**https://www.cnblogs.com/jokerjason/p/7815040.html**

NodeJs中Express框架使用morgan中间件记录日志

npm install –g express

npm install –g express-generator ‘ express.version> 4.0 , we should install express-generator together.

Express中的app.js文件已经默认引入了该中间件var logger = require('morgan');

使用app.use(logger('dev'));可以将请求信息打印在控制台，便于开发调试，但实际生产环境中，需要将日志记录在log文件里，可以使用如下代码

[复制代码](javascript:void(0);)

var express = require('express');

var fs = require('fs');

var logger = require('morgan');

var app = express();

var accessLog = fs.createWriteStream('../access.log', {flags : 'a'});

var errorLog = fs.createWriteStream('../error.log', {flags : 'a'});

app.use(logger('dev')); //打印到控制台

app.use(logger('combined', {stream : accessLog})); //打印到log日志

[复制代码](javascript:void(0);)

**这样便可以将请求信息打印在根目录下的access.log文件中（注意文件路径别填错，并不会自动创建该文件）**

示例：

**express/connect**

Simple app that will log all request in the Apache combined format to STDOUTq

Log出所有Apache请求（结合STTDOUT格式）

[复制代码](javascript:void(0);)

var express = require('express')

var morgan = require('morgan')

var app = express()

app.use(morgan('combined'))

app.get('/', function (req, res) {

res.send('hello, world!')

})

[复制代码](javascript:void(0);)

**vanilla http server**

Simple app that will log all request in the Apache combined format to STDOUT

[复制代码](javascript:void(0);)

var finalhandler = require('finalhandler')

var http = require('http')

var morgan = require('morgan')

// create "middleware"

var logger = morgan('combined')

http.createServer(function (req, res) {

var done = finalhandler(req, res)

logger(req, res, function (err) {

if (err) return done(err)

// respond to request

res.setHeader('content-type', 'text/plain')

res.end('hello, world!')

})

})

[复制代码](javascript:void(0);)

**write logs to a file**

**single file**

Simple app that will log all requests in the Apache combined format to the file access.log.

Log列出所有Apache请求到access.log中

[复制代码](javascript:void(0);)

var express = require('express')

var fs = require('fs')

var morgan = require('morgan')

var path = require('path')

var app = express()

// create a write stream (in append mode)

var accessLogStream = fs.createWriteStream(path.join(\_\_dirname, 'access.log'), {flags: 'a'})

// setup the logger

app.use(morgan('combined', {stream: accessLogStream}))

app.get('/', function (req, res) {

res.send('hello, world!')

})

[复制代码](javascript:void(0);)

**log file rotation**

Simple app that will log all requests in the Apache combined format to one log file per day in the log/ directory using the [rotating-file-stream module](https://www.npmjs.com/package/rotating-file-stream).

每天将所有Apache请求记录到log中

[复制代码](javascript:void(0);)

var express = require('express')

var fs = require('fs')

var morgan = require('morgan')

var path = require('path')

var rfs = require('rotating-file-stream')

var app = express()

var logDirectory = path.join(\_\_dirname, 'log')

// ensure log directory exists

fs.existsSync(logDirectory) || fs.mkdirSync(logDirectory)

// create a rotating write stream

var accessLogStream = rfs('access.log', {

interval: '1d', // rotate daily

path: logDirectory

})

// setup the logger

app.use(morgan('combined', {stream: accessLogStream}))

app.get('/', function (req, res) {

res.send('hello, world!')

})

[复制代码](javascript:void(0);)

**split / dual logging**

The morgan middleware can be used as many times as needed, enabling combinations like:

* Log entry on request and one on response
* Log all requests to file, but errors to console
* ... and more!

Sample app that will log all requests to a file using Apache format, but error responses are logged to the console:

Log所有Apache请求，错误Log到console中

[复制代码](javascript:void(0);)

var express = require('express')

var fs = require('fs')

var morgan = require('morgan')

var path = require('path')

var app = express()

// log only 4xx and 5xx responses to console

app.use(morgan('dev', {

skip: function (req, res) { return res.statusCode < 400 }

}))

// log all requests to access.log

app.use(morgan('common', {

stream: fs.createWriteStream(path.join(\_\_dirname, 'access.log'), {flags: 'a'})

}))

app.get('/', function (req, res) {

res.send('hello, world!')

})

[复制代码](javascript:void(0);)

**use custom token formats**

Sample app that will use custom token formats. This adds an ID to all requests and displays it using the :id token.

ID Token格式

[复制代码](javascript:void(0);)

var express = require('express')

var morgan = require('morgan')

var uuid = require('node-uuid')

morgan.token('id', function getId (req) {

return req.id

})

var app = express()

app.use(assignId)

app.use(morgan(':id :method :url :response-time'))

app.get('/', function (req, res) {

res.send('hello, world!')

})

function assignId (req, res, next) {

req.id = uuid.v4()

next()

}

[复制代码](javascript:void(0);)