

# ExpressJS - Error Handling

 [tutorialspoint.com/expressjs/expressjs\\_error\\_handling.htm](https://www.tutorialspoint.com/expressjs/expressjs_error_handling.htm)

---

- Error handling in Express is done using middleware. But this middleware has special properties. The error handling middleware are defined in the same way as other middleware functions, except that error-handling functions **MUST have four arguments** instead of three – **err, req, res, next**. For example, to send a response on any error, we can use –

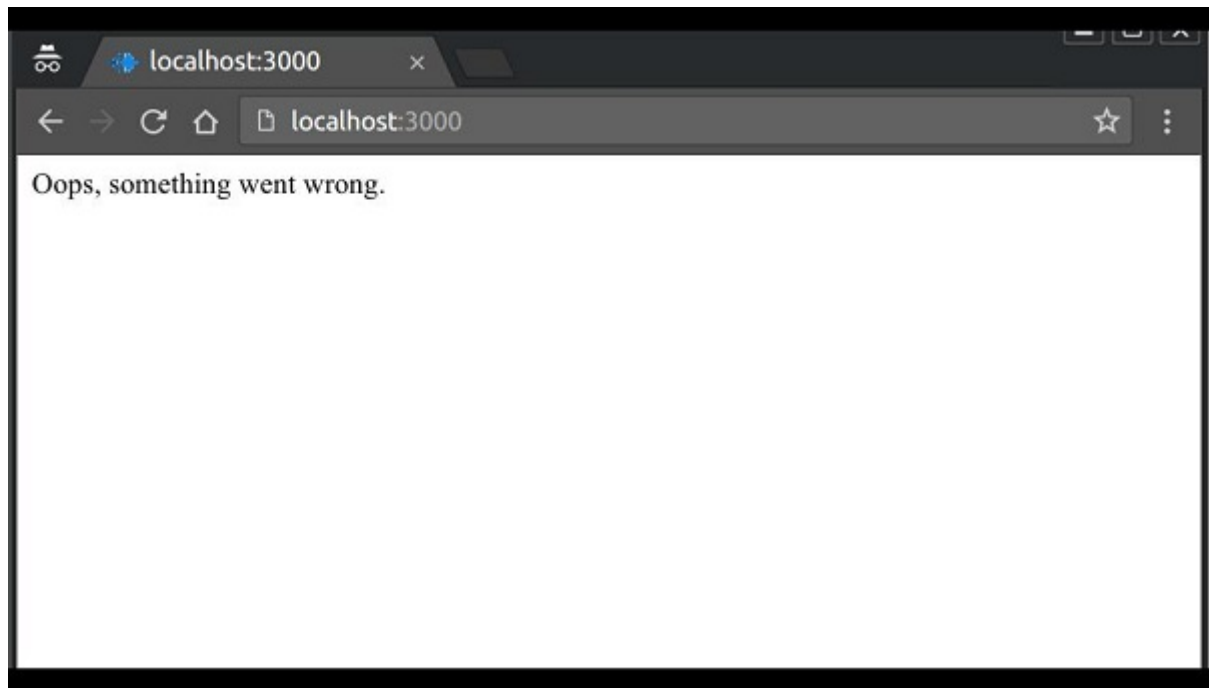
```
app.use(function(err, req, res, next) {  
    console.error(err.stack);  
    res.status(500).send('Something broke!');  
});
```

Till now we were handling errors in the routes itself. The error handling middleware allows us to separate our error logic and send responses accordingly. The next() method we discussed in middleware takes us to next **middleware/route handler**.

For error handling, we have the **next(err)** function. A call to this function skips all middleware and matches us to the next error handler for that route. Let us understand this through an example.

```
var express = require('express');  
var app = express();  
  
app.get('/', function(req, res){  
    //Create an error and pass it to the next function  
    var err = new Error("Something went wrong");  
    next(err);  
});  
  
/*  
 * other route handlers and middleware here  
 * ....  
 */  
  
//An error handling middleware  
app.use(function(err, req, res, next) {  
    res.status(500);  
    res.send("Oops, something went wrong.")  
});  
  
app.listen(3000);
```

This error handling middleware can be strategically placed after routes or contain conditions to detect error types and respond to the clients accordingly. The above program will display the following output.



Advertisements