

Con console.log

DEBUG CONSOLEPROBLEMS2OUTPUTGITLENSTERMINAL

Windows PowerShell  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
  
Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>  
  
PS D:\backend\_javascript\Entrega 14\entrega\_14\_logging> npm test  
  
> @1.0.0 test D:\backend\_javascript\Entrega 14\entrega\_14\_logging  
> node benchmark.js  
  
Running benchmarks in Parallel  
Running 20s test @ http://localhost:3001/info  
100 connections  
  

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	1080 ms	1169 ms	1593 ms	1834 ms	1220.67 ms	180.2 ms	1920 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	91	114	79.25	32.59	1
Bytes/Sec	0 B	0 B	188 kB	235 kB	163 kB	67.2 kB	2.06 kB

  
Req/Bytes counts sampled once per second.  
# of samples: 20  
  
2k requests in 20.22s, 3.27 MB read  
PS D:\backend\_javascript\Entrega 14\entrega\_14\_logging>

DevTools - Node.js

ProfilerConsoleConnectionSourcesMemory

Heavy (Bottom Up)

Profiles

CPU PROFILES

Profile 1

Saved

	Self Time	Total Time	Function
31797.0 ms	63.17 %	31797.0 ms	63.17 %
4022.5 ms	7.99 %	4022.5 ms	7.99 %
3924.4 ms	7.80 %	7554.9 ms	15.01 %
1759.4 ms	3.50 %	1759.4 ms	3.50 %
324.5 ms	0.64 %	324.5 ms	0.64 %
295.6 ms	0.59 %	295.6 ms	0.59 %
244.6 ms	0.49 %	1181.4 ms	2.35 %
216.0 ms	0.43 %	321.4 ms	0.64 %
187.2 ms	0.37 %	256.0 ms	0.51 %
162.0 ms	0.32 %	598.7 ms	1.19 %
147.1 ms	0.29 %	147.1 ms	0.29 %
137.3 ms	0.27 %	12147.9 ms	24.13 %
133.8 ms	0.27 %	133.8 ms	0.27 %
129.9 ms	0.26 %	474.4 ms	0.94 %
125.2 ms	0.25 %	85890.6 ms	170.64 %
97.0 ms	0.19 %	578.4 ms	1.15 %
86.7 ms	0.17 %	12670.9 ms	25.17 %
80.2 ms	0.16 %	1545.4 ms	3.07 %
80.0 ms	0.16 %	100.8 ms	0.20 %
78.7 ms	0.16 %	304.7 ms	0.61 %
75.7 ms	0.15 %	244.0 ms	0.48 %
74.9 ms	0.15 %	74.9 ms	0.15 %
69.6 ms	0.14 %	76.9 ms	0.15 %
69.6 ms	0.14 %	10558.9 ms	20.98 %
67.9 ms	0.13 %	67.9 ms	0.13 %
66.9 ms	0.13 %	202.5 ms	0.40 %
62.5 ms	0.12 %	6732.6 ms	13.38 %
58.8 ms	0.12 %	114.0 ms	0.23 %
57.7 ms	0.11 %	12767.6 ms	25.37 %
56.0 ms	0.11 %	657.2 ms	1.31 %
54.8 ms	0.11 %	54.8 ms	0.11 %
54.7 ms	0.11 %	54.7 ms	0.11 %
53.7 ms	0.11 %	53.7 ms	0.11 %
53.7 ms	0.11 %	180.7 ms	0.36 %
51.8 ms	0.10 %	51.8 ms	0.10 %

source-node.js:221

parse.js:395

source-node.js:172

parse.js:201

code-ops.js:118

authenticate.js:94

javascript-controller.js:223

index.js:177

code-ops.js:38

index.js:179

javascript-controller.js:454

math.js:130

javascript-controller.js:994

javascript-controller.js:1210

internal/tls.js:102

internal/sockets.js:105

initServer.js:104

utils.js:18

index.js:596

express-handlebars.js:5

internal/crypto/hash.js:39

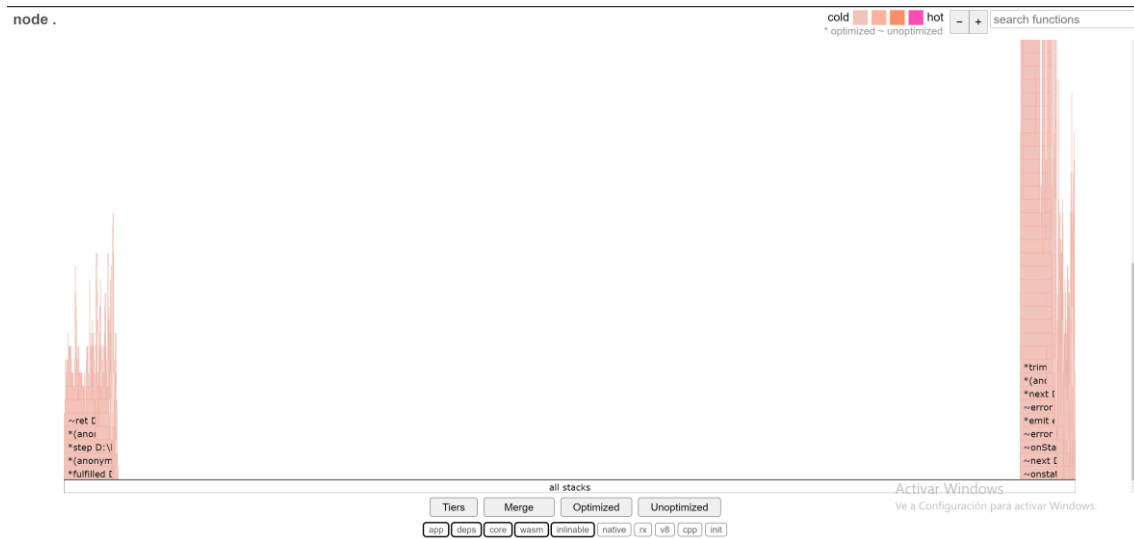
json.js:98

base.js:11

common.js:48

Activar Windows  
Ve a Configuración para activar Windows

```
102     });
103
104     app.get('/info', [(req, res) => {
105         const renderInfo = {
106             argumentosEntrada: process.argv.slice(2),
107             nombrePlataforma: process.platform,
108             versionNode: process.version,
109             memoriaTotalReservada: process.memoryUsage().rss,
110             pathEjecucion: process.execPath,
111             idProceso: process.pid,
112             carpetaProyecto: process.cwd(),
113             numeroProcesadoresDisponibles: os.cpus().length
114         };
115
116         console.log(renderInfo);
117
118         res.render('info', {process: process, rss: process.memoryUsage().rss, argv: process.argv.slice(2), processors: os.cpus().length});
119     });
120
```



Test corrido con la configuración que correspondía a Artillery

```
PS D:\backend_javascript\Entrega 14\entrega_14_logging> npm test
```

```
> @1.0.0 test D:\backend_javascript\Entrega 14\entrega_14_logging
> node benchmark.js
```

Running benchmarks in Parallel  
Running 20s test @ http://localhost:3001/info  
50 connections with 20 pipelining factor

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	5748 ms	5894 ms	6783 ms	6808 ms	6074.42 ms	346 ms	6822 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	0	1000	150	307	113
Bytes/Sec	0 B	0 B	0 B	2.06 MB	309 kB	633 kB	233 kB

Req/Bytes counts sampled once per second.  
# of samples: 20

4k requests in 20.23s, 6.19 MB read

```
PS D:\backend_javascript\Entrega 14\entrega_14_logging> []
```

```
JS initServer.js M result-prof-blog.txt U X
result-prof-blog.txt
1 Statistical profiling result from .\isolate-000001CDC0774810-13460-v8.log, (8554 ticks, 4 unaccounted, 0 excluded).
2
3 [Shared libraries]:
4 ticks total nonlib name
5 7381 86.3% C:\WINDOWS\SYSTEM32\ntdll.dll
6 1872 12.5% C:\Program Files\nodejs\node.exe
7 8 0.1% C:\WINDOWS\System32\KERNELBASE.dll
8 2 0.0% C:\WINDOWS\System32\KERNEL32.DLL
9 1 0.0% C:\WINDOWS\System32\WS2_32.dll
10 1 0.0% C:\WINDOWS\System32\ADVAPI32.dll
11
12 [JavaScript]:
13 ticks total nonlib name
14 8 0.1% 9.0% LazyCompile: "parse D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\handlebars\compiler\parser.js:269:30
15 6 0.1% 6.7% LazyCompile: "SourceNode_walk D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-map\lib\source-node.js:221:53
16 5 0.1% 5.6% LazyCompile: "next D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\handlebars\compiler\parser.js:478:32
17 4 0.0% 4.5% LazyCompile: "resolve path.js:130:10
18 3 0.0% 3.4% RegExp: (?:(\n?|\n).)*
19 3 0.0% 3.4% LazyCompile: "_generator D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express-handlebars\dist\express-handlebars.js:27:58
20 3 0.0% 3.4% LazyCompile: "SourceNode D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-map\lib\source-node.js:35:20
21 2 0.0% 2.2% LazyCompile: "wrap D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\handlebars\compiler\code-gen.js:101:22
22 2 0.0% 2.2% LazyCompile: "toJSON D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express-session\session\cookie.js:147:19
23 2 0.0% 2.2% LazyCompile: "next D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express\lib\router\index.js:177:16
24 2 0.0% 2.2% LazyCompile: "formatProperty internal/util\inspect.js:1674:24
25 2 0.0% 2.2% LazyCompile: "MustacheStatement D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\handlebars\compiler\compiler
26 2 0.0% 2.2% LazyCompile: "<anonymous> D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-map\lib\source-node.js:320:22
27 1 0.0% 1.1% RegExp: ^(?:\{(?:\{(?:\{
28 1 0.0% 1.1% RegExp: ^(?:\{(?:\{(?:\{
29 1 0.0% 1.1% RegExp: ^(?:\{(?:\{(?:\{
30 1 0.0% 1.1% RegExp: \/*
```

Sin console.log

```
PS D:\backend_javascript\Entrega 14\entrega_14_logging> npm test
```

```
> @1.0.0 test D:\backend_javascript\Entrega 14\entrega_14_logging
> node benchmark.js
```

Running benchmarks in Parallel

Running 20s test @ http://localhost:3001/info  
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	450 ms	552 ms	1209 ms	1258 ms	646.83 ms	191.38 ms	1286 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	100	100	153	200	154.9	46.89	100
Bytes/Sec	206 kB	206 kB	316 kB	412 kB	319 kB	96.6 kB	206 kB

Req/Bytes counts sampled once per second.

# of samples: 20

3k requests in 20.27s, 6.39 MB read

```
PS D:\backend_javascript\Entrega 14\entrega_14_logging> █
```

DevTools - Node.js

Profiler Console Connection Sources Memory

Heavy (Bottom Up)

Profiles

CPU PROFILES

Profile 1 2s

Self Time	Total Time	Function
21739.6 ms	53.43 %	(program)
1770.9 ms	4.35 %	getCPUs
1726.3 ms	4.24 %	writeUtf8String
598.5 ms	1.47 %	stat
544.6 ms	1.34 %	(garbage collector)
496.5 ms	1.22 %	SourceNode.walk
428.6 ms	1.05 %	next
371.7 ms	0.91 %	SourceNode.add
304.3 ms	0.75 %	parse
304.0 ms	0.75 %	quotedString
289.9 ms	0.71 %	writev
283.9 ms	0.70 %	createFunctionContext
267.2 ms	0.66 %	authenticate
239.7 ms	0.59 %	next
179.8 ms	0.44 %	wrap
167.9 ms	0.41 %	session
155.2 ms	0.38 %	resolve
151.8 ms	0.37 %	compile
146.2 ms	0.36 %	replaceStack
144.0 ms	0.35 %	hash
143.7 ms	0.35 %	setUpHelperArgs
128.8 ms	0.32 %	(anonymous)
125.4 ms	0.31 %	extend
118.0 ms	0.29 %	step
112.3 ms	0.28 %	jsonParser
108.9 ms	0.27 %	FSReqCallback
105.9 ms	0.26 %	nextTick
97.9 ms	0.24 %	open
93.9 ms	0.23 %	randomBytes
92.9 ms	0.23 %	writeHead
91.5 ms	0.22 %	store.generate
91.2 ms	0.22 %	send
90.0 ms	0.22 %	(anonymous)
89.7 ms	0.22 %	parseWithoutProcessing
85.7 ms	0.21 %	render

source-node.js:221  
parser.js:396  
source-node.js:172  
parser.js:201  
code-gen.js:118  
javascript-controller.js:223  
authenticate.js:94  
index.js:177  
code-gen.js:98  
index.js:179  
path.js:130  
javascript-controller.js:64  
javascript-controller.js:994  
index.js:596  
javascript-controller.js:1210  
utils.js:18  
express-handlebars.js:5  
json.js:98  
internal/streams/streams.js:105  
http-server.js:243  
index.js:158  
response.js:111  
express-handlebars.js:221  
base.js:11  
response.js:1016

Activar Windows  
Ve a Configuración para activar Windows.

```
103
104 29.2 ms app.get('/info', (req, res) => {
105
106    // const renderInfo = {
107    //   argumentosEntrada: process.argv.slice(2),
108    //   nombrePlataforma: process.platform,
109    //   versionNode: process.version,
110    //   memoriaTotalReservada: process.memoryUsage().rss,
111    //   pathEjecucion: process.execPath,
112    //   idProceso: process.pid,
113    //   carpetaProyecto: process.cwd(),
114    //   numeroProcesadoresDisponibles: os.cpus().length
115    // };
116
117    // console.log(renderInfo);
118 34.6 ms res.render('info', {process: process, rss: process.memoryUsage().rss, argv: process.argv.slice(2), processors: os.cpus().length});
119
120 });
```

For help, see: <https://nodejs.org/en/docs/inspector>  
Cantidad de procesadores en Fork: 8  
Server is up and running on port 3001  
Database connected  
PS D:\backend\_javascript\Entrega 14\entrega\_14\_logging> npm test

```
> @1.0.0 test D:\backend_javascript\Entrega 14\entrega_14_logging
> node benchmark.js
```

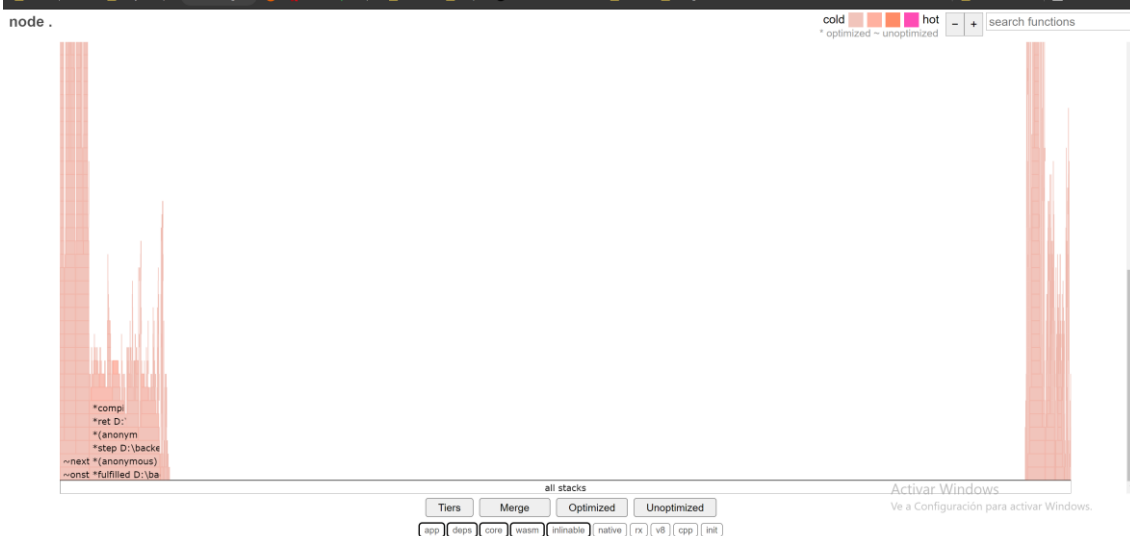
Running benchmarks in Parallel  
Running 20s test @ http://localhost:3001/info  
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	222 ms	312 ms	432 ms	449 ms	316.16 ms	52.53 ms	509 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	200	200	300	451	312.8	51.05	200
Bytes/Sec	412 kB	412 kB	619 kB	930 kB	645 kB	105 kB	412 kB

Req/Bytes counts sampled once per second.  
# of samples: 20

6k requests in 20.14s, 12.9 MB read  
PS D:\backend\_javascript\Entrega 14\entrega\_14\_logging>



Test corrido con la configuración que correspondía a Artillery

```
DEBUG CONSOLE PROBLEMS OUTPUT GITLENS TERMINAL
```

```
PS D:\backend_javascript\Entrega 14\entrega_14_logging> npm test
```

```
> @1.0.0 test D:\backend_javascript\Entrega 14\entrega_14_logging
> node benchmark.js
```

```
Running benchmarks in Parallel
Running 20s test @ http://localhost:3001/info
50 connections with 20 pipelining factor
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	2866 ms	3097 ms	3789 ms	3833 ms	3168.62 ms	248.62 ms	3867 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	238	1000	300	317.63	238
Bytes/Sec	0 B	0 B	491 kB	2.06 MB	618 kB	655 kB	491 kB

```
Req/Bytes counts sampled once per second.
# of samples: 20

7k requests in 20.18s, 12.4 MB read
PS D:\backend_javascript\Entrega 14\entrega_14_logging> 
```

```

1  result-prof-no-blog.txt
2  [Statistical profiling result from .\isolate-000001f2c2b77b70-8088-v8.log, (5799 ticks, 7 unaccounted, 0 excluded).]
3
4  [Shared libraries]:
5      ticks  total  nonlib   name
6      4536   78.2%
7      1106   19.1%  C:\WINDOWS\SYSTEM32\ntdll.dll
8      5       0.1%  C:\Program Files\nodejs\node.exe
9      3       0.1%  C:\WINDOWS\System32\KERNELBASE.dll
10     3       0.1%  C:\WINDOWS\System32\KERNEL32.DLL
11
12 [JavaScript]:
13     ticks  total  nonlib   name
14     15     0.3%  10.1% LazyCompile: *parse D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs
15     12     0.2%   8.1% LazyCompile: *next D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\
16     11     0.2%   7.4% LazyCompile: *wrap D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist\cjs\
17     10     0.2%   6.7% LazyCompile: *resolve path.js:130:10
18     5       0.1%   3.4% LazyCompile: *ret_setup D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dis
19     5       0.1%   3.4% LazyCompile: *fulfilled D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express-handle
20     4       0.1%   2.7% LazyCompile: *SourceNode_walk D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-ma
21     3       0.1%   2.0% LazyCompile: *normalize path.js:265:12
22     3       0.1%   2.0% LazyCompile: *onwrite_stream_writable.js:431:17
23     3       0.1%   2.0% LazyCompile: *generateList D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\c
24     3       0.1%   2.0% LazyCompile: *anonymous D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\dist
25     3       0.1%   2.0% LazyCompile: *SourceNode D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-map\lib
26     2       0.0%   1.3% RegExp: /^(?:\\(?:\\|
27     2       0.0%   1.3% LazyCompile: *toNamespacedPath path.js:549:19
28     2       0.0%   1.3% LazyCompile: *parse D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\minimatch\minimatch
29     2       0.0%   1.3% LazyCompile: *appendEscaped D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\handlebars\
30     2       0.0%   1.3% LazyCompile: *<anonymous> D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\source-map\li
31     2       0.0%   1.3% LazyCompile: *<anonymous> D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express-hand
32     1       0.0%   0.7% LazyCompile: *<anonymous> D:\backend_javascript\Entrega 14\entrega_14_logging\node_modules\express-hand
33     1       0.0%   0.7% RegExp: /^(?:undefined(?:[~(){}]))
34     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
35     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
36     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
37     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
38     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
39     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
40     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
41     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
42     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
43     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
44     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
45     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
46     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
47     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
48     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
49     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
50     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
51     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
52     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
53     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
54     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
55     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
56     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
57     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
58     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
59     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
60     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
61     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
62     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
63     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
64     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
65     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
66     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
67     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
68     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
69     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
70     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
71     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
72     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
73     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
74     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
75     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
76     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
77     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
78     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
79     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
80     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
81     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
82     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
83     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
84     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
85     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
86     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
87     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
88     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
89     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
90     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
91     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
92     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
93     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
94     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
95     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
96     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
97     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
98     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
99     1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
100    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
101    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
102    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
103    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
104    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
105    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
106    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
107    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
108    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
109    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
110    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
111    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
112    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
113    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
114    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
115    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
116    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
117    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
118    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
119    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
120    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
121    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
122    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
123    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
124    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
125    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
126    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
127    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
128    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
129    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
130    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
131    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
132    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
133    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
134    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
135    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
136    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
137    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
138    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
139    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
140    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
141    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
142    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
143    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
144    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
145    1       0.0%   0.7% RegExp: /^(?:\\(?:\\|
146    1       
```

**Conclusión:** En el caso de incluir un `console.log` en la ruta, el nivel de carga aumenta muchísimo su tiempo de respuesta con respecto a cuando no se incluye, tal como se puede ver en los valores devueltos por consola, inspector de Chrome y archivo de flames.