

Debugging



Mosh Hamedani
programmingwithmosh.com

ActivitiesVisual Studio Code

Mon 14:07 •0B/s ↓0B/s ↑

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

07. Configuration

OUTLINE

JS index.js 08. Debugging

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

07. Configuration

08. Debugging

JS index.js

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

08. Debugging > JS index.js > ...

1const express = require('express');

2const app = express();

3const morgan = require('morgan');

4

5if (app.get('env') == 'development') {

6app.use(morgan('tiny'));

7console.log('Morgan enabled...');

8}

9

10const port = process.env.PORT || 3000;

11app.listen(port, () => console.log(`Listening to port \${port}`));

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: bash

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$

master*0 0 0javascript | index.js

Ln 11, Col 71Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style

Activities

Visual Studio Code

Mon 14:08

82.0B/s

113B/s

index.js - Applications - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

APPLICATIONS

08. Debugging

JS index.js

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

08. Debugging

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

07. Configuration

08. Debugging

JS index.js

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

08. Debugging

1 const express = require('express');

2 const app = express();

3 const morgan = require('morgan');

4

5 if (app.get('env') == 'development') {

6 app.use(morgan('tiny'));

7 console.log('Morgan enabled...');

8 }

9

10 const port = process.env.PORT || 3000;

11 app.listen(port, () => console.log(`Listening to port \${port}`));

DEBUG CONSOLE

PROBLEMS

OUTPUT

TERMINAL

1: bash

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ sudo npm i debug

[sudo] password for dandy:

npm WARN deprecated joi@14.3.1: This module has moved and is now available at @hapi/joi. Please update your dependencies as this version is no longer maintained and may contain bugs and security issues.

ms@2.0.0 node_modules/debug/node_modules/ms -> node_modules/mquery/node_modules/ms

/home/dandy

├─ @types/node@13.11.0

├─ config@3.3.1

├─ debug@4.1.1

├─ express@4.17.1

├─ helmet@3.22.0

├─ joi@14.3.1

├─ mongoose@5.9.7

├─ mquery@3.2.2

├─ debug@3.1.0

├─ morgan@1.10.0

├─ pg@8.0.0

└─ underscore@1.10.2

npm WARN enoent ENOENT: no such file or directory, open '/home/dandy/package.json'

npm WARN dandy No description

npm WARN dandy No repository field.

npm WARN dandy No README data

npm WARN dandy No license field.

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$

master*

0 0

✓ javascript | ✓ index.js

Ln 11, Col 71

Spaces: 4

UTF-8

LF

Babel JavaScript

JavaScript Standard Style

Activities

Visual Studio Code

Mon 14:26

98.0B/s 94.0B/s

index.js - Applications - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

APPLICATIONS

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

index.js

07. Configuration

config

custom-environment-variables.json

default.json

development.json

production.json

index.js

08. Debugging

index.js

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

08. Debugging > JS index.js > ...

```
1 const startupDebugger = require('debug')('app:startup');
2 const dbDebugger = require('debug')('app:db');
3 const express = require('express');
4 const app = express();
5 const morgan = require('morgan');
6
7 if (app.get('env') === 'development') {
8   app.use(morgan('tiny'));
9   // console.log('Morgan enabled...');
10  startupDebugger('Morgan enabled...');
11 }
12
13 // Db work...
14 dbDebugger('Connected to the database...');
15
16 const port = process.env.PORT || 3000;
17 app.listen(port, () => console.log(`Listening to port ${port} ....`));
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

1: node

```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging$ export NODE_ENV=development
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging$ export DEBUG=app:startup
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging$ nodemon index.js
[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
app:startup Morgan enabled... +0ms
Listening to port 3000 ....
```

master*

0 0 0

✓ javascript | ✓ index.js

Ln 9, Col 41 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style

ActivitiesVisual Studio Code

65.9B/s336B/s

Mon 14:27

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

index.js

07. Configuration

config

custom-environment-variables.json

default.json

development.json

production.json

index.js

08. Debugging

index.js

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

08. Debugging > JS index.js > ...

1const startupDebugger = require('debug')('app:startup');

2const dbDebugger = require('debug')('app:db');

3const express = require('express');

4const app = express();

5const morgan = require('morgan');

6

7if (app.get('env') == 'development') {

8app.use(morgan('tiny'));

9// console.log('Morgan enabled...');

10startupDebugger('Morgan enabled...');

11}

12

13// Db work...

14dbDebugger('Connected to the database...');

15

16const port = process.env.PORT || 3000;

17app.listen(port, () => console.log(`Listening to port \${port}`));

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ export NODE_ENV=development

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ export DEBUG=app:startup

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ nodemon index.js

[nodemon] 2.0.2

[nodemon] to restart at any time, enter `rs`

[nodemon] watching dir(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index.js`

app:startup Morgan enabled... +0ms

Listening to port 3000

^C

nodemon update check failed

Try running with sudo or get access

to the local update config store via

sudo chown -R \$USER:(id -gn \$USER) /home/dandy/.config

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ export DEBUG=

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ nodemon index.js

[nodemon] 2.0.2

[nodemon] to restart at any time, enter `rs`

[nodemon] watching dir(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index.js`

Listening to port 3000

Ln 9, Col 41Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style

EXPLORER

JS index.js X

08. Debugging > JS index.js > ...

```

1  const startupDebugger = require('debug')('app:startup');
2  const dbDebugger = require('debug')('app:db');
3  const express = require('express');
4  const app = express();
5  const morgan = require('morgan');
6
7  if (app.get('env') === 'development') {
8      app.use(morgan('tiny'));
9      // console.log('Morgan enabled...');
10     startupDebugger('Morgan enabled...');
11 }
12
13 // Db work...
14 dbDebugger('Connected to the database...');
15
16 const port = process.env.PORT || 3000;
17 app.listen(port, () => console.log(`Listening to port ${port} ....`));

```

```
6
7 if (app.get('env') == 'development') {
```

```
8 app.use(morgan('tiny'));
9 // console.log('Morgan enabled...');
10 startupDebugger('Morgan enabled...')
```

11

12

```
13 // Db work...
```

```
14 dbDebugger('Connected to the database...');
```

15

```
16  const port = process.env.PORT || 3000;
```

```
17 app.listen(port, () => console.log(`Listening to port ${port} ....`));
```

DEBUG COM

```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging$ export DEBUG=app:*
```

```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging$ nodemon index.js
```

[nodemon] 2.0.2

[nodemon] to restart at any time, enter `rs`

```
[nodemon] watching dir(s): *.*
```

```
[nodemon] watching extensions: js,mjs,json
```

```
[nodemon] starting `node index.js`
```

```
app:startup Morgan enabled... +0ms
```

```
app:db Connected to the database
```

```
Listening to port 3000 ....
```

1

master* 0 0 ✓ javascript | ✓ index.js

Ln 9, Col 41 Spaces: 4 UTF-8 LF Babel JavaScript JavaScript Standard Style

ActivitiesVisual Studio Code

Mon 14:32 •0B/s ↓0B/s ↑

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

JS index.js

07. Configuration

config

custom-environment-variables.json

default.json

development.json

production.json

JS index.js

08. Debugging

JS index.js

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

08. Debugging > JS index.js > ...

1const startupDebugger = require('debug')('app:startup');

2const dbDebugger = require('debug')('app:db');

3const express = require('express');

4const app = express();

5const morgan = require('morgan');

6

7if (app.get('env') == 'development') {

8app.use(morgan('tiny'));

9// console.log('Morgan enabled...');

10startupDebugger('Morgan enabled...');

11}

12

13// Db work...

14dbDebugger('Connected to the database...');

15

16const port = process.env.PORT || 3000;

17app.listen(port, () => console.log(`Listening to port \${port}`));

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ DEBUG=app:db nodemon index.js

[nodemon] 2.0.2

[nodemon] to restart at any time, enter `rs`

[nodemon] watching dir(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index.js`

app:db Connected to the database... +0ms

Listening to port 3000

master*0 0 0javascript | index.jsLn 9, Col 41Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style

ActivitiesVisual Studio Code

Mon 14:360B/s ↓0B/s ↑

index.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

01. Introduction

02. Middleware

03. Creating Custom Middleware

04. Built-In Middleware

05. Third-party Middleware

06. Environments

JS index.js

07. Configuration

config

custom-environment-variables.json

default.json

development.json

production.json

JS index.js

08. Debugging

JS index.js

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

08. Debugging > JS index.js > ...

18

19const debug = require('debug')('app:startup');

20const express = require('express');

21const app = express();

22const morgan = require('morgan');

23

24if (app.get('env') == 'development') {

25app.use(morgan('tiny'));

26// console.log('Morgan enabled...');

27debug('Morgan enabled...');

28}

29

30const port = process.env.PORT || 3000;

31app.listen(port, () => console.log(`Listening to port \${port}`));

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

1: node

dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/08. Debugging\$ DEBUG=app:startup nodemon ind

ex.js

[nodemon] 2.0.2

[nodemon] to restart at any time, enter `rs`

[nodemon] watching dir(s): *.*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index.js`

app:startup Morgan enabled... +0ms

Listening to port 3000

master*0 0 0javascript | index.jsLn 28, Col 2Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style