

Let me keep this brief.

The `next()` function is given to us by the Express framework, and it has two major uses:

### Use case #1: Ending middleware execution

- You can use `next()` to signal to the middleware that the middleware code is finished
- If you don't include `next()`, the browser will continue to hang and 'wait' for the middleware to finish

There is another way to end a middleware function besides calling `next()`. You can also send a response to the client by calling the `res.end()` function.

***Bottom line: You have to end middleware using either `next()` or sending a response back to the browser.***

### Use case #2: Defining errors

The `next()` function allows you to define errors.

If you pass *anything* to the `next()` function (except the word "route"), Express will by default regard the request as being an error. The effect is that the Node parser will STOP any remaining non-error routing code and middleware functions.

**Bottom line: If you pass in an error object into next, `next(err)`, then Express will immediately terminate all routes beneath it and handle the error.**

