

ExpressJS - Debugging

 [tutorialspoint.com/expressjs/expressjs_debugging.htm](https://www.tutorialspoint.com/expressjs/expressjs_debugging.htm)



- ExpressJS Tutorial
- [ExpressJS - Home](#)
- [ExpressJS - Overview](#)
- [ExpressJS - Environment](#)
- [ExpressJS - Hello World](#)
- [ExpressJS - Routing](#)
- [ExpressJS - HTTP Methods](#)
- [ExpressJS - URL Building](#)
- [ExpressJS - Middleware](#)
- [ExpressJS - Templating](#)
- [ExpressJS - Static Files](#)
- [ExpressJS - Form Data](#)
- [ExpressJS - Database](#)
- [ExpressJS - Cookies](#)
- [ExpressJS - Sessions](#)
- [ExpressJS - Authentication](#)
- [ExpressJS - RESTful APIs](#)
- [ExpressJS - Scaffolding](#)
- [ExpressJS - Error handling](#)
- [ExpressJS - Debugging](#)
- [ExpressJS - Best Practices](#)
- [ExpressJS - Resources](#)
- ExpressJS Useful Resources
- [ExpressJS - Quick Guide](#)
- [ExpressJS - Useful Resources](#)
- [ExpressJS - Discussion](#)
- Selected Reading
- [UPSC IAS Exams Notes](#)
- [Developer's Best Practices](#)
- [Questions and Answers](#)
- [Effective Resume Writing](#)
- [HR Interview Questions](#)
- [Computer Glossary](#)

- Who is Who

[Previous Page](#)

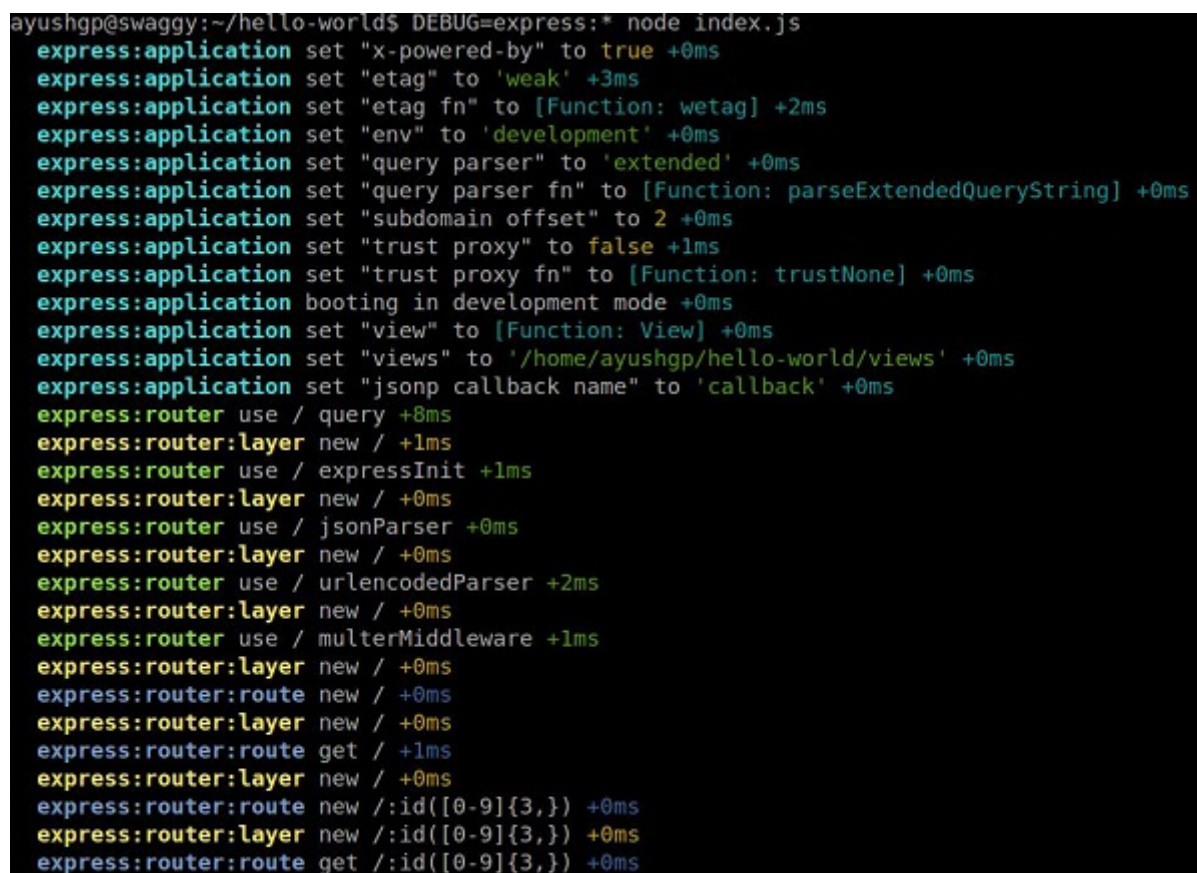
[Next Page](#)

Express uses the [Debug](#) module to internally log information about route matching, middleware functions, application mode, etc.

To see all internal logs used in Express, set the DEBUG environment variable to **Express:*** when starting the app –

```
DEBUG = express:* node index.js
```

The following output will be displayed.

A terminal window with a dark background and light-colored text. The prompt is 'ayushgp@swaggy:~/hello-world\$'. The command entered is 'DEBUG=express:* node index.js'. The output shows a series of log messages from the Express.js application and router, each preceded by a timestamp and a log level (e.g., 'express:application', 'express:router:layer'). The logs detail the initialization of the application, including setting various options like 'x-powered-by', 'etag', 'env', 'query parser', 'subdomain offset', 'trust proxy', 'view', and 'views'. It also shows the setup of the router, including the use of various middleware functions like 'query', 'expressInit', 'jsonParser', 'urlencodedParser', 'multerMiddleware', and the definition of routes for GET requests to '/:id([0-9]{3,})'.

These logs are very helpful when a component of your app is not functioning right. This verbose output might be a little overwhelming. You can also restrict the DEBUG variable to specific area to be logged. For example, if you wish to restrict the logger to application and router, you can use the following code.

```
DEBUG = express:application,express:router node index.js
```

Debug is turned off by default and is automatically turned on in production environment. Debug can also be extended to meet your needs, you can read more about it at [its npm page](#).

