

finalhandler

Node.js function to invoke as the final step to respond to HTTP request.

Installation

This is a [Node.js](#) module available through the [npm registry](#). Installation is done using the [`npm install` command](#):

API

Can be used for writing errors to a central location without providing a function to be called with the error when it exists.

options.onerror

variable, but it can be overridden by this option.

By default, the environment is determined by NODE_ENV

options.env

is invoked.

The final handler will also unpipe anything from req when it

- Any headers specified in an error.headers object is produced, otherwise will be err.stack.
- The body will be the HTML of the status code message if env code.

The res.statusMessage is set according to the status range, it will be set to 500. The res.statusCode is set from err.status (or err.statusCode). If this value is outside the 4xx or 5xx the response:

When an error is written, the following information is added to res will be terminated if a response has already started.

If it is truthy, an error response will be written out to the res or err is falsy, the handler will write out a 404 response to the res. Returns function to be invoked as fn(error). If req and res. This function is to be invoked as fn(error). If

finalhandler(req, res, [options])

License

[MIT](#)

excessive function generation. Called as
onerror(err, req, res).

Examples

always 404

```
var finalhandler =
    require('finalhandler')
var http = require('http')

var server = http.createServer(function
    (req, res) {
    var done = finalhandler(req, res)
    done()
})

server.listen(3000)
```

perform simple action

```
var finalhandler =
    require('finalhandler')
var fs = require('fs')
var http = require('http')
```

```
keep log of all errors
ar finalhandller = require('finalhandler')
ar fs = require('fs')
ar http = require('http')
ar server = http.createServer(function(req, res) {
  var done = finalhandler(req, res, {
    onerror: logger })
  var server = http.createServer(function(req, res, {
```

```
var server = http.createServer(function() {
    var done = finalhandler(red, res)
    var serve = serveStatic('public')
    var static = require('serve-static')
    var http = require('http')
    var finalhandler = require('finalhandler')
    var serveStatic = require('serve-static')
    var server = http.createServer(function(req, res) {
        if (err) return done(err)
        fs.readFile('index.html', function(err, buf) {
            if (err) return done(err)
            res.setHeader('Content-Type', 'text/html')
            res.end(buf)
        })
    })
    server.listen(3000)
})
```