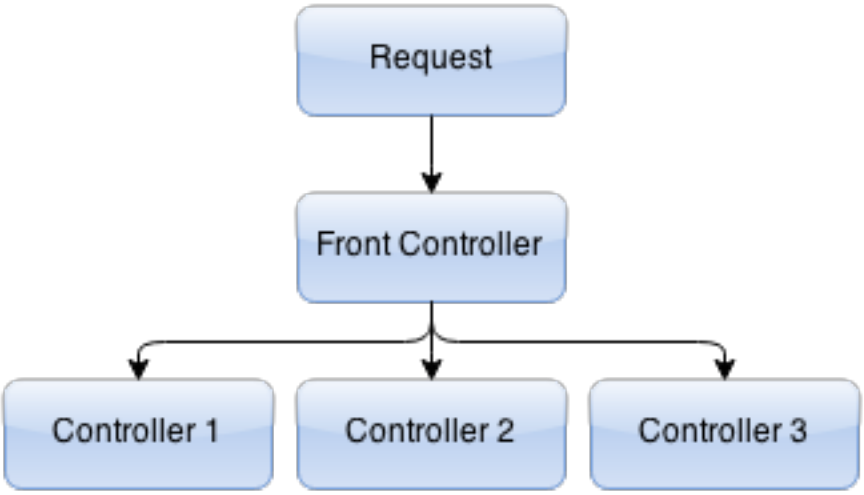
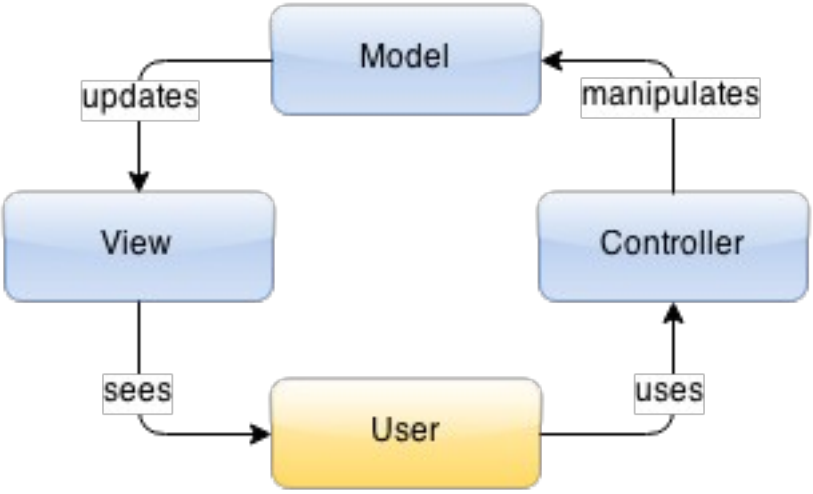
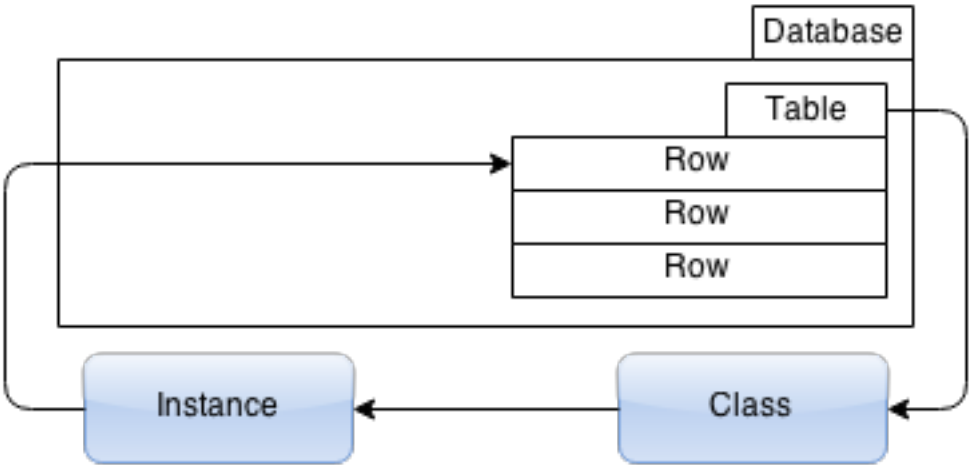
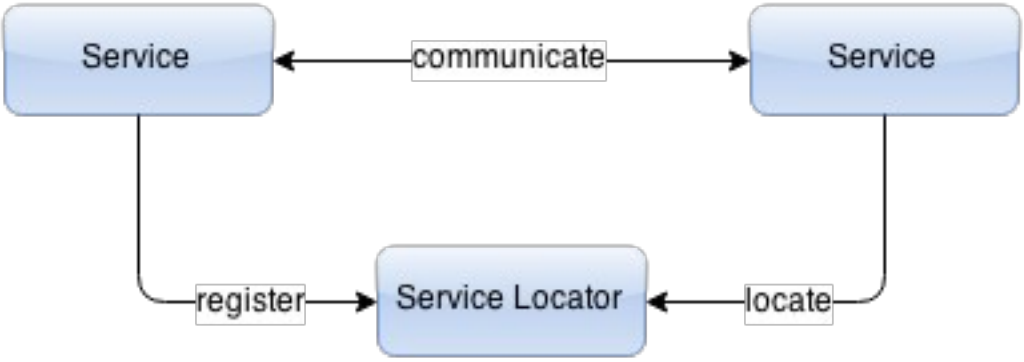
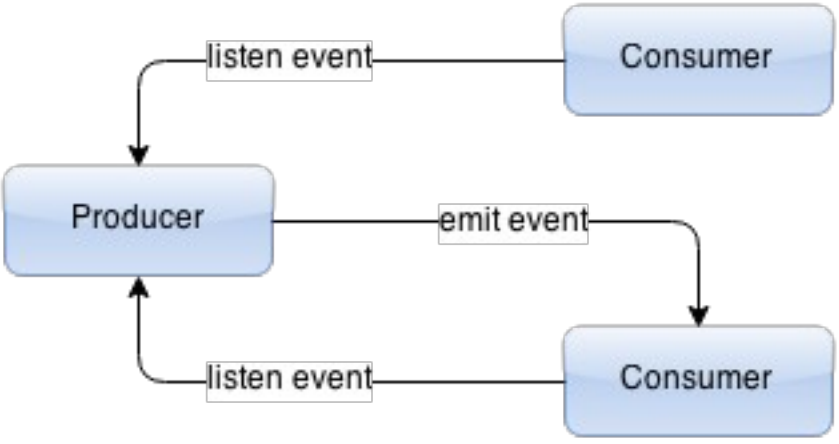
# Chapter 2: Development Patterns

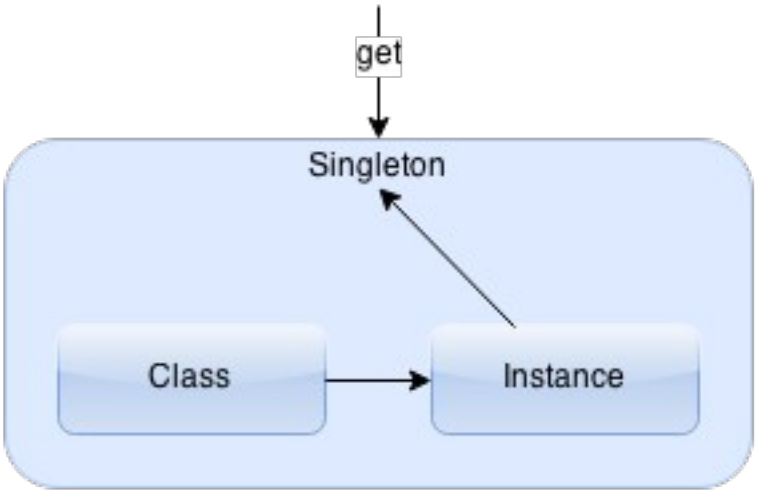


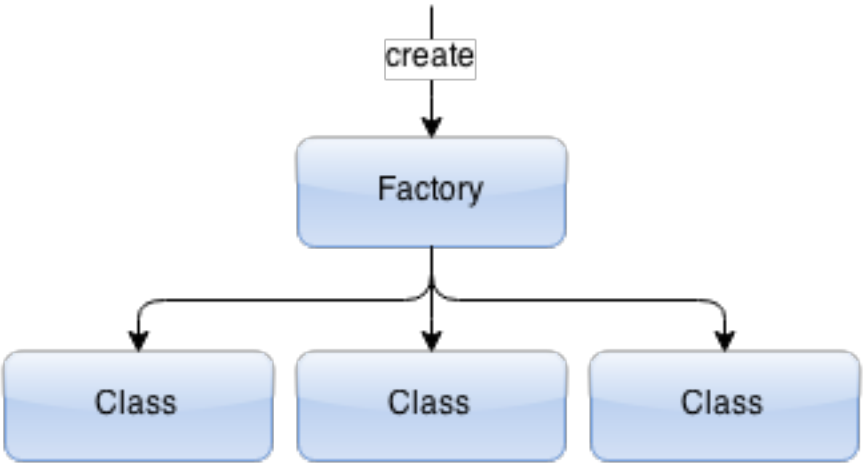



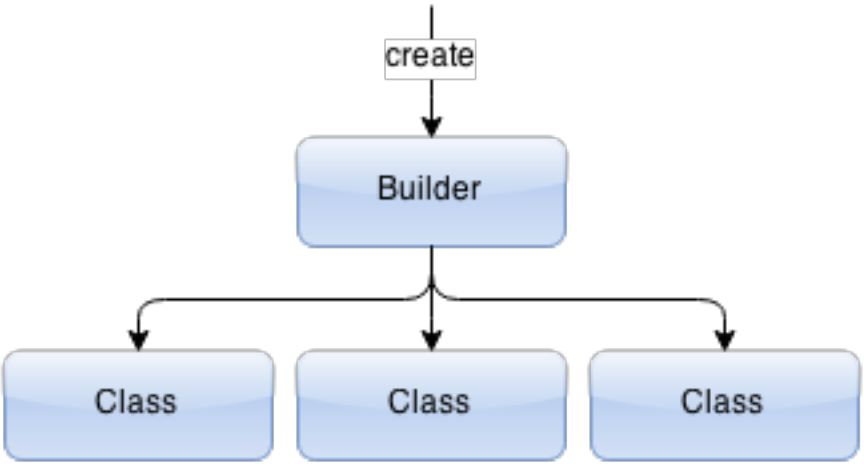

Database
Table
Row
Row
Row
Instance
Class

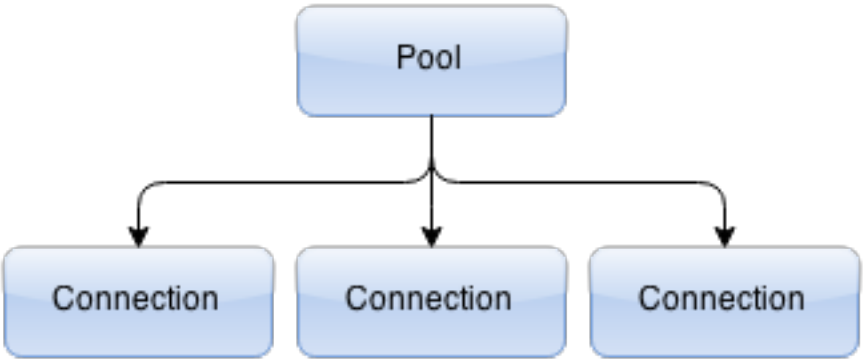

Service
communicate
Service
register
Service Locator
locate


listen event
Consumer
Producer
emit event
listen event
Consumer

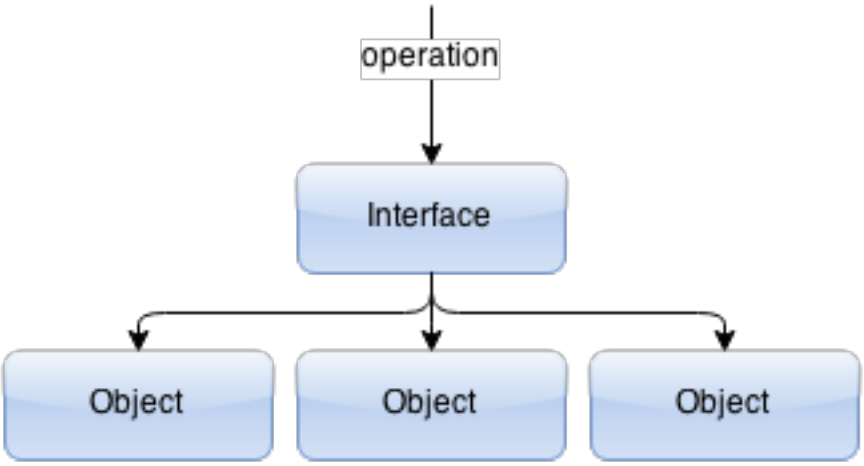

get
Singleton
Class
Instance

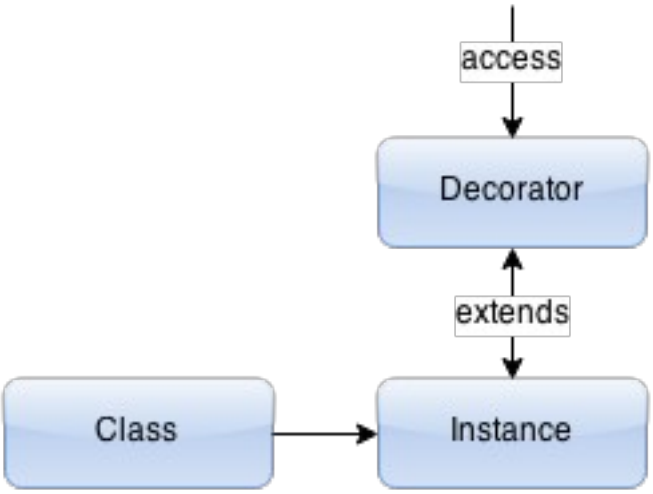

create
Factory
Class
Class
Class

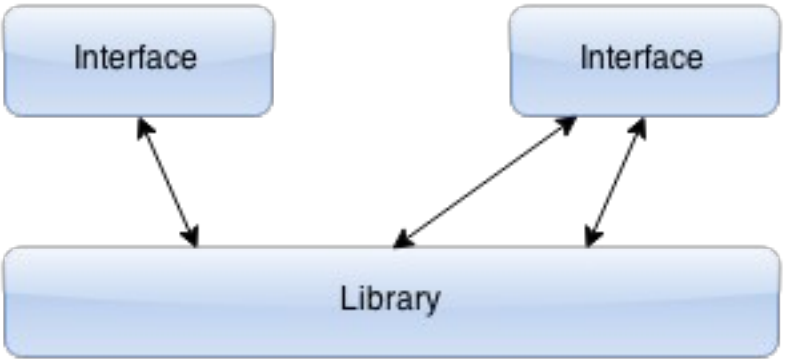

create
Builder
Class
Class
Class

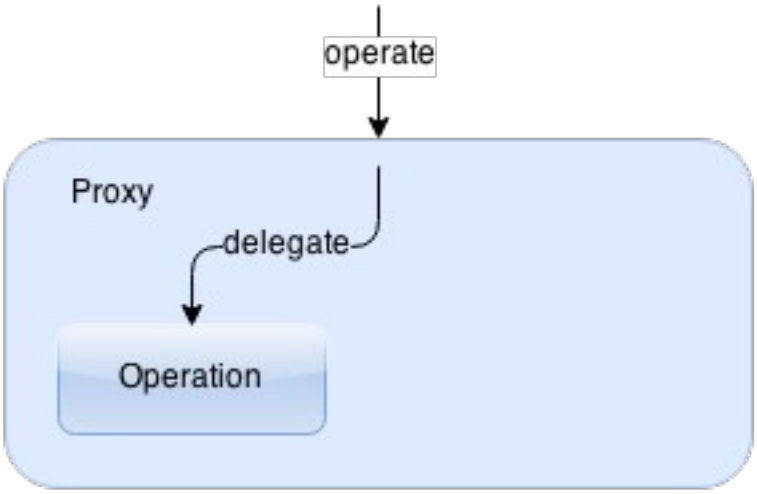

Pool
Connection
Connection
Connection

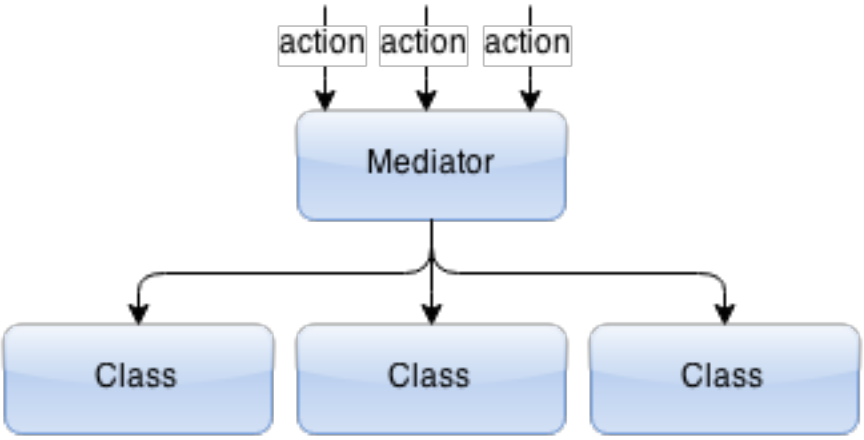

Interface
Adapter
Interface


operation
Interface
Object
Object
Object

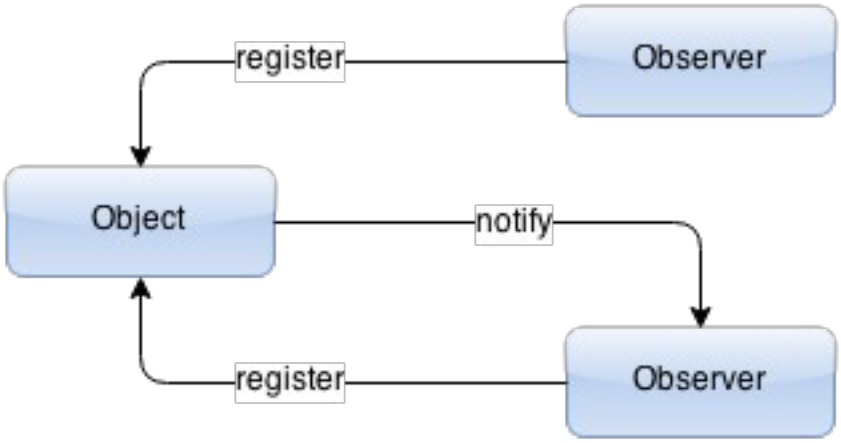

access
Decorator
extends
Class
Instance


Interface
Interface
Library

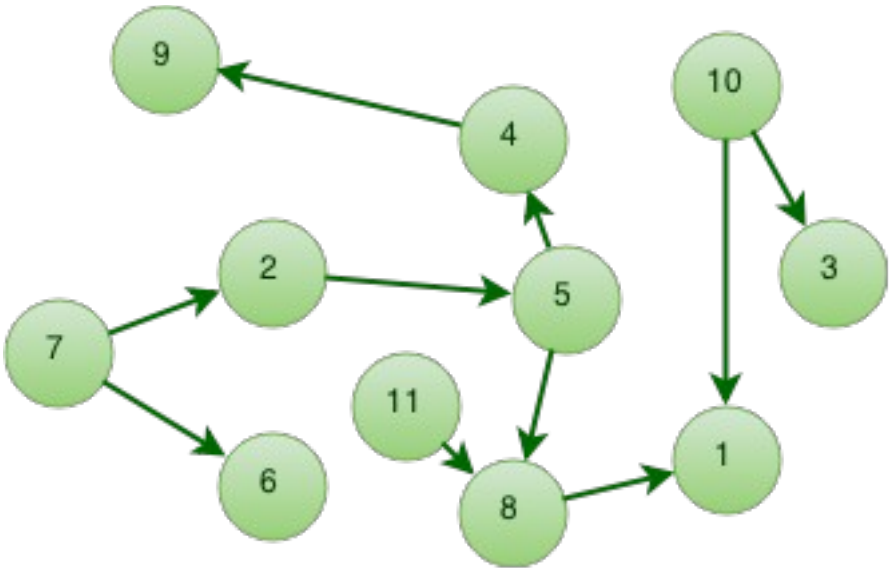

operate
Proxy
delegate
Operation

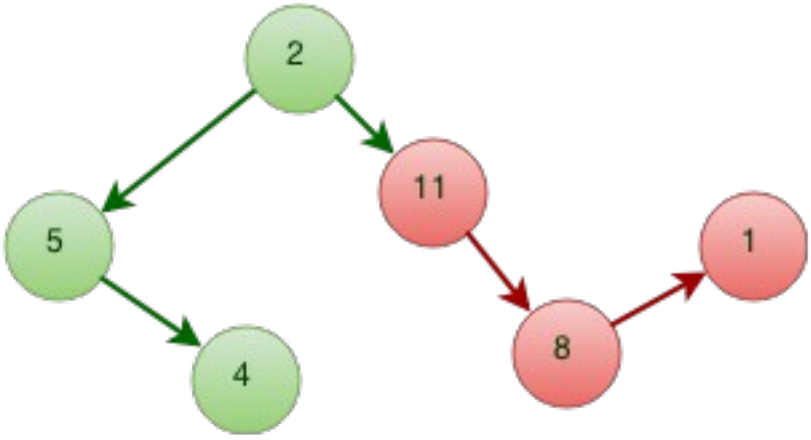

action
action
action
Mediator
Class
Class
Class

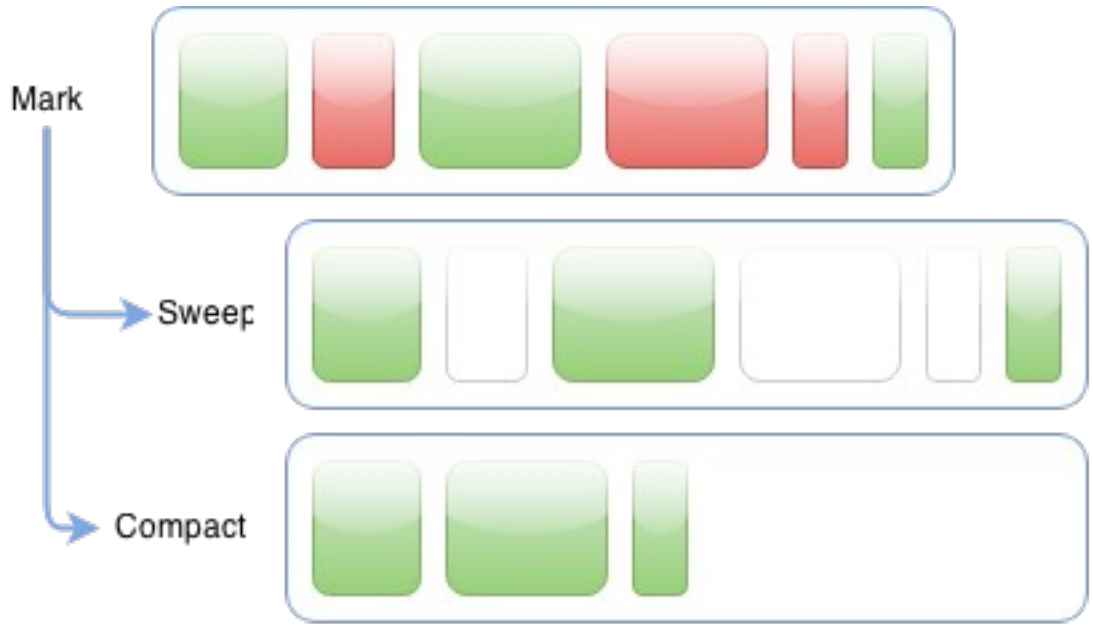

Template
compile
Pre-compiled Template
render
Output

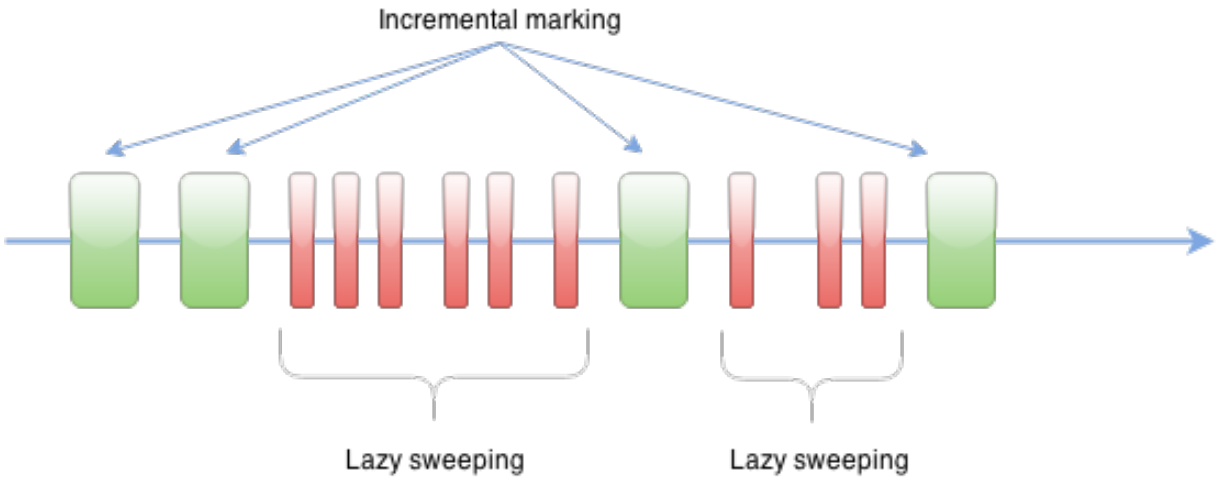

register
Observer
Object
notify
register
Observer

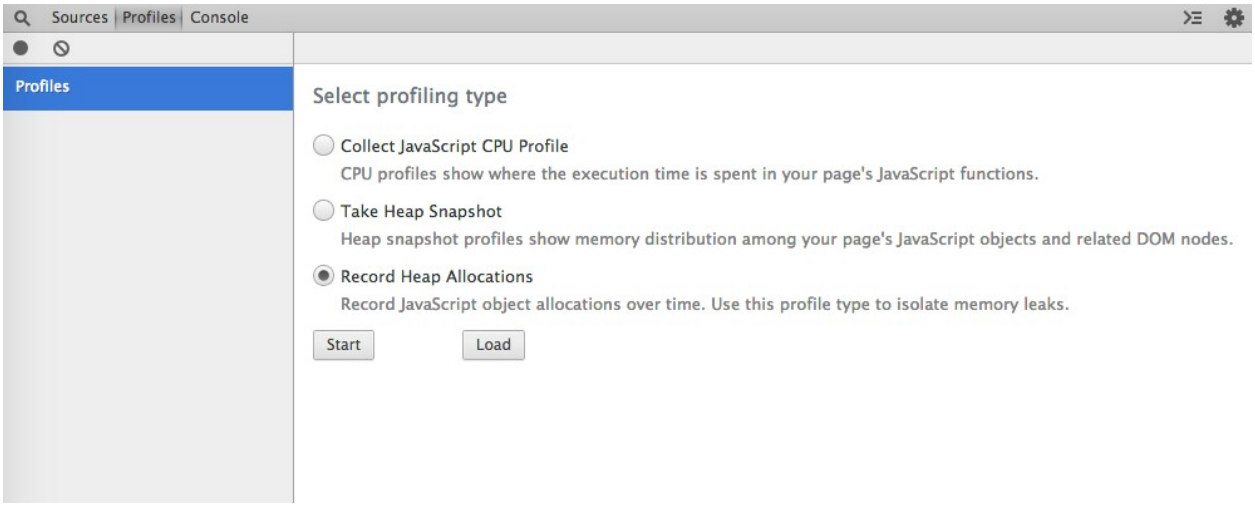
# Chapter 3: Garbage Collection

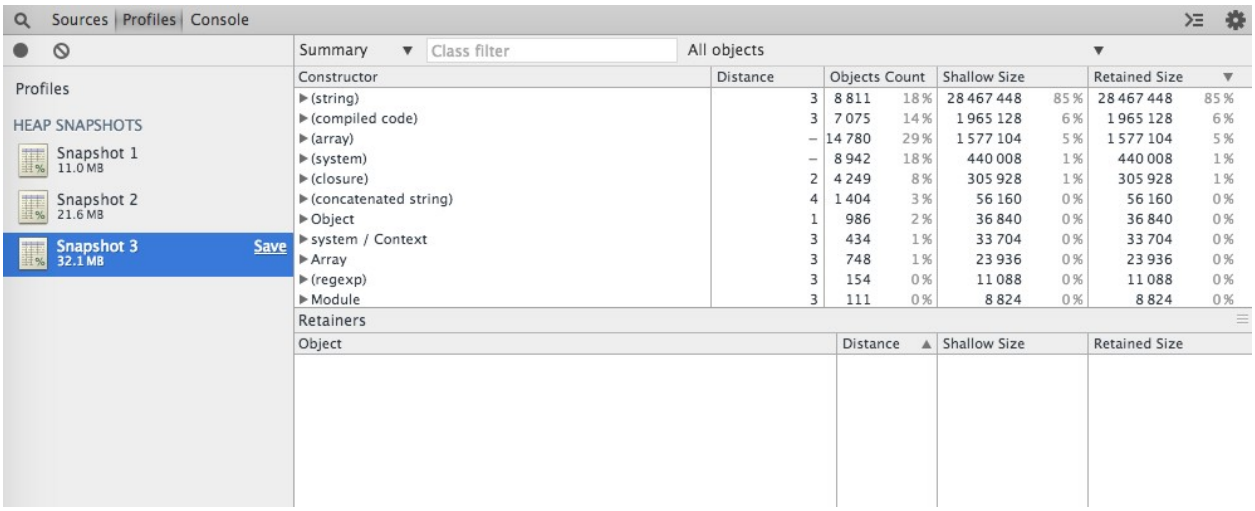


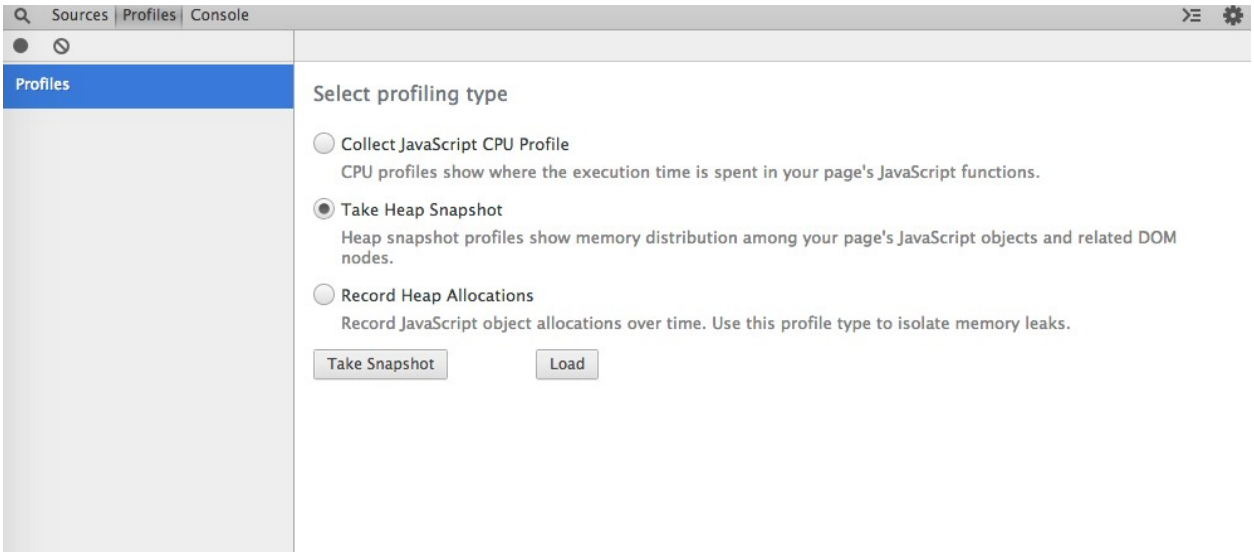



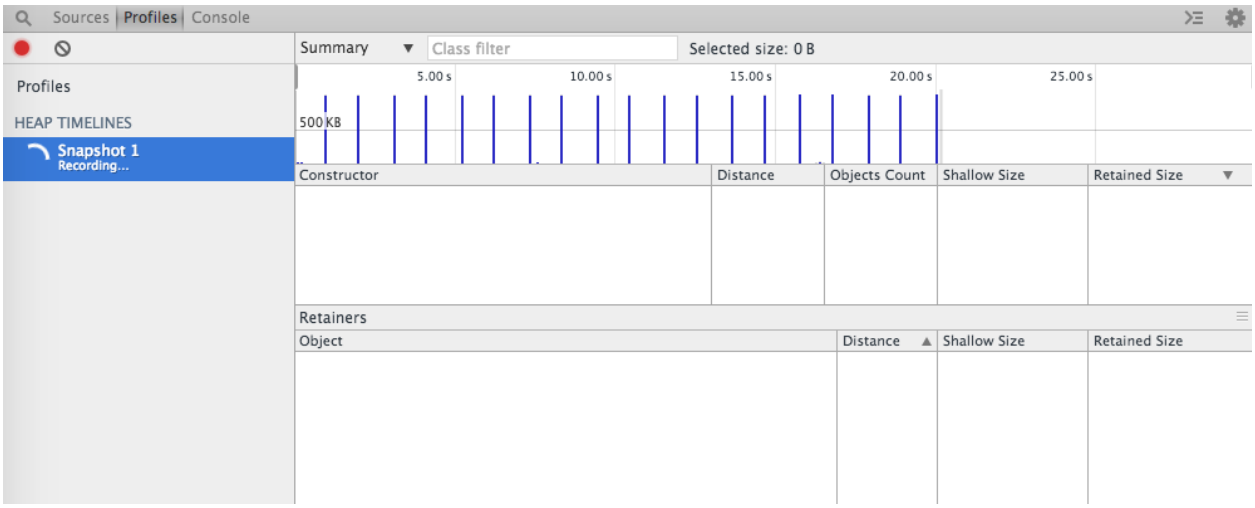

Mark
Sweep
Compact

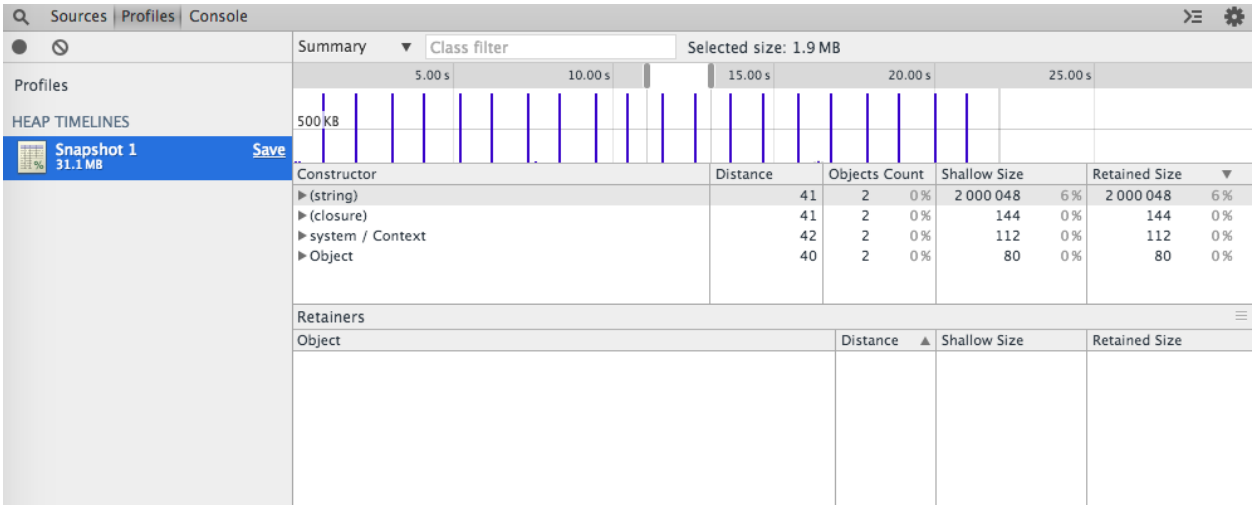

Incremental marking
Lazy sweeping
Lazy sweeping

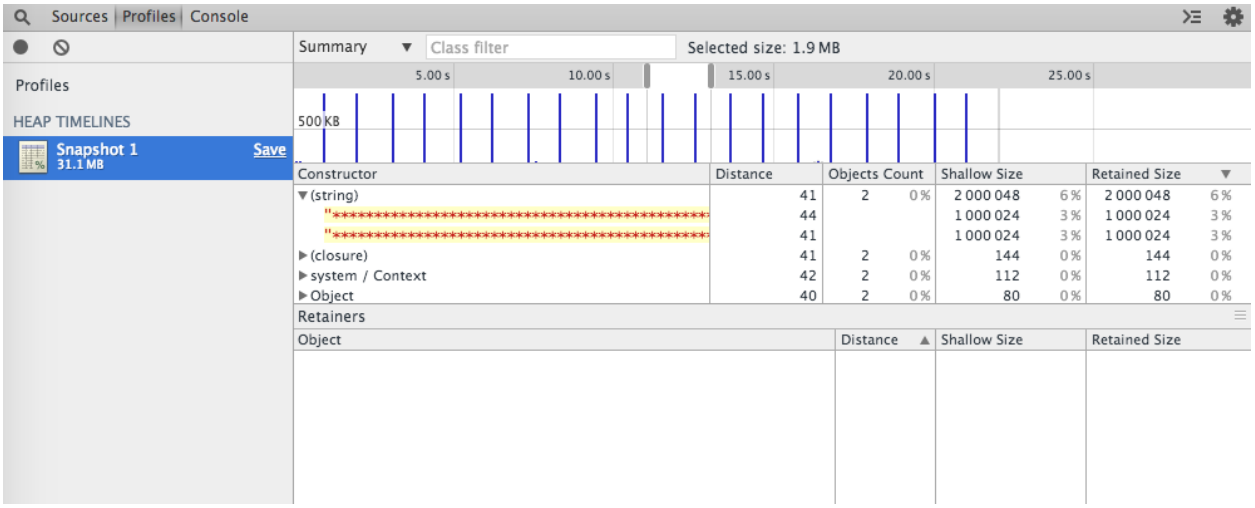

Sources
Profiles
Console
Profiles
Select profiling type
Collect JavaScript CPU Profile
CPU profiles show where the execution time is spent in your page's JavaScript functions.
Take Heap Snapshot
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
Record Heap Allocations
Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.
Start
Load

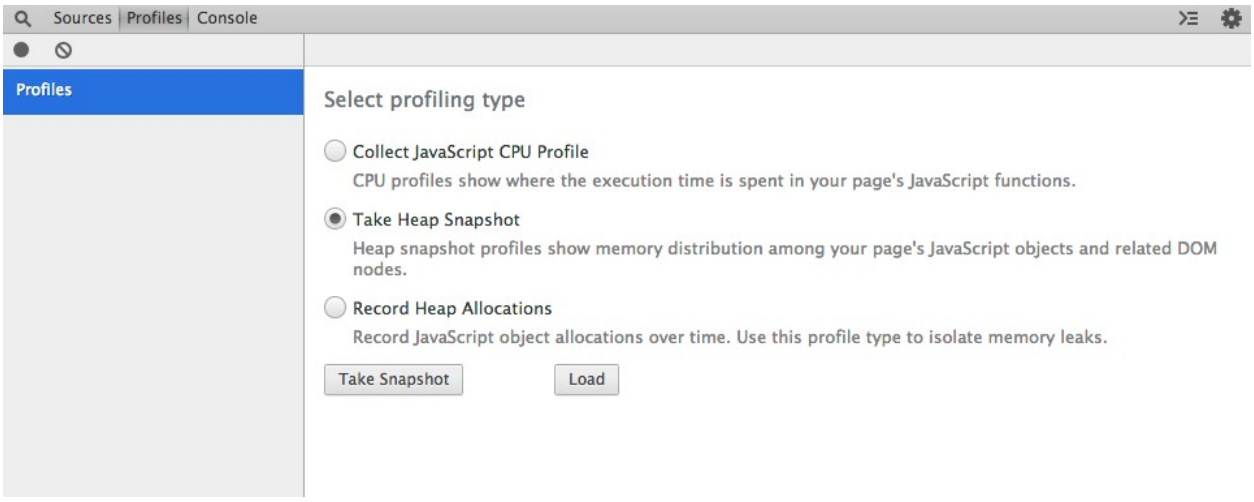

Sources
Profiles
Console
Profiles
HEAP SNAPSHOTS
Snapshot 1
11.0 MB
Snapshot 2
21.6 MB
Snapshot 3
32.1 MB
Save
Summary
Class filter
All objects
Constructor
Distance
Objects Count
Shallow Size
Retained Size
(string) 3 8 811 18% 28 467 448 85% 28 467 448 85%
(compiled code) 3 7 075 14% 1 965 128 6% 1 965 128 6%
(array) – 14 780 29% 1 577 104 5% 1 577 104 5%
(system) – 8 942 18% 440 008 1% 440 008 1%
(closure) 2 4 249 8% 305 928 1% 305 928 1%
(concatenated string) 4 1 404 3% 56 160 0% 56 160 0%
Object 1 986 2% 36 840 0% 36 840 0%
system / Context 3 434 1% 33 704 0% 33 704 0%
Array 3 748 1% 23 936 0% 23 936 0%
(regexp) 3 154 0% 11 088 0% 11 088 0%
Module 3 111 0% 8 824 0% 8 824 0%
Retainers
Object
Distance
Shallow Size
Retained Size


Sources
Profiles
Console
Profiles
Select profiling type
Collect JavaScript CPU Profile
CPU profiles show where the execution time is spent in your page's JavaScript functions.
Take Heap Snapshot
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
Record Heap Allocations
Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.
Take Snapshot
Load

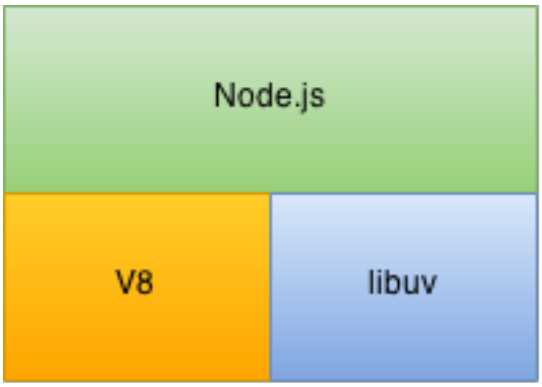

Sources
Profiles
Console
Summary
Class filter
Selected size: 0 B
Profiles
HEAP TIMELINES
Snapshot 1
Recording...
5.00 s
10.00 s
15.00 s
20.00 s
25.00 s
500 KB
Constructor
Distance
Objects Count
Shallow Size
Retained Size
Retainers
Object
Distance
Shallow Size
Retained Size

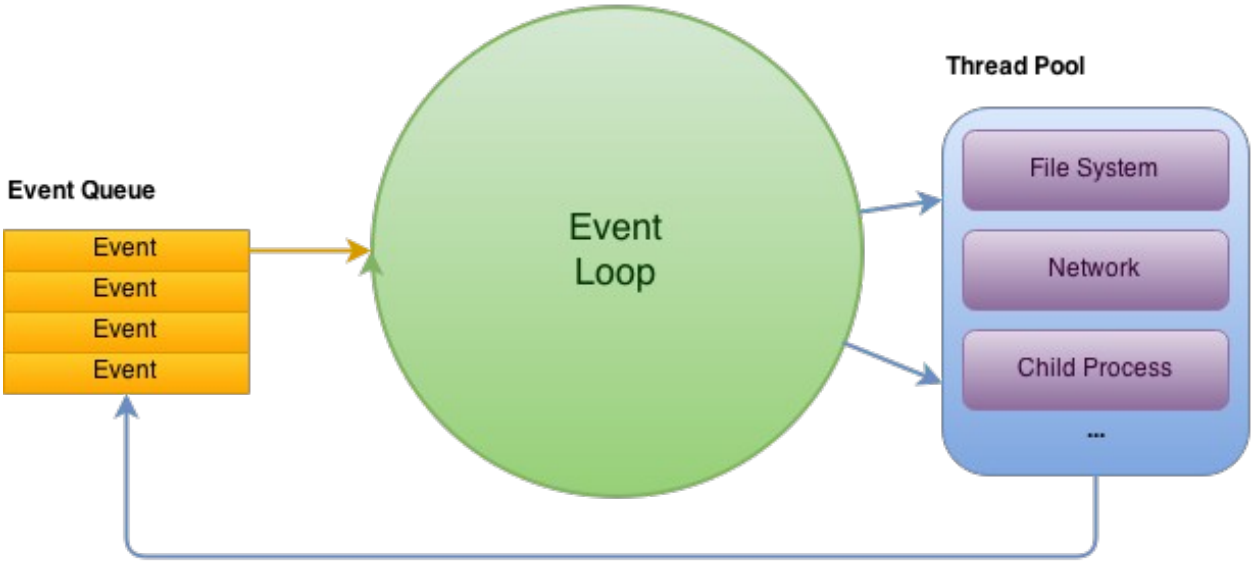

Sources
Profiles
Console
Summary
Class filter
Selected size: 1.9 MB
Profiles
HEAP TIMELINES
Snapshot 1
31.1 MB
Save
5.00 s
10.00 s
15.00 s
20.00 s
25.00 s
500 KB
Constructor
Distance
Objects Count
Shallow Size
Retained Size
(string) 41 2 0% 2 000 048 6% 2 000 048 6%
(closure) 41 2 0% 144 0% 144 0%
system / Context 42 2 0% 112 0% 112 0%
Object 40 2 0% 80 0% 80 0%
Retainers
Object
Distance
Shallow Size
Retained Size


Sources
Profiles
Console
Summary
Class filter
Selected size: 1.9 MB
Profiles
HEAP TIMELINES
Snapshot 1
31.1 MB
Save
5.00 s
10.00 s
15.00 s
20.00 s
25.00 s
500 KB
Constructor
Distance
Objects Count
Shallow Size
Retained Size
(string) 41 2 0% 2 000 048 6% 2 000 048 6%
" 44 1 000 024 3% 1 000 024 3%
" 41 1 000 024 3% 1 000 024 3%
(closure) 41 2 0% 144 0% 144 0%
system / Context 42 2 0% 112 0% 112 0%
Object 40 2 0% 80 0% 80 0%
Retainers
Object
Distance
Shallow Size
Retained Size


Sources
Profiles
Console
Profiles
Select profiling type
Collect JavaScript CPU Profile
CPU profiles show where the execution time is spent in your page's JavaScript functions.
Take Heap Snapshot
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
Record Heap Allocations
Record JavaScript object allocations over time. Use this profile type to isolate memory leaks.
Take Snapshot
Load

# Chapter 4: CPU Profiling



V8 Tick Processor

Diogo

v8.googlecode.com/svn/trunk/tools/tick-processor.html

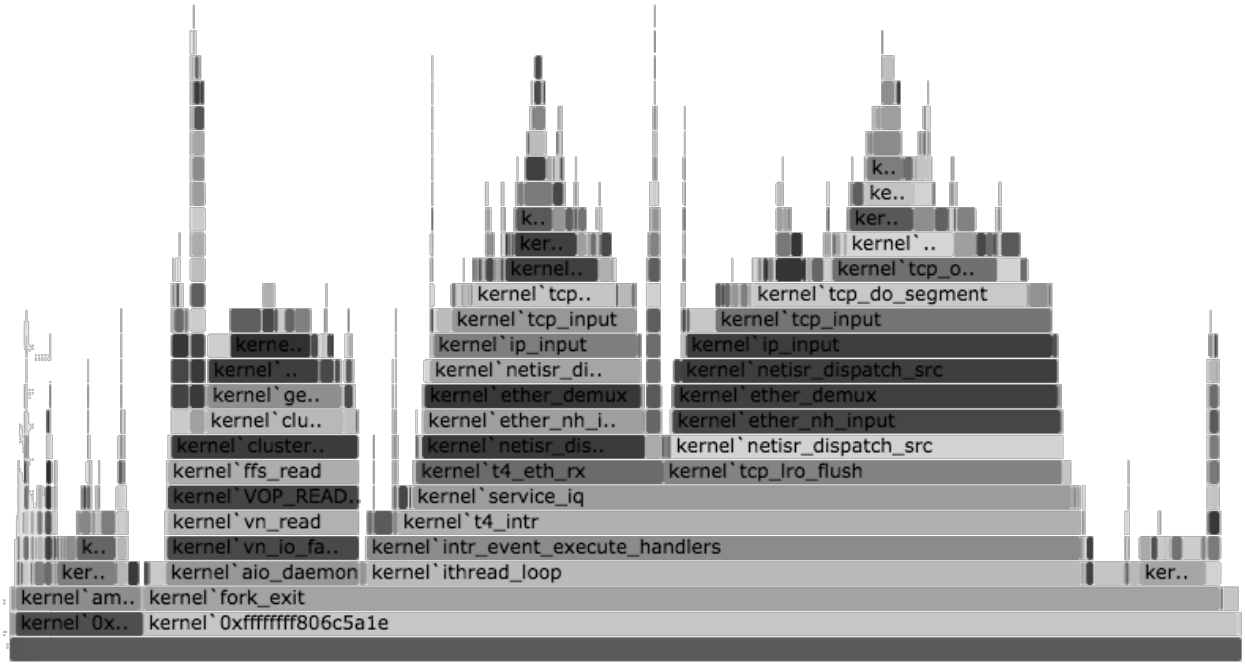
## Chrome V8 profiling log processor

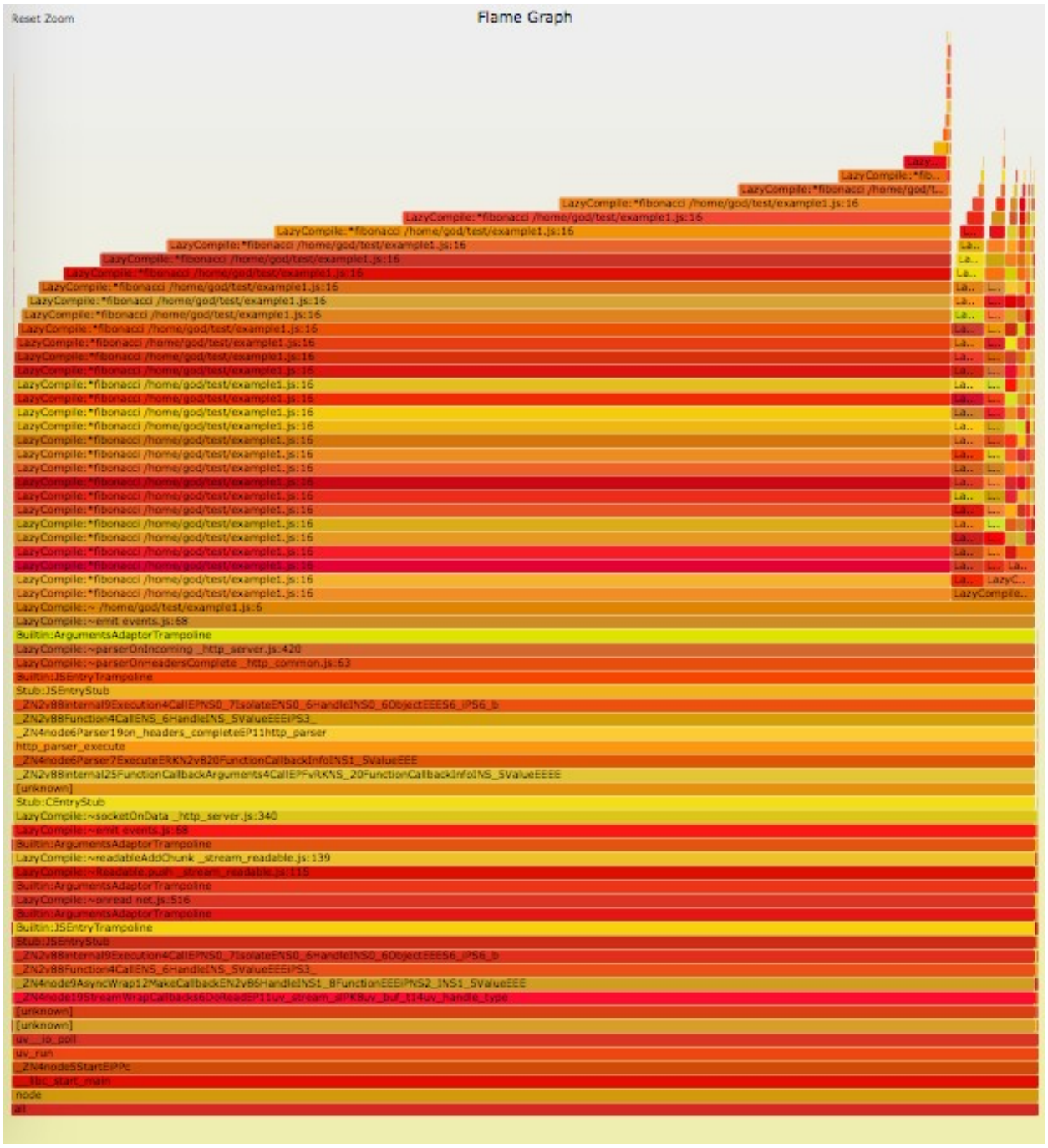
Process V8's profiling information log (sampling profiler tick information) in your browser. Particularly useful if you don't have the V8 shell (d8) at hand on your system. You still have to run Chrome with the appropriate command line flags to produce the profiling log.

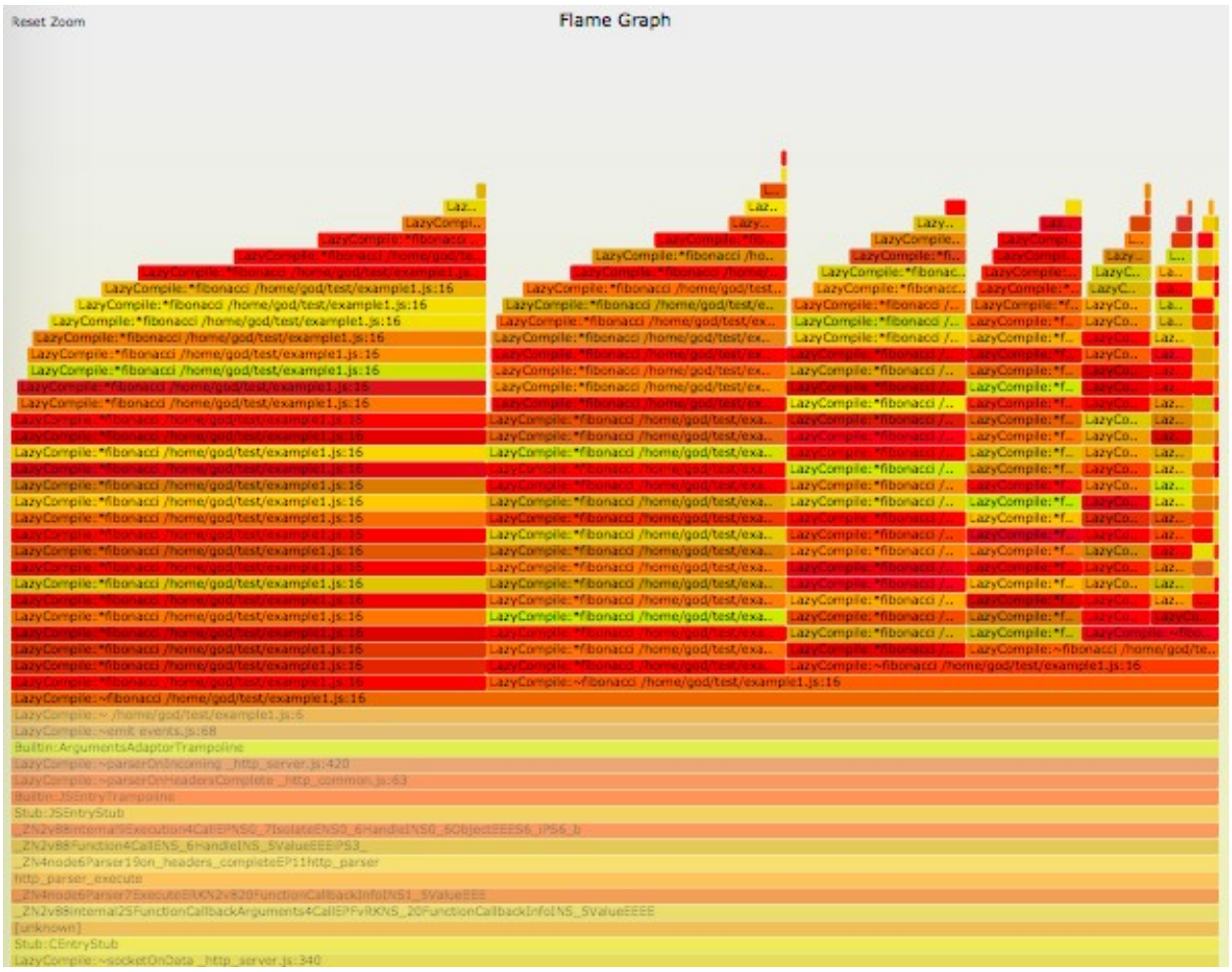
**Usage:**
Click on the button and browse to the profiling log file (usually, v8.log). Process will start automatically and the output will be visible in the below text area.

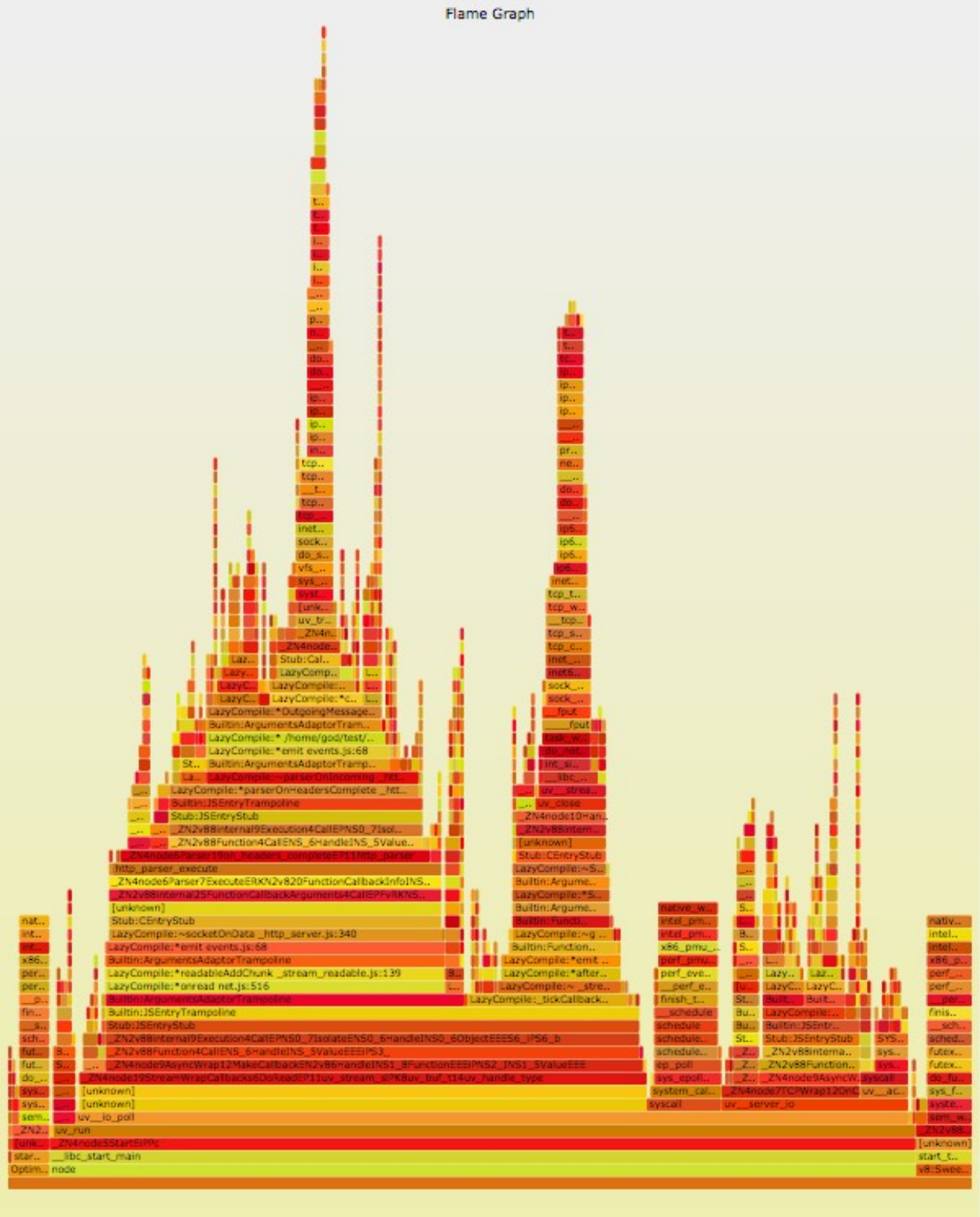
**Limitations and disclaimer:**
This page offers a subset of the functionalities of the command-line tick processor utility in the V8 repository. In particular, this page cannot access the command-line utility that provides library symbol information, hence the [C++] section of the output stays empty. Also consider that this web-based tool is provided only for convenience and quick reference, you should refer to the command-line version for full output.

Choose File No file chosen

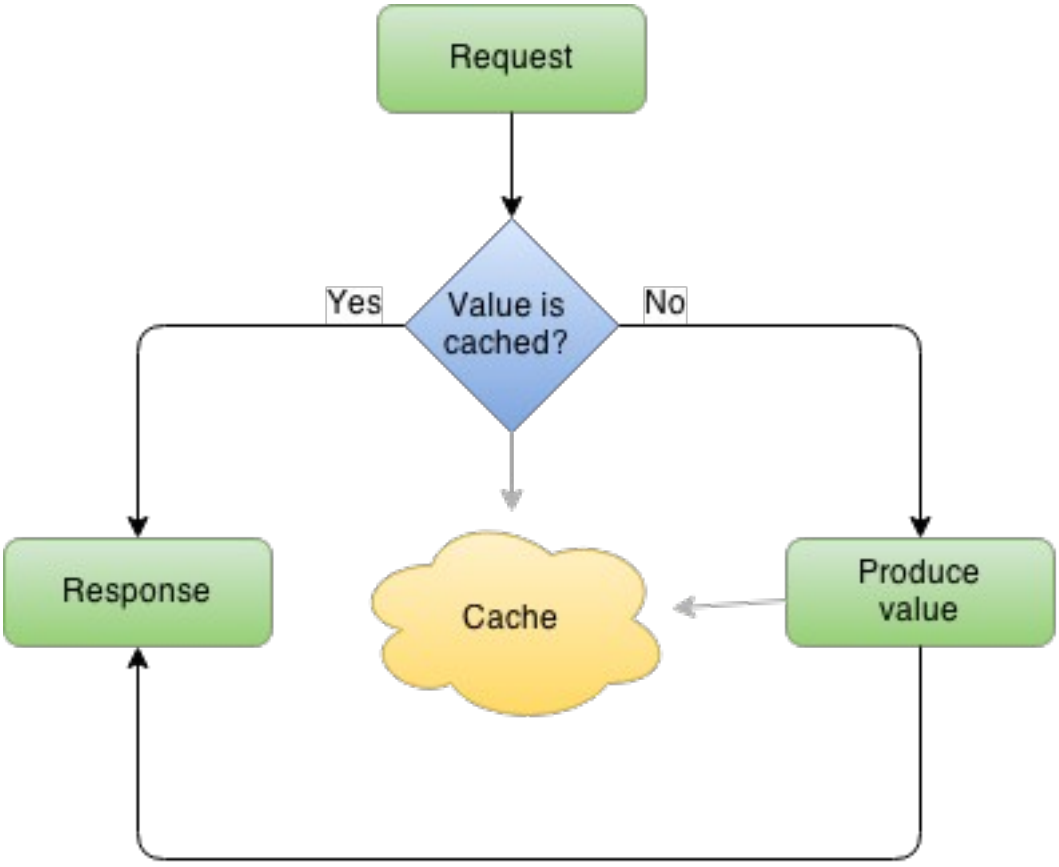


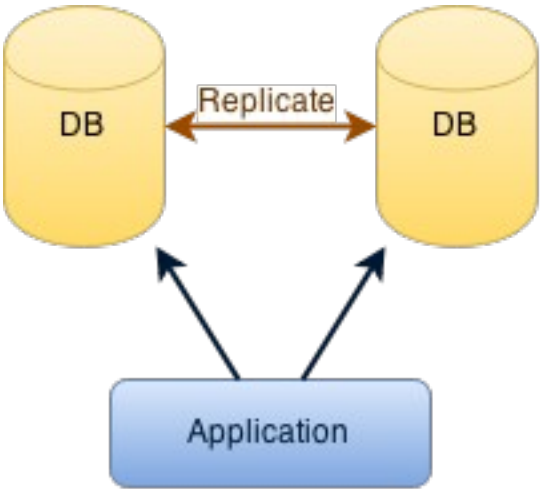

Reset Zoom
Flame Graph
LazyCompile:*fibonacci /home/god/test/example1.js:16
LazyCompile:~ /home/god/test/example1.js:6
LazyCompile:~emit events.js:68
Builtin:ArgumentsAdaptorTrampoline
LazyCompile:~parserOnIncoming _http_server.js:420
LazyCompile:~parserOnHeadersComplete _http_common.js:63
Builtin:JSEntryTrampoline
Stub:JSEntryStub
_ZN2v88internal9Execution4CallEPNS0_7IsolateENS0_6HandleINS0_6ObjectEEES6_iPS6_b
_ZN2v88Function4CallENS_6HandleINS_5ValueEEEiPS3_
_ZN4node6Parser19on_headers_completeEP11http_parser
http_parser_execute
_ZN4node6Parser7ExecuteERKN2v820FunctionCallbackInfoINS1_5ValueEEE
_ZN2v88internal25FunctionCallbackArguments4CallEPFvRKNS_20FunctionCallbackInfoINS_5ValueEEEE
[unknown]
Stub:CEntryStub
LazyCompile:~socketOnData _http_server.js:340
LazyCompile:~emit events.js:68
Builtin:ArgumentsAdaptorTrampoline
LazyCompile:~readableAddChunk _stream_readable.js:139
LazyCompile:~Readable.push _stream_readable.js:115
Builtin:ArgumentsAdaptorTrampoline
LazyCompile:~onread net.js:516
Builtin:ArgumentsAdaptorTrampoline
Builtin:JSEntryTrampoline
Stub:JSEntryStub
_ZN2v88internal9Execution4CallEPNS0_7IsolateENS0_6HandleINS0_6ObjectEEES6_iPS6_b
_ZN2v88Function4CallENS_6HandleINS_5ValueEEEiPS3_
_ZN4node9AsyncWrap12MakeCallbackEN2v86HandleINS1_8FunctionEEEiPNS2_INS1_5ValueEEE
_ZN4node19StreamWrapCallbacks6DoReadEP11uv_stream_siPK8uv_buf_t14uv_handle_type
[unknown]
[unknown]
uv__io_poll
uv_run
_ZN4node5StartEPPc
__libc_start_main
node
at

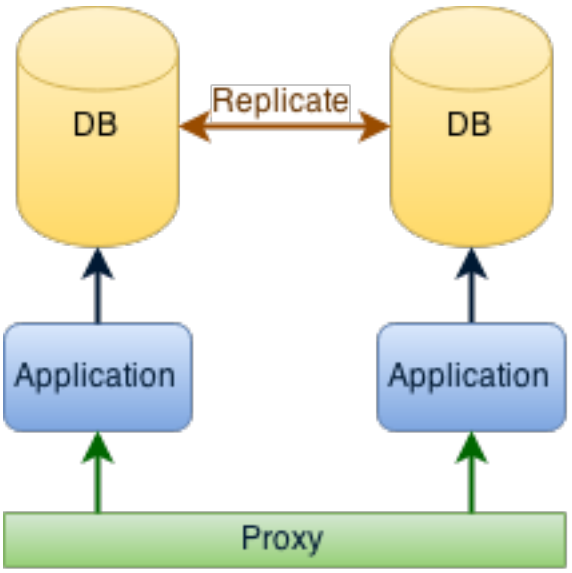

Reset Zoom
Flame Graph
LazyCompile:~fibonacci /home/god/test/example1.js:16
LazyCompile:~ /home/god/test/example1.js:6
LazyCompile:~emit events.js:68
Builtin:ArgumentsAdaptorTrampoline
LazyCompile:~parserOnIncoming _http_server.js:420
LazyCompile:~parserOnHeadersComplete _http_common.js:63
Builtin:JSEntryTrampoline
Stub:JSEntryStub
_ZN2v88internal9Execution4CallEPNS0_7IsolateENS0_6HandleINS0_6ObjectEEES6_iPS6_b
_ZN2v88Function4CallENS_6HandleINS_5ValueEEEPS3_
_ZN4node6Parser19on_headers_completeEP11http_parser
http_parser_execute
_ZN4node6Parser7ExecuteERKN2v820FunctionCallbackInfoINS1_5ValueEEE
_ZN2v88internal25FunctionCallbackArguments4CallEPFvRKNS_20FunctionCallbackInfoINS_5ValueEEEE
[unknown]
Stub:CEntryStub
LazyCompile:~socketOnData _http_server.js:340


Flame Graph
node
__libc_start_main
_start
uv_run
uv__io_poll
[unknown]
Stub:CEntryStub
Stub:JSEntryStub
Builtin:JSEntryTrampoline
Builtin:ArgumentsAdaptorTrampoline
LazyCompile:*emit events.js:68
LazyCompile:*onread net.js:516
LazyCompile:*readableAddChunk _stream_readable.js:139
LazyCompile:~socketOnData _http_server.js:340
http_parser_execute
LazyCompile:*parserOnHeadersComplete _htt...
LazyCompile:*OutgoingMessage...
LazyCompile:* /home/god/test/...
Builtin:ArgumentsAdaptorTram...
LazyCompile:*emit events.js:68
LazyCompile:~parserOnIncoming _htt...
LazyCompile:*c...
LazyCompile:...
LazyComp...
Stub:Cal...
Laz...
_ZN4node...
_ZN4n...
uv_tr...
[unk...
syst...
sys_...
vfs_...
do_s...
sock...
inet...
tcp...
__t...
ip...
ip6...
inet6...
tcp_t...
tcp_w...
__tcp...
tcp_s...
tcp_c...
sock_...
__fput
task_w...
do_exit
int_si...
__libc_...
uv__strea...
uv_close
_ZN4node10Han...
_ZN2v88Intern...
Stub::CEntryStub
LazyCompile:~S...
Builtin:Argume...
LazyCompile:*S...
Builtin:Funct...
LazyCompile:~g ...
Builtin:Function...
LazyCompile:*emit ...
LazyCompile:*after...
LazyCompile:~ _stre...
LazyCompile:_tickCallback...
_ZN2v88Internal9Execution4CallEPNS0_7IsolateENS0_6HandleINS0_6ObjectEEES6_IPS6_b
_ZN2v88Function4CallENS_6HandleINS_5ValueEEEiPS3_
_ZN4node9AsyncWrap12MakeCallbackEN2v86HandleINS1_8FunctionEEEiPNS2_INS1_5ValueEEE
_ZN4node19StreamWrapCallbacksD0EvEUlP11uv_stream_sP7K8uv_buf_t14uv_handle_type
_ZN2v88Internal9Execution4CallEPNS0_7IsolateERKNS0_6HandleINS0_6ObjectEEEiPPS6_
_ZN2v88Internal25FunctionCallbackArguments4CallEPFvRKNS...
_ZN4node6Parser7ExecuteERKN2v820FunctionCallbackInfoINS...
_ZN4node6Parser19on_headers_completeEP11http_parser
_ZN2v88Function4CallENS_6HandleINS_5ValueE...
_ZN2v88Internal9Execution4CallEPNS0_7Isol...
LazyCompile:*parserOnHeadersComplete _htt...
native_w...
intel_pm...
x86_pmu_...
perf_pmu...
perf_eve...
__perf_e...
finish_t...
__schedule
schedule
schedule...
ep_poll
sys_epoll...
system_cal...
syscall
S...
B...
Lazy...
LazyC...
Laz...
St...
Bu...
Stub::JSEntryStub
_ZN2v88Interna...
_ZN2v88Function...
_ZN4node9AsyncW...
_ZN4node7TCPWrap12OnC...
uv__server_io
SYS...
sys...
syscall
uv__ac...
nativ...
intel...
perf_...
__per...
finis...
__sch...
sched...
futex...
sys_f...
syste...
sem_w...
[unknown]
start_t...
v8::Swee...

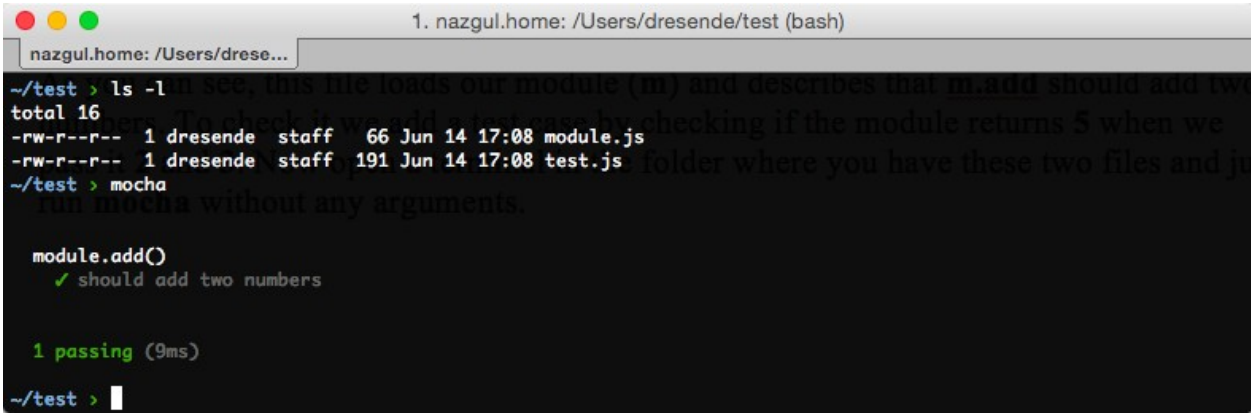
# Chapter 5: Data and Cache

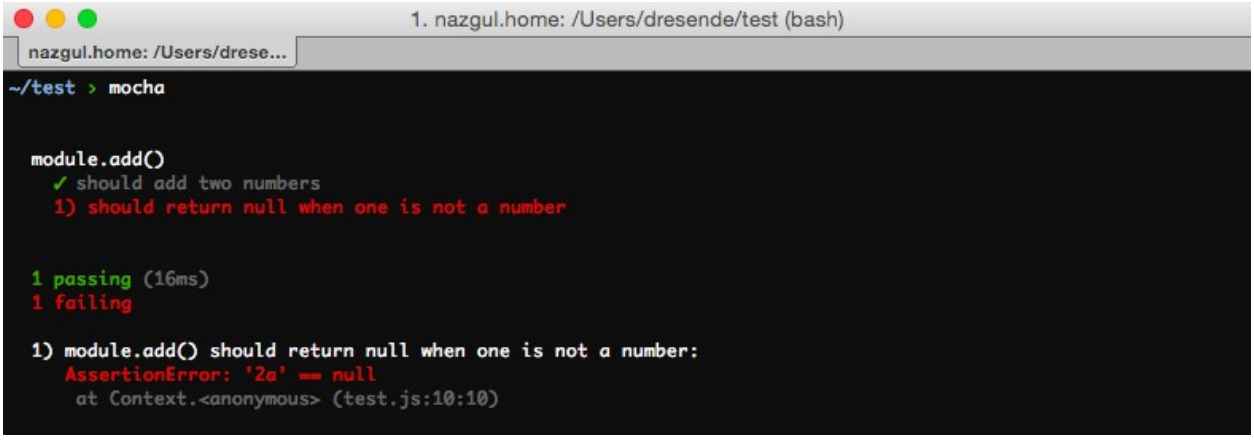


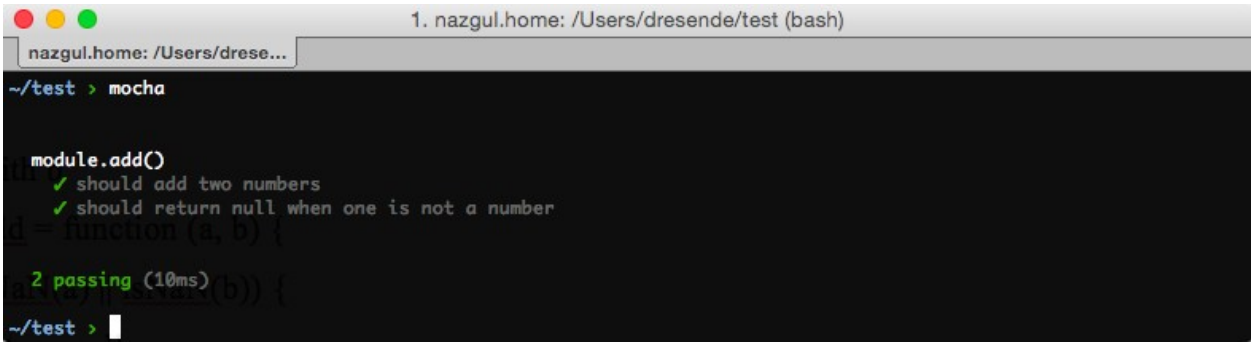




DB
Replicate
DB
Application
Application
Proxy

# Chapter 6: Test, Benchmark, and Analyze



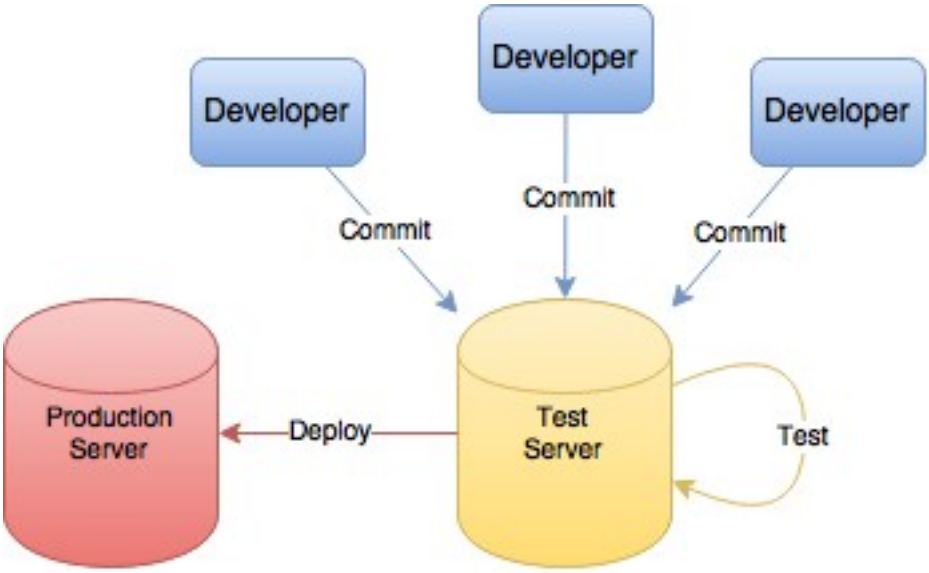



```
1. nazgul.home: /Users/dresende/test (bash)
nazgul.home: /Users/drese...
~/test › docker run -it --rm -v `pwd`:/opt/app env/test mocha /opt/app

  module.add()
    ✓ should add two numbers
    ✓ should return null when one is not a number


  2 passing (11ms)

~/test ›
```




```
1. nazgul.local: /Users/dresende/test (bash)
nazgul.local: /Users/drese...
~/test › mocha --reporter mocha-istanbul test.js

============================ Coverage summary ============================
Statements   : 100% ( 4/4 )
Branches     : 100% ( 4/4 )
Functions    : 100% ( 1/1 )
Lines        : 100% ( 4/4 )
==========================================================================
~/test ›
```

Code coverage report for **All files**

Statements: **100%** (4 / 4) Branches: **100%** (4 / 4) Functions: **100%** (1 / 1) Lines: **100%** (4 / 4) Ignored: none

| File | Statements | | Branches | | Functions | | Lines | |
|---|---|---|---|---|---|---|---|---|
| test/ | 100% | (4 / 4) | 100% | (4 / 4) | 100% | (1 / 1) | 100% | (4 / 4) |

Generated by istanbul at Wed Jun 17 2015 18:25:21 GMT+0100 (WEST)

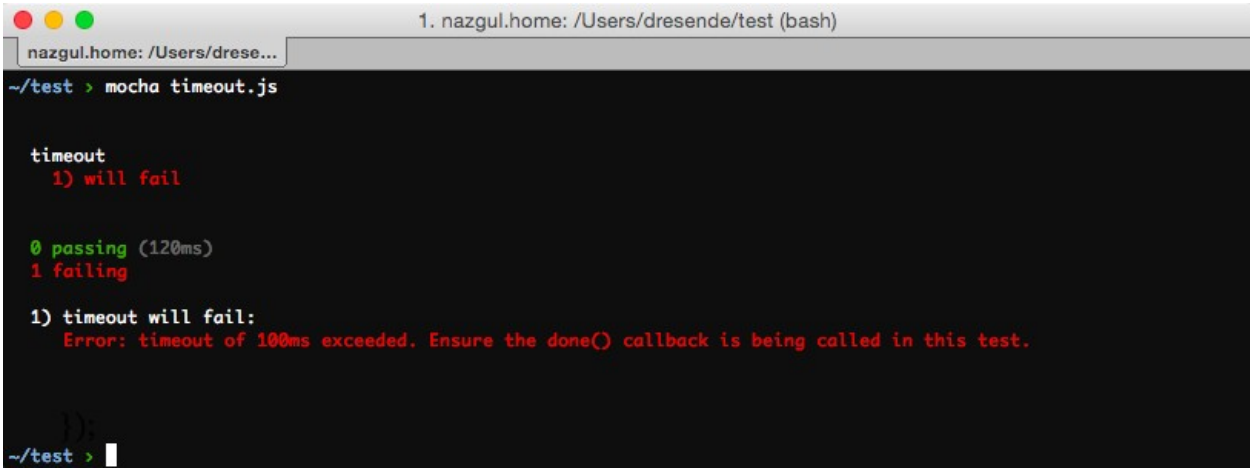
Code coverage report for **test/module.js**

Statements: **100%** (4 / 4) Branches: **100%** (4 / 4) Functions: **100%** (1 / 1) Lines: **100%** (4 / 4) Ignored: none

All files » test/ » module.js

```
1       // add a with b
2   1   exports.add = function (a, b) {
3   2     if (isNaN(a) || isNaN(b)) {
4   1       return null;
5         }
6   1     return a + b;
7       };
8
```

Generated by istanbul at Wed Jun 17 2015 18:25:21 GMT+0100 (WEST)



# Chapter 7: Bottlenecks

**Test runner**

Done. Ready to run again. Run again

| Testing in Chrome 43.0.2357.124 on OS X 10.10.3 | | |
|---|---|---|
| **Test** | | **Ops/sec** |
| **jQuery 1.3.2** | `tests($jq132);` | 21,947<br>±3.11%<br>77% slower |
| **jQuery 1.4.x** | `tests($jq14);` | 28,068<br>±2.51%<br>70% slower |
| **jQuery 1.6.x** | `tests($jq16);` | 61,020<br>±2.86%<br>35% slower |
| **jQuery 1.8.3** | `tests($jq183);` | 65,470<br>±1.89%<br>30% slower |
| **jQuery 1.9.1** | `tests($jq191);` | 70,215<br>±2.31%<br>25% slower |
| **jQuery 1.10.1** | `tests($jq1101);` | 91,629<br>±2.85%<br>fastest |
| **jQuery 2.1.0** | `tests($jq210);` | 94,035<br>±3.02%<br>fastest |
| **JQuery 1.7.2** | `tests($jq172);` | 61,212<br>±1.94%<br>34% slower |