

00:00 / 03:57

Built-in Middleware



Mosh Hamedani
programmingwithmosh.com

ActivitiesVisual Studio Code

Mon 12:180B/s ↓0B/s ↑

README.md - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

OUTLINE

JS index.js04. Built-In MiddlewareU

× README.md04. Built-In MiddlewareU

JS logger.js04. Built-In MiddlewareU

> 01. Introduction

> 02. Middleware

> 03. Creating Custom Middleware

> 04. Built-In Middleware

> 05. Third-party Middleware

> 06. Environments

> 07. Configuration

> 08. Debugging

> 09. Templating Engines

> 10. Database Integration

> 11. Authentication

> 12. Structuring Express Applications

> 13. Project- Restructure the App

> 14. Express- Advanced Topics Recap

JS index.js

README.md ×

JS logger.js

04. Built-In Middleware > README.md

1

- Here is the explanation that should clear doubts on express.json() and express.urlencoded() and the use of body-parser. It took me some time to figure this out.

2

- What is Middleware? It is those methods/functions/operations that are called BETWEEN processing the Request and sending the Response in your application method.

3

- When talking about express.json() and express.urlencoded() think specifically about POST requests (i.e. the .post request object) and PUT Requests (i.e. the .put request object)

4

- You DO NOT NEED express.json() and express.urlencoded() for GET Requests or DELETE Requests.

5

- You NEED express.json() and express.urlencoded() for POST and PUT requests, because in both these requests you are sending data (in the form of some data object) to the server and you are asking the server to accept or store that data (object), which is enclosed in the body (i.e. req.body) of that (POST or PUT) Request

6

- Express provides you with middleware to deal with the (incoming) data (object) in the body of the request.

7

- a. express.json() is a method inbuilt in express to recognize the incoming Request Object as a JSON Object. This method is called as a middleware in your application using the code: app.use(express.json());

8

- b. express.urlencoded() is a method inbuilt in express to recognize the incoming Request Object as strings or arrays. This method is called as a middleware in your application using the code: app.use(express.urlencoded());

9

- ALTERNATIVELY, I recommend using body-parser (it is an NPM package) to do the same thing. It is developed by the same peeps who built express and is designed to work with express. body-parser used to be part of express. Think of body-parser specifically for POST Requests (i.e. the .post request object) and/or PUT Requests (i.e. the .put request object).

10

- In body-parser you can do :

11

- // calling body-parser to handle the Request Object from POST requests

12

- var bodyParser = require('body-parser');

13

- // parse application/json, basically parse incoming Request Object as a JSON Object

14

- app.use(bodyParser.json());

15

- // parse application/x-www-form-urlencoded, basically can only parse incoming Request Object if - - strings or arrays

16

- app.use(bodyParser.urlencoded({ extended: false }));

17

- // combines the 2 above, then you can parse incoming Request Object if object, with nested objects, or generally any type.

18

- app.use(bodyParser.urlencoded({ extended: true }));

master*

0 0

✓ markdown | ✓ README.md

Ln 1, Col 163

Spaces: 4

UTF-8

LF

Markdown

ActivitiesVisual Studio Code

Mon 12:18

0B/s0B/s

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

OUTLINE

JS index.js 04. Built-In Middleware

README.md 04. Built-In Middleware

JS logger.js 04. Built-In Middleware

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

JS index.js

JS logger.js

README.md

05. Third-party Middleware

06. Environments

07. Configuration

08. Debugging

09. Templating Engines

10. Database Integration

11. Authentication

12. Structuring Express Applications

13. Project- Restructure the App

14. Express- Advanced Topics Recap

JS index.js

04. Built-In Middleware > JS index.js > ...

```
1  const Joi = require('joi');
2  const express = require('express');
3  const app = express();
4  const logger = require('./logger')
5
6  app.use(express.json()); // req.body
7  app.use(express.urlencoded()); // key=value&key=value
8
9  // app.use(logger.log);
10 // app.use(logger.authenticate);
11
12 // app.use(function(req, res, next){
13 //   console.log('Logging...');
14 //   next(); // This is used to pass control to the next middleware function.
15 // });
16
17 // app.use(function(req, res, next){
18 //   console.log('Authenticating....');
19 //   next();
20 // });
21 // // Middleware functions are called in sequence
22
23 const courses = [
24   { id: 1, name: 'course1' },
25   { id: 2, name: 'course2' },
26   { id: 3, name: 'course3' }
27 ];
28
29 app.get('/', (req, res) => {
30   res.send('Hello World !!! Its corona-time');
31 });
32 app.get('/api/courses', (req, res) => {
33   res.send(courses);
34 });
35
36 Complexity is 4 Everything is cool!
37 app.get('/api/courses/:id', (req, res) => {
```

master*

0 0 0

✓ javascript | ✓ index.js

Ln 7, Col 54

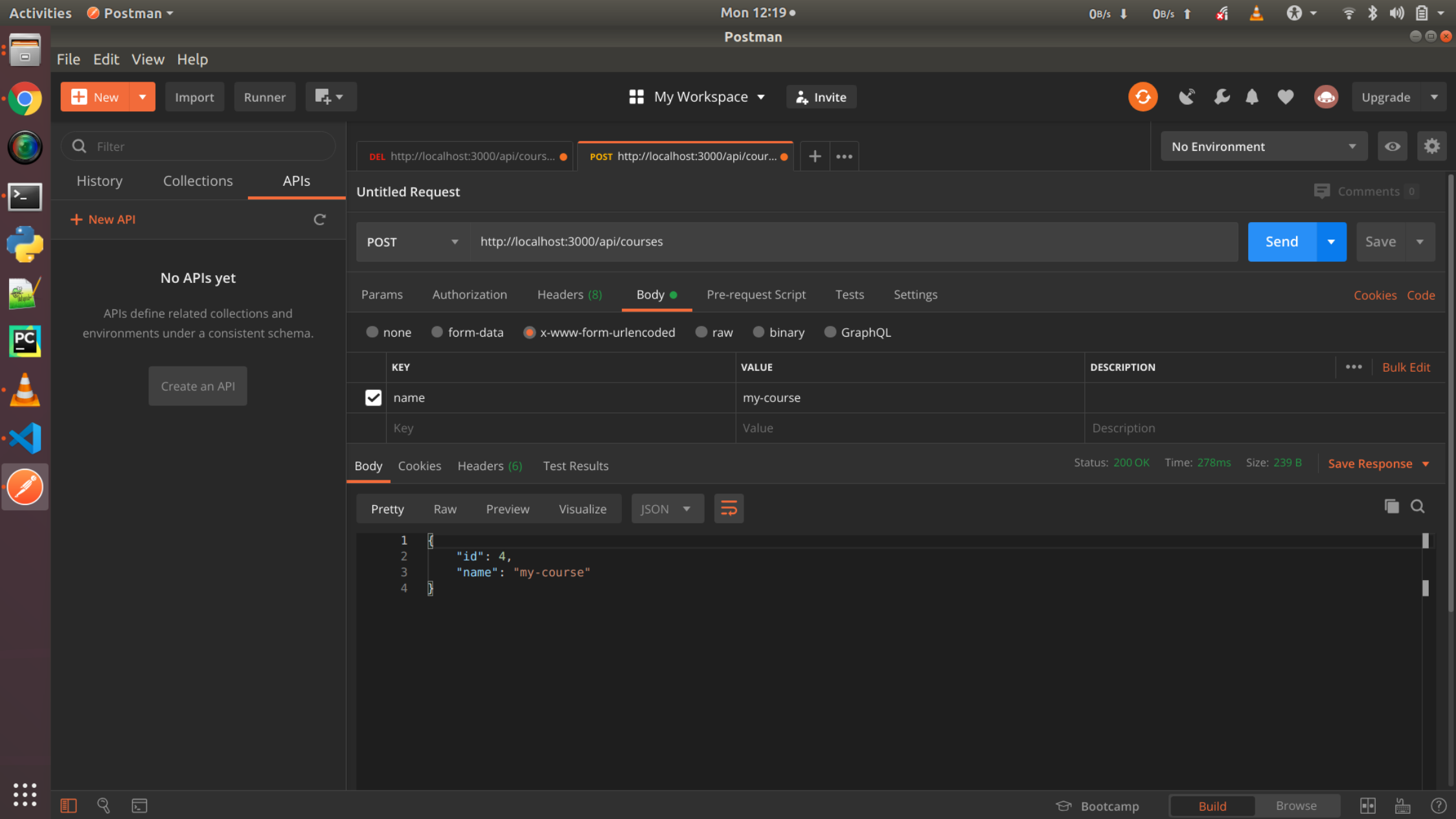
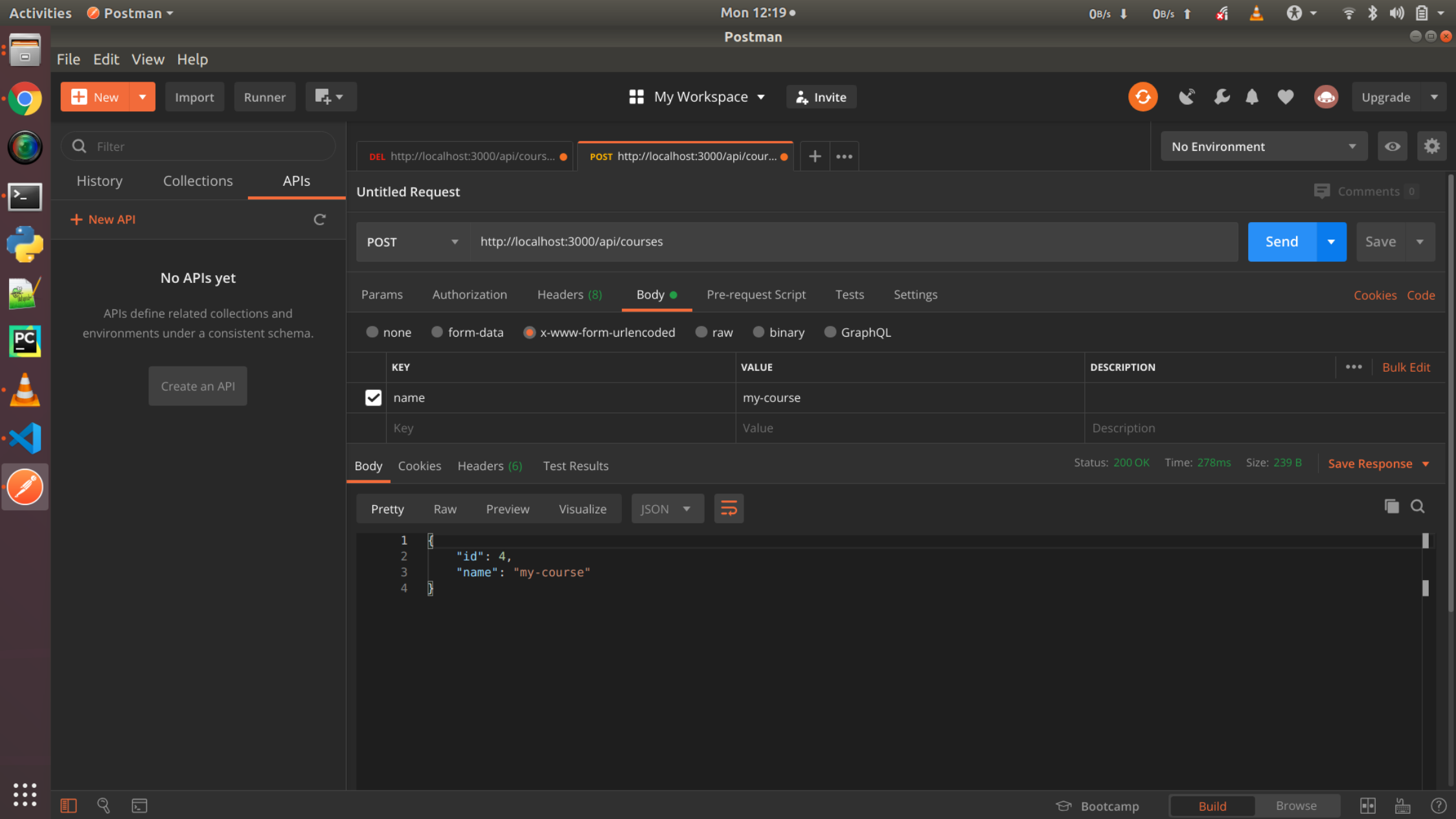
Spaces: 4

UTF-8

LF

Babel JavaScript

JavaScript Standard Style



ActivitiesVisual Studio Code

Mon 12:20 •86.0B/s ↓ 114B/s ↑🔴🟡🟢📶🔊🗑

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

📁

🔍

🌐

📄

🔧

📌

🔑

⚙️

EXPLORER

OPEN EDITORS

APPLICATIONS

OUTLINE

JS index.js ×README.mdJS logger.js

04. Built-In Middleware > JS index.js > ...

```
1  const Joi = require('joi');
2  const express = require('express');
3  const app = express();
4  const logger = require('./logger')
5
6  app.use(express.json()); // req.body
7  app.use(express.urlencoded()); // key=value&key=value
8
9  // app.use(logger.log);
10 // app.use(logger.authenticate);
11
12 // app.use(function(req, res, next){
13 //   console.log('Logging...');
14 //   next(); // This is used to pass control to the next middleware function.
15 // });
16
17 // app.use(function(req, res, next){
18 //   console.log('Authenticating...');
19 //   next();
20 // });
21 // // Middleware functions are called in sequence
22
23 const courses = [
24   { id: 1, name: 'course1' },
25   { id: 2, name: 'course2' }
26 ]
```

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

at Module._compile (module.js:652:30)
at Object.Module._extensions..js (module.js:663:10)
at Module.load (module.js:565:32)
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node indes.js index.js`
body-parser deprecated undefined extended: provide extended option index.js:7:17
Listening to port 3000
[]

master*0 0 0✓ javascript | ✓ index.jsLn 21, Col 51Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style🔍🗑📄🔧🔑⚙️

ActivitiesVisual Studio Code

Mon 12:2185.9B/s114B/s

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS1 UNSAVED

JS index.js04. Built-In MiddlewareU

README.md04. Built-In MiddlewareU

JS logger.js04. Built-In MiddlewareU

APPLICATIONS

> 01. Introduction

> 02. Middleware

> 03. Creating Custom Middleware

> 04. Built-In Middleware

JS index.jsU

JS logger.jsU

README.mdU

> 05. Third-party Middleware

> 06. Environments

> 07. Configuration

> 08. Debugging

> 09. Templating Engines

> 10. Database Integration

> 11. Authentication

> 12. Structuring Express Applications

> 13. Project- Restructure the App

> 14. Express- Advanced Topics Recap

OUTLINE

JS index.js

README.md

JS logger.js

04. Built-In Middleware > JS index.js > extended

1const Joi = require('joi');

2const express = require('express');

3const app = express();

4const logger = require('./logger')

5

6app.use(express.json()); // req.body

7app.use(express.urlencoded({extended : true})); // key=value&key=value

8

9// app.use(logger.log);

10// app.use(logger.authenticate);

11

12// app.use(function(req, res, next){

13// console.log('Logging...');

14// next(); // This is used to pass control to the next middleware function.

15// });

16

17// app.use(function(req, res, next){

18// console.log('Authenticating...');

19// next();

20// });

21// // Middleware functions are called in sequence

22

23const courses = [

24{ id: 1, name: 'course1' },

25{ id: 2, name: 'course2' }.

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

at Module._compile (module.js:652:30)
at Object.Module._extensions..js (module.js:663:10)
at Module.load (module.js:565:32)
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node indes.js index.js`
body-parser deprecated undefined extended: provide extended option index.js:7:17
Listening to port 3000
[

Ln 7, Col 44Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style

ActivitiesVisual Studio Code

Mon 12:24 •164B/s ↓ 93.9B/s ↑

readme.txt - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

OUTLINE

JS index.js04. Built-In MiddlewareU

readme.txt04. Built-In Middleware/publicU

README.md04. Built-In MiddlewareU

JS logger.js04. Built-In MiddlewareU

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

public

readme.txtU

JS index.jsU

JS logger.jsU

README.mdU

05. Third-party Middleware

06. Environments

07. Configuration

08. Debugging

09. Templating Engines

10. Database Integration

11. Authentication

12. Structuring Express Applications

13. Project- Restructure the App

14. Express- Advanced Topics Recap

JS index.jsreadme.txt xREADME.mdJS logger.js

04. Built-In Middleware > public > readme.txt

1 This is a readme file!

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

at Function.Module._load (module.js:497:3)
at Function.Module.runMain (module.js:693:10)
at startup (bootstrap_node.js:188:16)
at bootstrap_node.js:609:3
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node indes.js index.js`
Listening to port 3000
[

master*0 0 0plaintext | ✓ readme.txtLn 1, Col 23Spaces: 4UTF-8LFPlain Text

EXPLORER

JS logger.js

04. Built-In Middleware > JS index.js > extended

U

U

U

```
6 app.use(express.json()); // req.body
```

U

U

U

```
25      { id: 1, name: 'course1' },
```

TERMINAL

> OUTLINE

