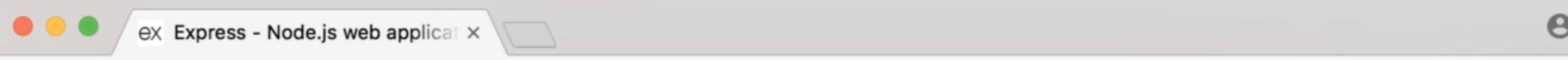


# Third-party Middleware



**Mosh Hamedani**

[programmingwithmosh.com](http://programmingwithmosh.com)



← → ⌂ ⓘ expressjs.com ⭐ ⚙ ⋮

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

# Express

Fast, unopinionated,  
minimalist web  
framework for Node.js

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

**Home** Getting started Guide API reference Advanced topics Resources

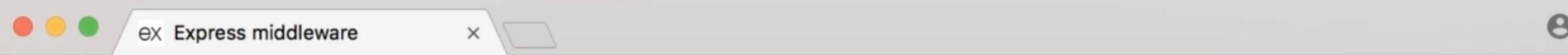
Community Glossary **Middleware** Utility mod Framework Books and Companies Contributing Release Ch

**KEYNOTE: Express, State of the Union**

 **Interaction**  
NORTH AMERICA  
Austin, Texas  
November 28–December 1

**KEYNOTE: Express, State of the Union**

Doug Wilson, Express

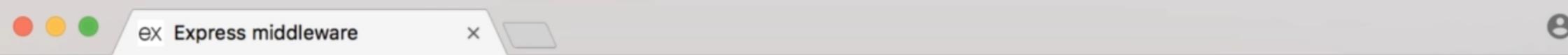


- [body-parser](#)
- [compression](#)
- [connect-rid](#)
- [cookie-parser](#)
- [cookie-session](#)
- [cors](#)
- [csurf](#)
- [errorhandler](#)
- [method-override](#)
- [morgan](#)
- [multer](#)
- [response-time](#)
- [serve-favicon](#)
- [serve-index](#)
- [serve-static](#)
- [session](#)
- [timeout](#)

## Express middleware

The Express middleware modules listed here are maintained by the [Expressjs team](#).

Middleware module	Description	Replaces built-in function (Express 3)
<a href="#">body-parser</a>	Parse HTTP request body. See also: <a href="#">body</a> , <a href="#">co-body</a> , and <a href="#">raw-body</a> .	<code>express.bodyParser</code>
<a href="#">compression</a>	Compress HTTP responses.	<code>express.compress</code>
<a href="#">connect-rid</a>	Generate unique request ID.	NA
<a href="#">cookie-parser</a>	Parse cookie header and populate <code>req.cookies</code> . See also <a href="#">cookies</a> and <a href="#">keygrip</a> .	<code>express.cookieParser</code>
<a href="#">cookie-session</a>	Establish cookie-based sessions.	<code>express.cookieSession</code>

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**

<a href="#">response</a>	query-string; by using Google API's Partial Response.
<a href="#">express-simple-cdn</a>	Use a CDN for static assets, with multiple host support.
<a href="#">express-slash</a>	Handles routes with and without trailing slashes.
<a href="#">express-stormpath</a>	User storage, authentication, authorization, SSO, and data security.
<a href="#">express-uncapitalize</a>	Redirects HTTP requests containing uppercase to a canonical lowercase form.
<a href="#">helmet</a>	Helps secure your apps by setting various HTTP headers.
<a href="#">join-io</a>	Joins files on the fly to reduce the requests count.
<a href="#">passport</a>	Authentication using "strategies" such as OAuth, OpenID and many others. See <a href="http://passportjs.org/">http://passportjs.org/</a> for more information.
<a href="#">static-expiry</a>	Fingerprint URLs or caching headers for static assets.
<a href="#">view-helpers</a>	Common helper methods for views.
<a href="#">sriracha-admin</a>	Dynamically generate an admin site for Mongoose.



Apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Looking for a version of Helmet that supports the Koa framework?

## Quick start

First, run `npm install helmet --save` for your app. Then, in an Express (or Connect) app:

```
var express = require('express')
var helmet = require('helmet')

var app = express()

app.use(helmet())

// ...
```

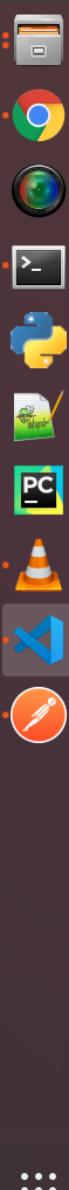
It's best to `use` Helmet early in your middleware stack so that its headers are sure to be set.

You can also use its pieces individually:

index.js - Applications - Visual Studio Code

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

1: bash



```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/04. Built-In Middleware$ sudo npm i helmet
[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.

/home/dandy
└── helmet@3.22.0
    ├── @types/node@13.11.0
    ├── express@4.17.1
    ├── depd@2.0.0
    ├── dns-prefetch-control@0.2.0
    ├── dont-sniff-mimetype@1.1.0
    ├── expect-ct@0.2.0
    ├── feature-policy@0.3.0
    ├── frameguard@3.1.0
    ├── helmet-crossdomain@0.4.0
    ├── helmet-csp@2.10.0
    │   ├── bowser@2.9.0
    │   ├── camelize@1.0.0
    │   └── content-security-policy-builder@2.1.0
    ├── dasherize@2.0.0
    ├── hide-powered-by@1.1.0
    ├── hpkp@2.0.0
    ├── hsts@2.2.0
    │   └── depd@2.0.0
    ├── ienopen@1.1.0
    ├── nocache@2.1.0
    ├── referrer-policy@1.2.0
    └── x-xss-protection@1.3.0
    └── joi@14.3.1
    ├── mongoose@5.9.7
    ├── 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/04. Built-In Middleware\$

index.js - Applications - Visual Studio Code

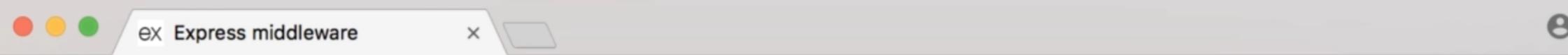
js index.js x

05. Third-party Middleware > **js** index.js > ...

```
1 const Joi = require('joi');
2 const express = require('express');
3 const app = express();
4 const helmet = require('helmet');
5 // Helmet helps you secure your Express.js apps by setting various HTTP headers.
6
7 app.use(express.json()); // req.body
8 app.use(express.urlencoded({extended : true})); // key=value&key=value
9 app.use(express.static('public'));
10 app.use(helmet());|
```

11

```
12 const courses = [
13     { id: 1, name: 'course1' },
14     { id: 2, name: 'course2' },
15     { id: 3, name: 'course3' }
16 ];
17
18 app.get('/', (req, res) => {
19     res.send('Hello World !!! Its corona-time');
20 });
21 app.get('/api/courses', (req, res) => {
22     res.send(courses);
23 });
Complexity is 4 Everything is cool!
24 app.get('/api/courses/:id', (req, res) => { █
25     const course = courses.find(c => c.id === parseInt(req.params.id));
26     if (!course) return res.status(404).send('The course with the given ID was not found.');
27     res.send(course);
28 });
29
Complexity is 3 Everything is cool!
30 app.post('/api/courses', (req, res) => { █
31     const {error} = validateCourse(req.body);█
32     if (error) return res.status(400).send(error.details[0].message);
33     const course = {
34         id: courses.length + 1,
```

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**

<a href="#">csurf</a>	Protect from CSRF exploits.	<code>express.csrf</code>
<a href="#">errorhandler</a>	Development error-handling/debugging.	<code>express.errorHandler</code>
<a href="#">method-override</a>	Override HTTP methods using header.	<code>express.methodOverride</code>
<a href="#">morgan</a>	HTTP request logger.	<code>express.logger</code>
<a href="#">multer</a>	Handle multi-part form data.	<code>express.bodyParser</code>
<a href="#">response-time</a>	Record HTTP response time.	<code>express.responseTime</code>
<a href="#">serve-favicon</a>	Serve a favicon.	<code>express.favicon</code>
<a href="#">serve-index</a>	Serve directory listing for a given path.	<code>express.directory</code>
<a href="#">serve-static</a>	Serve static files.	<code>express.static</code>
<a href="#">session</a>	Establish server-based sessions (development only).	<code>express.session</code>
<a href="#">timeout</a>	Set a timeout period for HTTP request processing.	<code>express.timeout</code>



index.js - Applications - Visual Studio Code

```
js index.js ×

05. Third-party Middleware > js index.js > ...

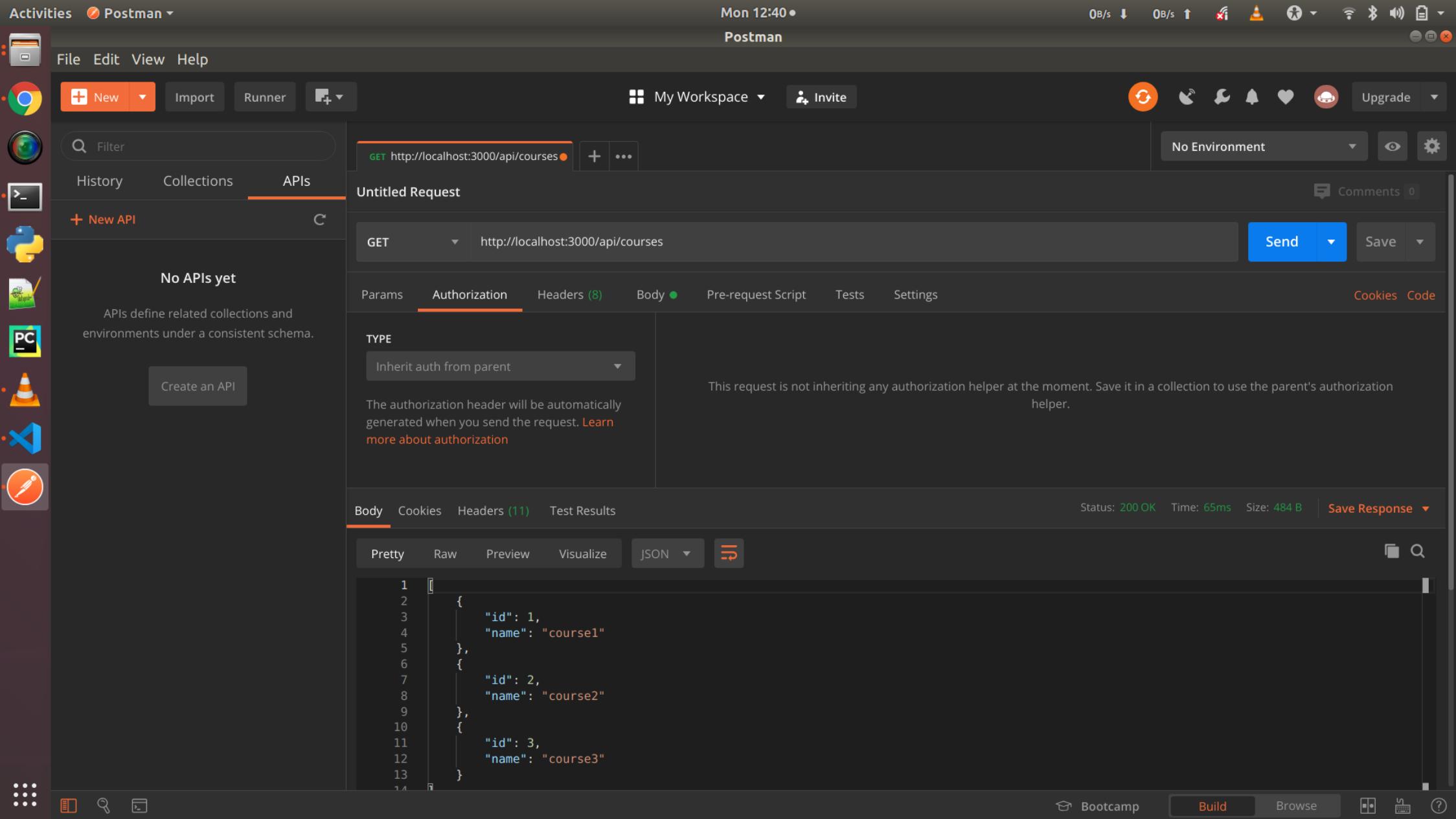
1  const Joi = require('joi');
2  const express = require('express');
3  const app = express();
4  const helmet = require('helmet');
5  // Helmet helps you secure your Express.js apps by setting various HTTP headers.
6  const morgan = require('morgan');
7  // HTTP request logger middleware for node.js.

8
9
10 app.use(express.json()); // req.body
11 app.use(express.urlencoded({extended : true})); // key=value&key=value
12 app.use(express.static('public'));
13 app.use(helmet());
14 app.use(morgan('tiny'));

15
16 const courses = [
17   { id: 1, name: 'course1' },
18   { id: 2, name: 'course2' },
19   { id: 3, name: 'course3' }
20 ];
21
22 app.get('/', (req, res) => {
23   res.send('Hello World !!! Its corona-time');
24 });
25 app.get('/api/courses', (req, res) => {
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications$ cd 05.\ Third-party\ Middleware/
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/05. Third-party Middleware$ 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 ....
```



index.js - Applications - Visual Studio Code

```
05. Third-party Middleware > js index.js > ...
1  const Joi = require('joi');
2  const express = require('express');
3  const app = express();
4  const helmet = require('helmet');
5  // Helmet helps you secure your Express.js apps by setting various HTTP headers.
6  const morgan = require('morgan');
7  // HTTP request logger middleware for node.js.
8
9
10 app.use(express.json()); // req.body
11 app.use(express.urlencoded({extended : true})); // key=value&key=value
12 app.use(express.static('public'));
13 app.use(helmet());
14 app.use(morgan('tiny'));
15
16 const courses = [
17   { id: 1, name: 'course1' },
18   { id: 2, name: 'course2' },
19   { id: 3, name: 'course3' }
20 ];
21
22 app.get('/', (req, res) => {
23   res.send('Hello World !!! Its corona-time');
24 });
25 app.get('/api/courses', (req, res) => {
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

1: node

```
dandy@My-Ubuntu:~/Keep/Node.js/05. Express- Advanced Topics/Applications/05. Third-party Middleware$ 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 ....
GET /api/courses 200 79 - 5.853 ms
```

index.js - Applications - Visual Studio Code

```
6  const morgan = require('morgan');
7  // HTTP request logger middleware for node.js.
8
9
10 app.use(express.json()); // req.body
11 app.use(express.urlencoded({extended : true})); // key=value&key=value
12 app.use(express.static('public'));
13 app.use(helmet());
14 app.use(morgan('tiny'));

15
16 // Using a predefined format string
17 // morgan('tiny')
18 // Using format string of predefined tokens
19 // morgan(':method :url :status :res[content-length] - :response-time ms')
20 // Using a custom format function
21 // morgan(function (tokens, req, res) {
22 //   return [
23 //     tokens.method(req, res),
24 //     tokens.url(req, res),
25 //     tokens.status(req, res),
26 //     tokens.res(req, res, 'content-length'), '-',
27 //     tokens['response-time'](req, res), 'ms'
28 //   ].join(' ')
29 // })
30
```

```
DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL
[nodemon] watching dir(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Listening to port 3000 ....
GET /api/courses 200 79 - 5.853 ms
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Listening to port 3000 ....
[]
```