

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

finalhandler(req, res, [options])

Returns function to be invoked as the final step for the given req and res. This function is to be invoked as fn(err). If err is falsy, the handler will write out a 404 response to the res. If it is truthy, an error response will be written out to the res or res will be terminated if a response has already started. When an error is written, the following information is added to the response:

- The res.statusCode is set from err.status (or err.statusCode). If this value is outside the 4xx or 5xx range, it will be set to 500.
 - The res.statusMessage is set according to the status code.
 - The body will be the HTML of the status code message if env is 'production', otherwise will be err.stack.
 - Any headers specified in an err.headers object.
- The final handler will also unpipe anything from req when it is invoked.

options.env

By default, the environment is determined by NODE_ENV variable, but it can be overridden by this option.

options.onerror

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

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')
```

```

var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res)
  if (err) return done(err)
  res.setHeader('Content-Type', 'text/html')
  res.end(buf)
})
server.listen(3000)

```

use with middleware-style functions

```

var finalhandler =
  require('finalhandler')
var http = require('http')
var serveStatic = require('serve-static')

var serve = serveStatic('public')

var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res)
  serve(req, res, done)
})
server.listen(3000)

```

4

keep log of all errors

```

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

var server = http.createServer(function (req, res) {
  var done = finalhandler(req, res, {
    onerror: logerror
  })
  fs.readFile('index.html', function (err, buf) {
    if (err) return done(err)
    res.setHeader('Content-Type', 'text/html')
    res.end(buf)
  })
})

function logerror (err) {
  console.error(err.stack) ||
  err.toString())
}

server.listen(3000)

```

5