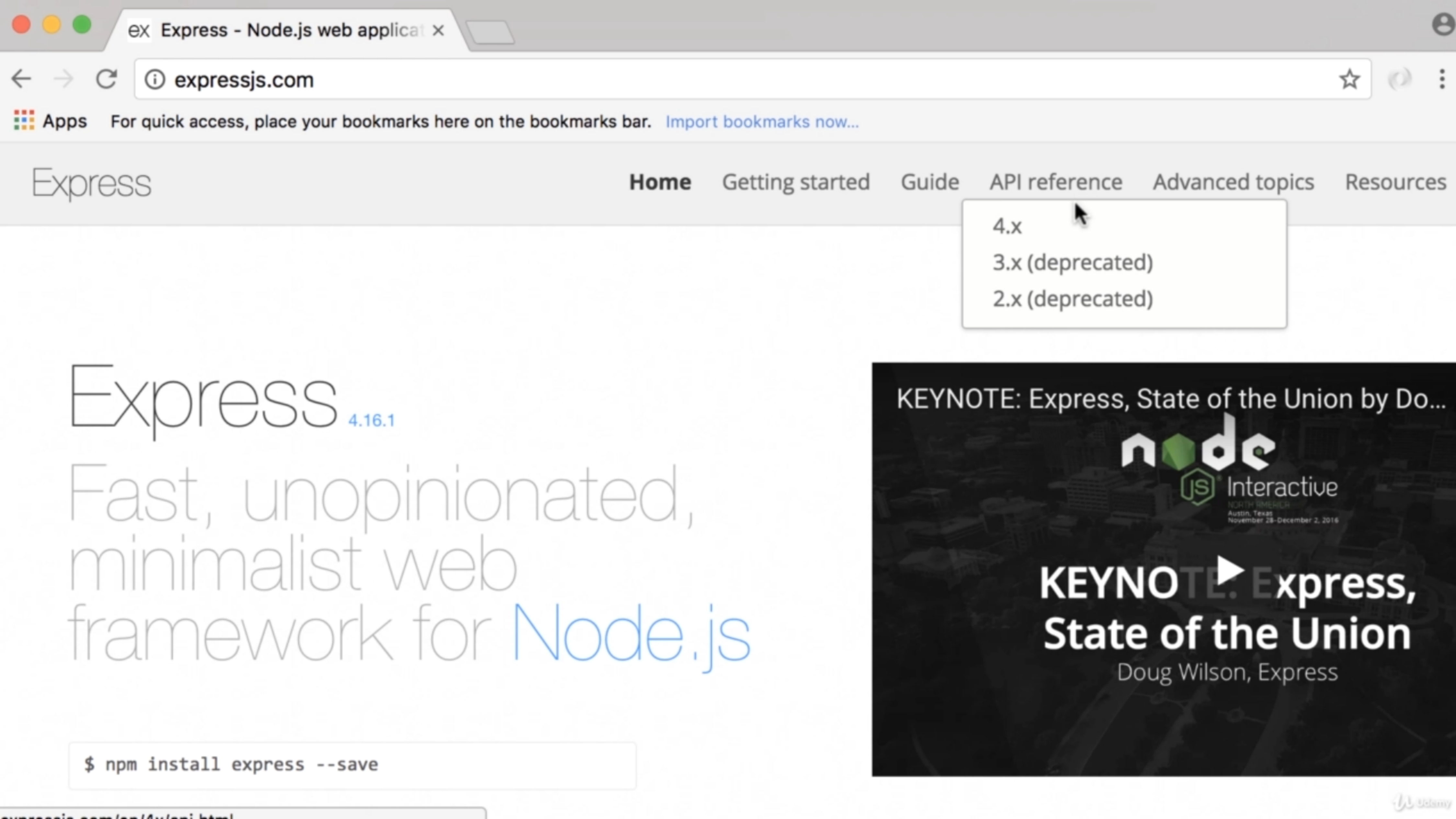
```
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/04. Building Your First Web Serv
er/express-demo$
```

Activities
Visual Studio Code
Sun 19:42
index.js - Applications - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS 1 UNSAVED
README.md 03. Introducing Express/expres...
index.js 04. Building Your First Web Server
APPLICATIONS
01. Introduction
02. RESTful Services
03. Introducing Express / express-demo
04. Building Your First Web Server
express-demo
package.json
index.js
05. Nodemon
06. Environment Variables
07. Route Parameters
08. Handling HTTP GET Requests
09. Handling HTTP POST Requests
10. Calling Endpoints Using Postman
11. Input Validation
12. Handling HTTP PUT Requests
13. Handling HTTP Delete Requests
14. Project- Build the Genres API
15. Building RESTful APIs with Express Recap
OUTLINE
README.md
index.js
04. Building Your First Web Server > index.js > ...
DEBUG CONSOLE
PROBLEMS 1
OUTPUT
TERMINAL
2: bash
master*
javascript | index.js
Ln 2, Col 23
Spaces: 4
UTF-8
LF
Babel JavaScript
JavaScript Standard Style



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

app.get()
app.post()
app.put()
app.delete()
```


Express - Node.js web applica

expressjs.com

Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Express

Home Getting started Guide API reference Advanced topics Resources

4.x
3.x (deprecated)
2.x (deprecated)

Express 4.16.1

Fast, unopinionated, minimalist web framework for Node.js

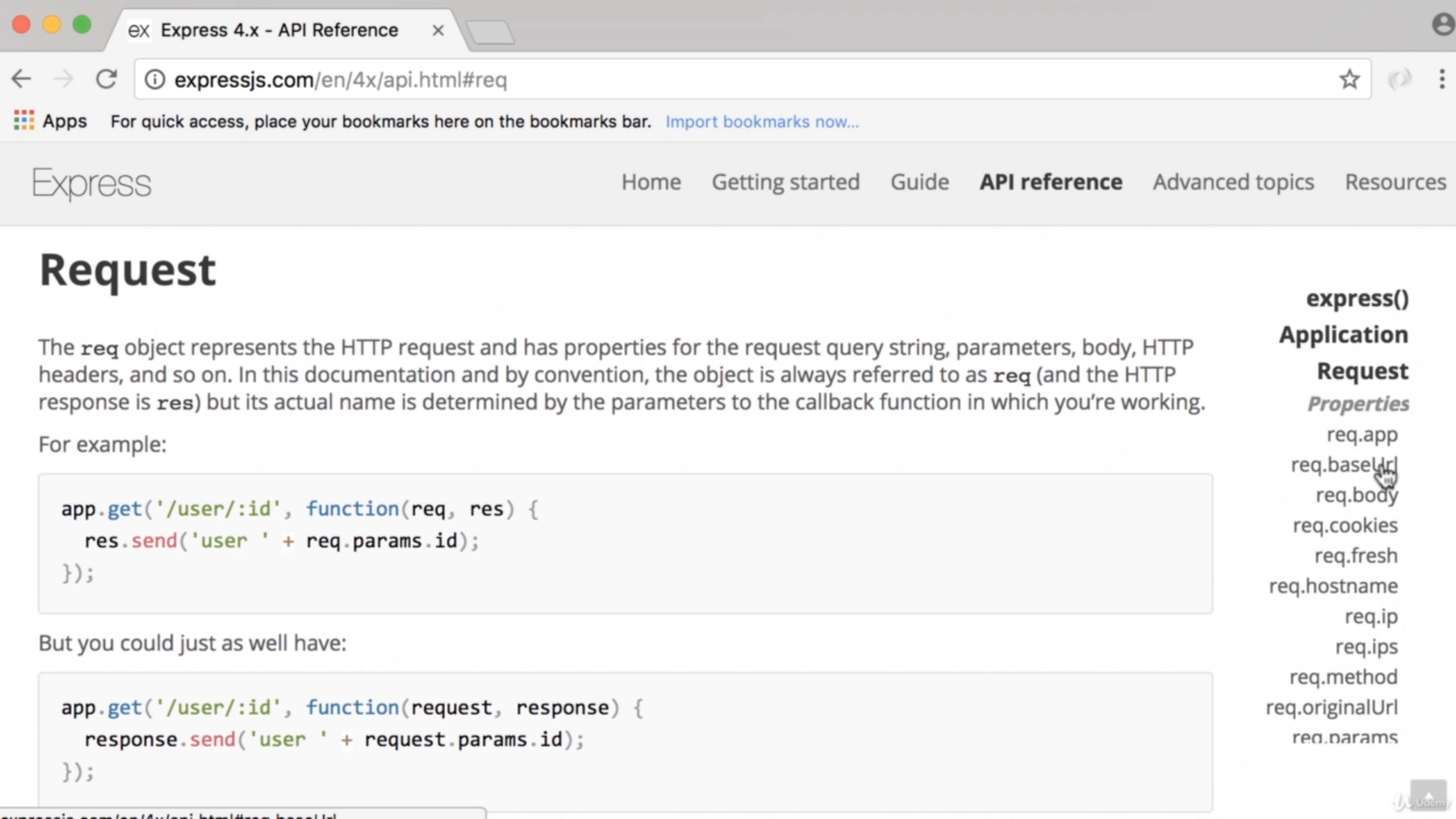
```
$ npm install express --save
```

KEYNOTE: Express, State of the Union by Do...

node js Interactive
NORTH AMERICA
Austin, Texas
November 28-December 2, 2016

KEYNOTE: Express, State of the Union
Doug Wilson, Express

expressjs.com/en/4x/api.html



Request

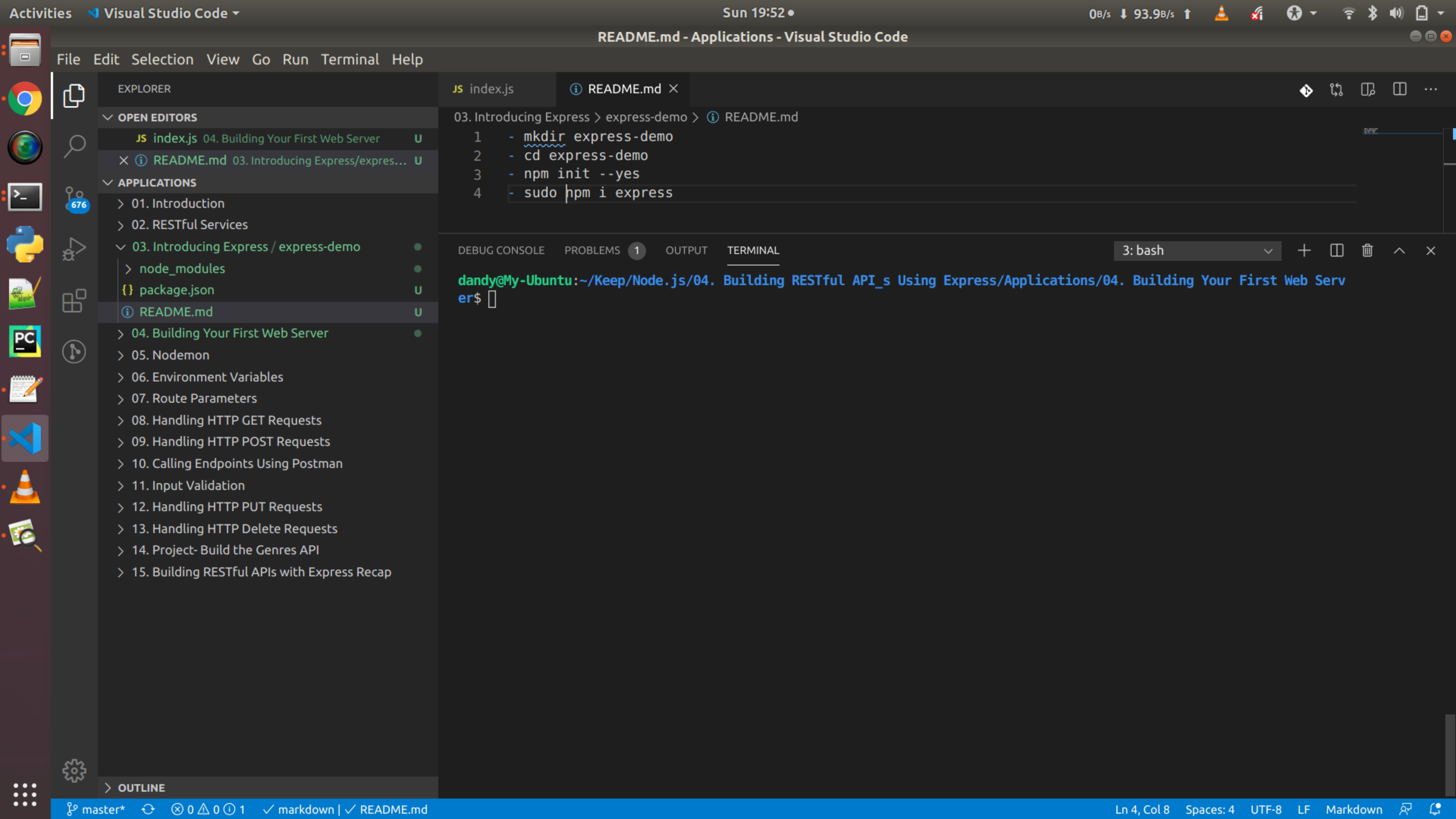
The `req` object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on. In this documentation and by convention, the object is always referred to as `req` (and the HTTP response is `res`) but its actual name is determined by the parameters to the callback function in which you're working.

For example:

```
app.get('/user/:id', function(req, res) {
  res.send('user ' + req.params.id);
});
```

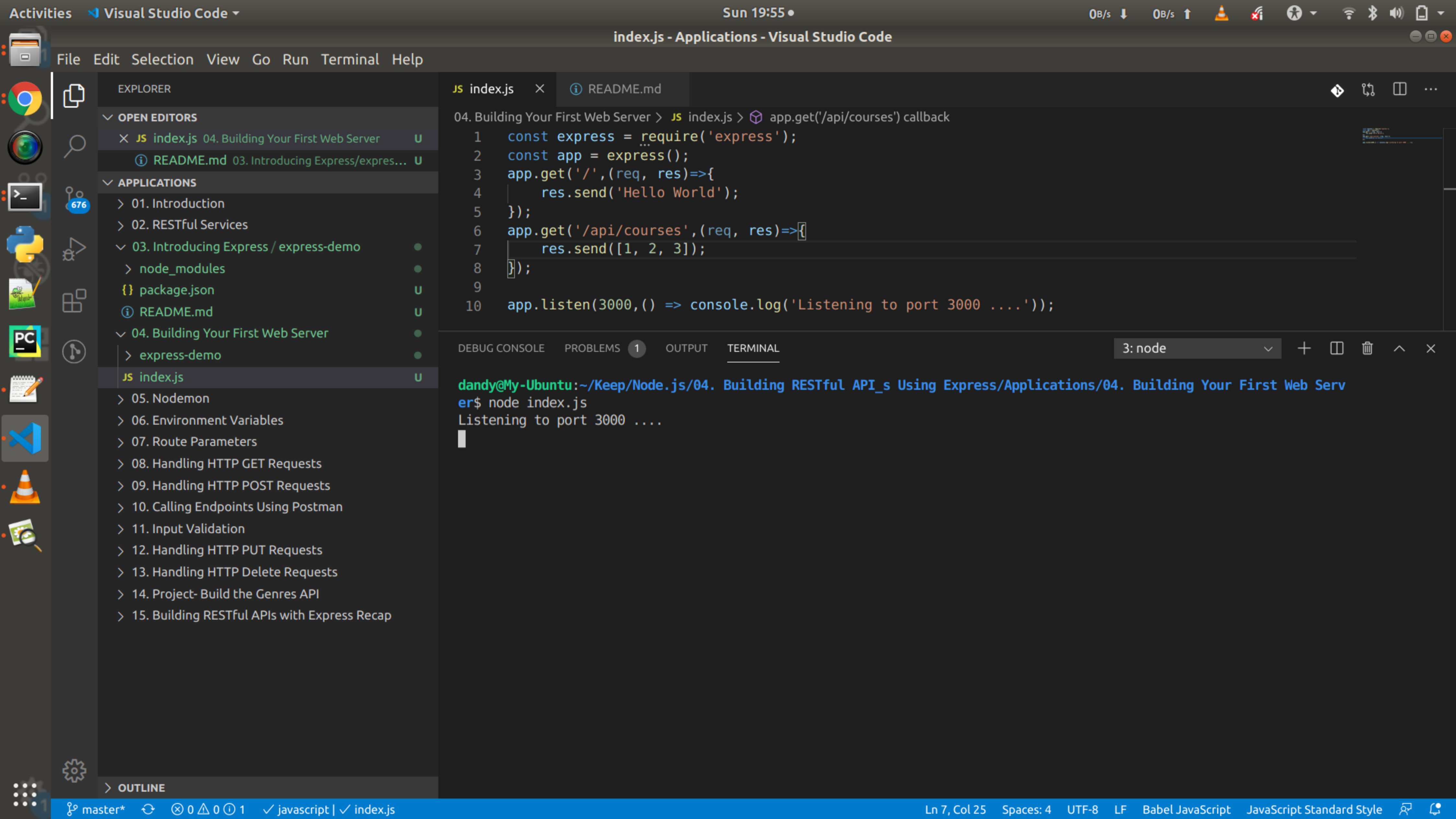
But you could just as well have:

```
app.get('/user/:id', function(request, response) {
  response.send('user ' + request.params.id);
});
```


Activities
Visual Studio Code
Sun 19:52
README.md - Applications - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
index.js 04. Building Your First Web Server U
README.md 03. Introducing Express/expres... U
APPLICATIONS
01. Introduction
02. RESTful Services
03. Introducing Express / express-demo
node_modules
package.json U
README.md U
04. Building Your First Web Server
05. Nodemon
06. Environment Variables
07. Route Parameters
08. Handling HTTP GET Requests
09. Handling HTTP POST Requests
10. Calling Endpoints Using Postman
11. Input Validation
12. Handling HTTP PUT Requests
13. Handling HTTP Delete Requests
14. Project- Build the Genres API
15. Building RESTful APIs with Express Recap
OUTLINE
index.js
README.md
03. Introducing Express > express-demo > README.md
DEBUG CONSOLE
PROBLEMS 1
OUTPUT
TERMINAL
3: bash
master*
0 0 1
markdown | README.md
Ln 4, Col 8
Spaces: 4
UTF-8
LF
Markdown

```
- mkdir express-demo
- cd express-demo
- npm init --yes
- sudo npm i express
```

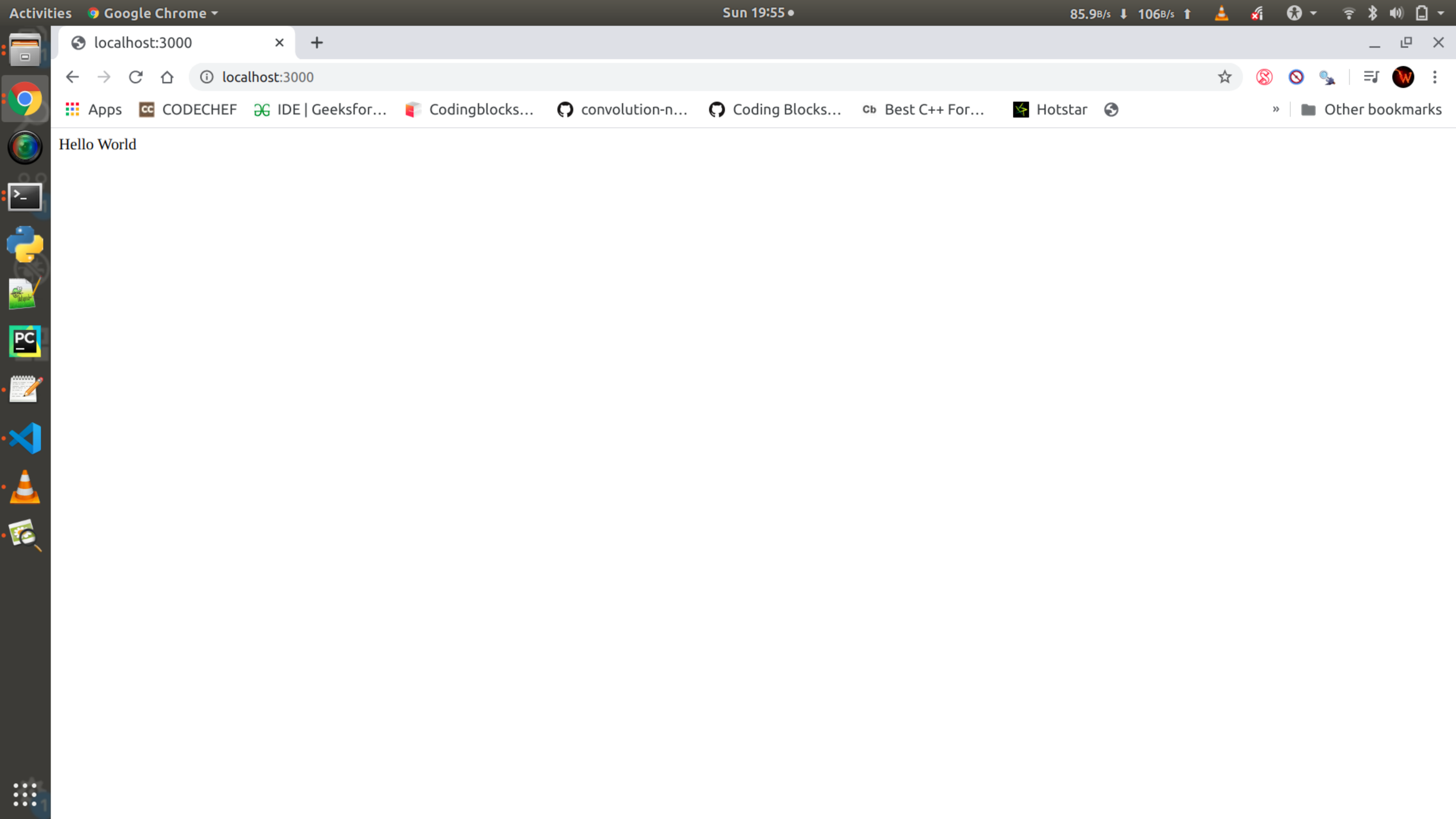
```
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/04. Building Your First Web Serv
er$ 
```


Activities
Visual Studio Code
Sun 19:55
index.js - Applications - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
index.js 04. Building Your First Web Server U
README.md 03. Introducing Express/expres... U
APPLICATIONS
01. Introduction
02. RESTful Services
03. Introducing Express / express-demo
node_modules
package.json U
README.md U
04. Building Your First Web Server
express-demo
index.js U
05. Nodemon
06. Environment Variables
07. Route Parameters
08. Handling HTTP GET Requests
09. Handling HTTP POST Requests
10. Calling Endpoints Using Postman
11. Input Validation
12. Handling HTTP PUT Requests
13. Handling HTTP Delete Requests
14. Project- Build the Genres API
15. Building RESTful APIs with Express Recap
OUTLINE
index.js
README.md
04. Building Your First Web Server > index.js > app.get('/api/courses') callback
DEBUG CONSOLE
PROBLEMS 1
OUTPUT
TERMINAL
3: node
master*
0 0 1
javascript | index.js
Ln 7, Col 25
Spaces: 4
UTF-8
LF
Babel JavaScript
JavaScript Standard Style

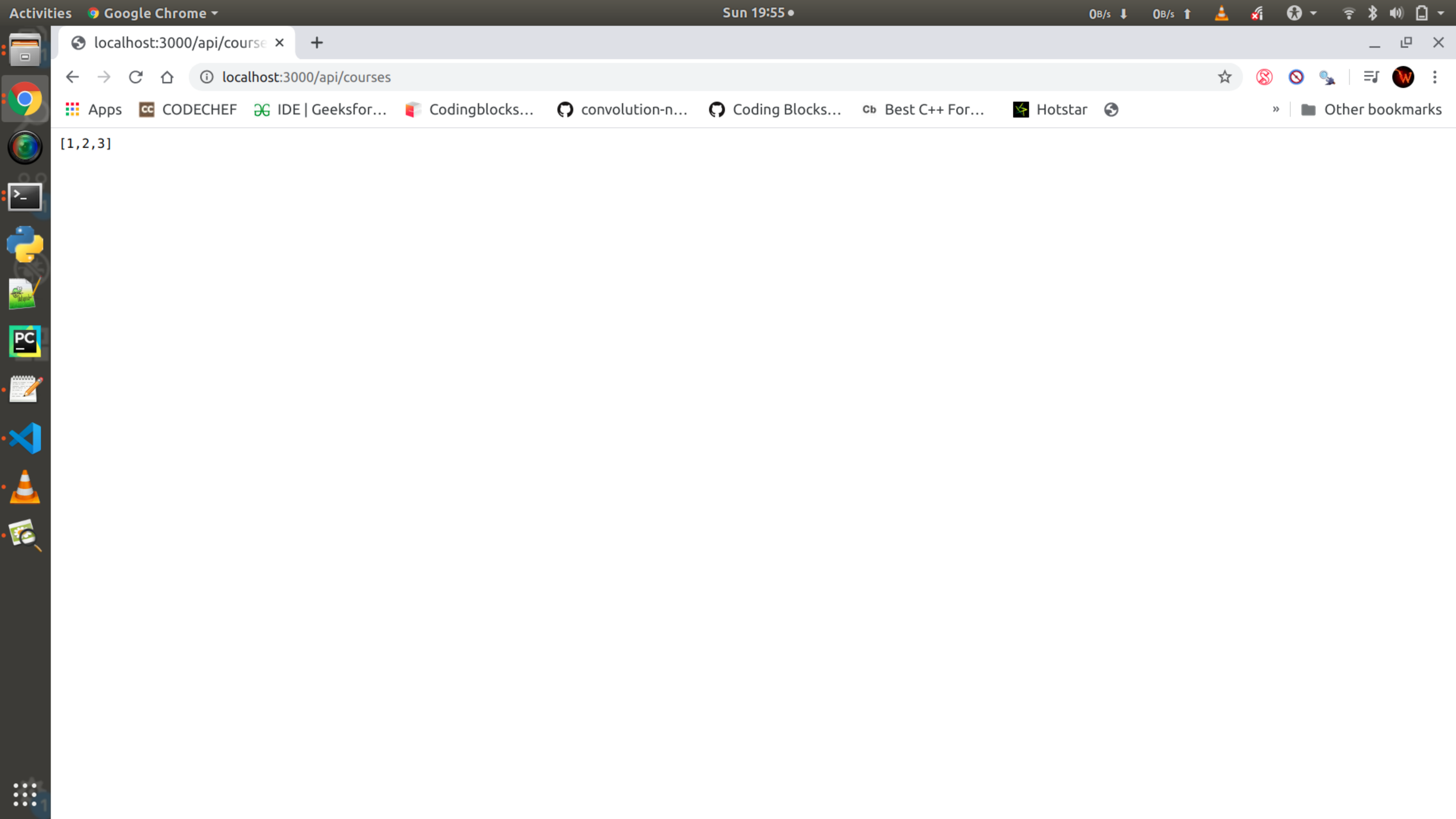
```javascript
const express = require('express');
const app = express();
app.get('/',(req, res)=>{
    res.send('Hello World');
});
app.get('/api/courses',(req, res)=>{
    res.send([1, 2, 3]);
});

app.listen(3000,() => console.log('Listening to port 3000 ....'));
```

```
dandy@My-Ubuntu:~/Keep/Node.js/04. Building RESTful API_s Using Express/Applications/04. Building Your First Web Serv
er$ node index.js
Listening to port 3000 ....
```


Activities
Google Chrome
Sun 19:55
85.9B/s
106B/s
localhost:3000
localhost:3000
Apps
CODECHEF
IDE | Geeksfor...
Codingblocks...
convolution-n...
Coding Blocks...
Best C++ For...
Hotstar
Other bookmarks

Hello World



Activities
Google Chrome
Sun 19:55
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]
```