

Отладка кода в браузере

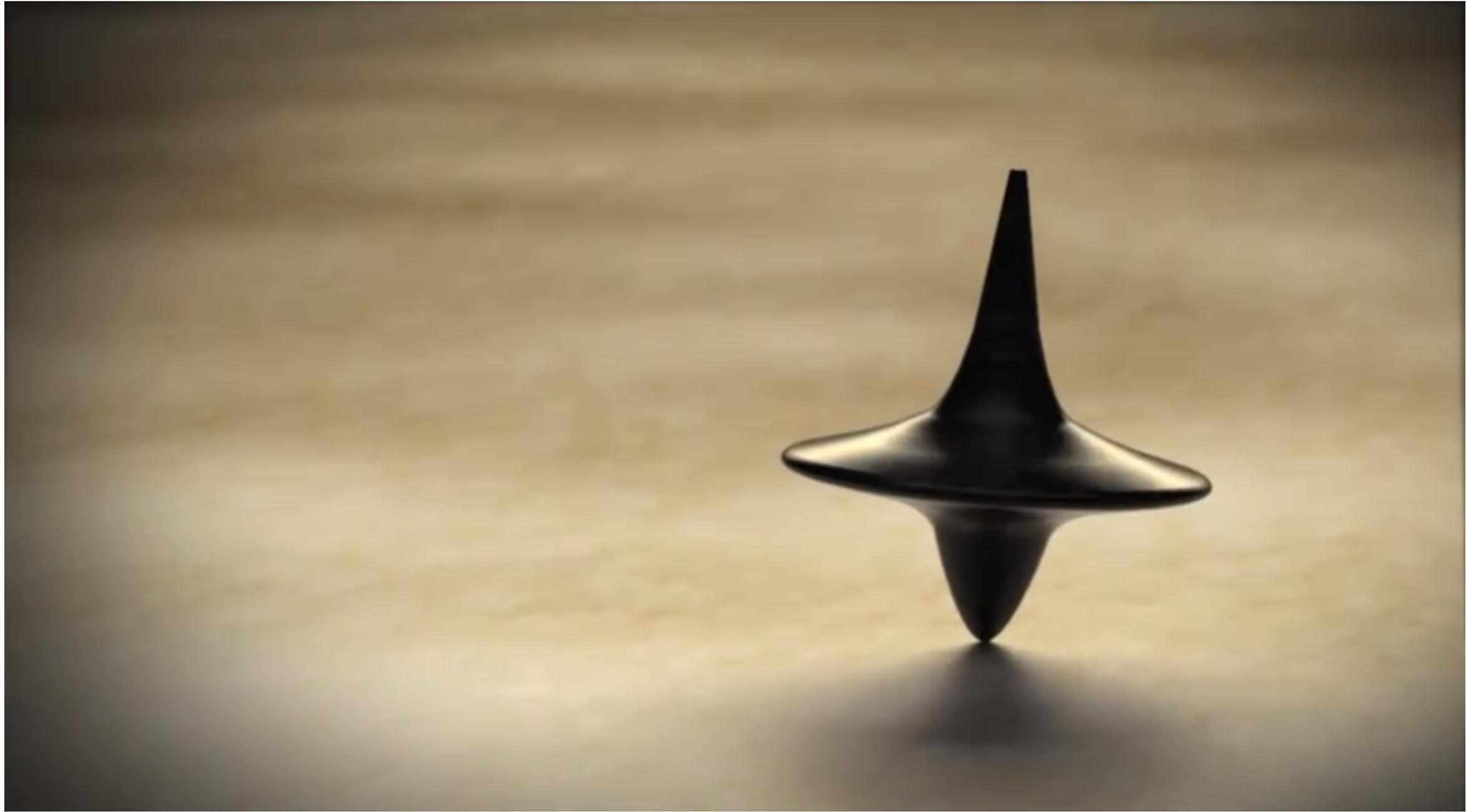
Шувалов Антон

Web Standards Days

Отладка кода в браузере

- Зачем нужен `console.log` и почему он не подходит для отладки?
- Как отладчик поможет вам?

Console



С чего всё началось?



`console.log()`, КАРЛ!

```
1  function dostuff() {
2      var item = findById(getId());
3      setActive(item);
4  }
5
6  function findById(id) {
7      return _.findWhere(store, { id: id });
8  }
9
10 var store = [
11     {id: 1, name: 'Anton'},
12     {id: 2, name: 'Sasha'}
13 ];
```

```
1 function dostuff() {
2   var item = findById(getId());
3   console.log(item); // undefined
4   setActive(item);
5 }
6
7 function findById(id) {
8   return _.findWhere(store, { id: id });
9 }
10
11 var store = [
12   {id: 1, name: 'Anton'},
13   {id: 2, name: 'Sasha'}
14 ];
```

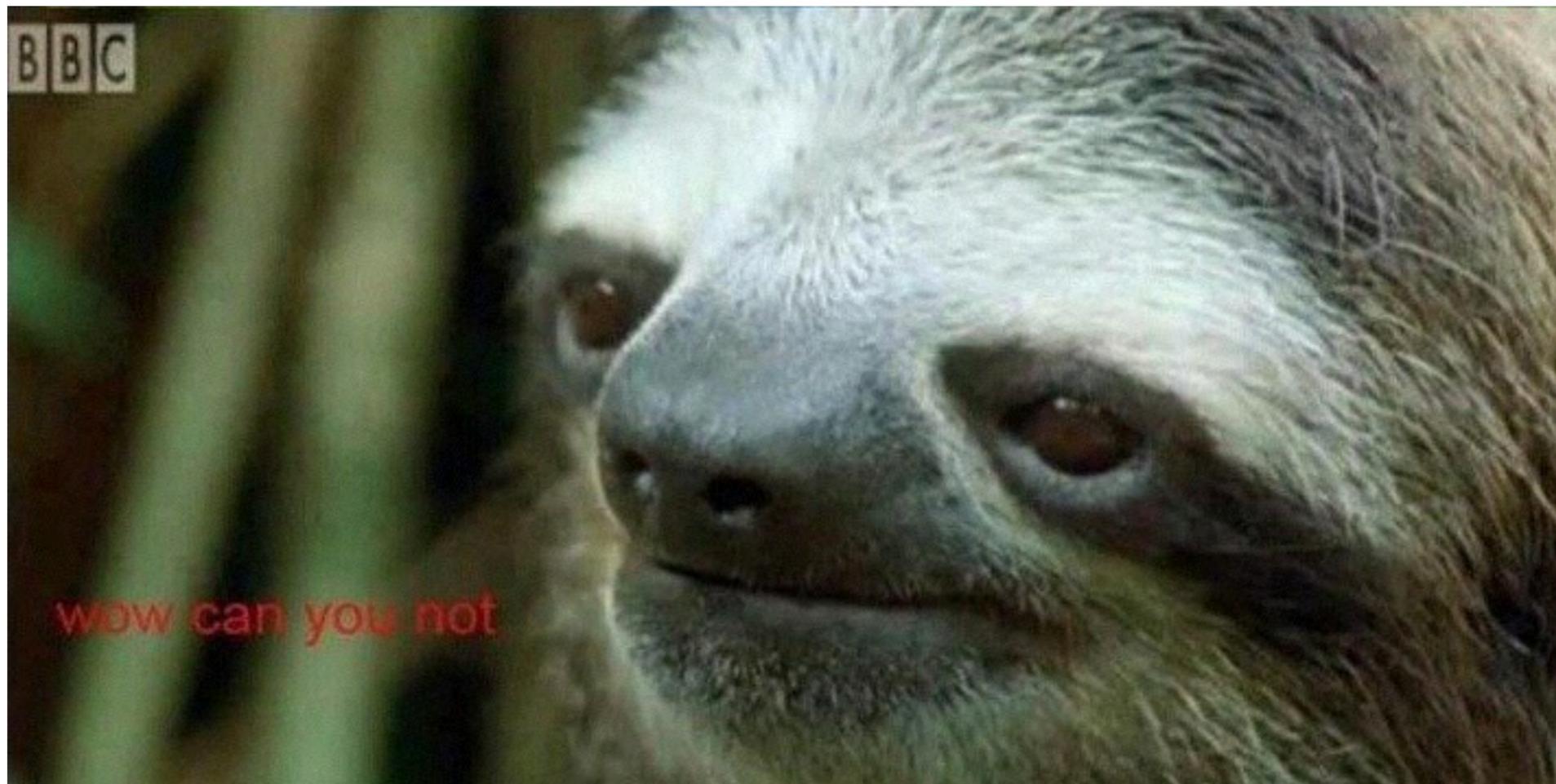
```
1 function dostuff() {
2     var item = findById(getId());
3     setActive(item);
4 }
5
6 function findById(id) {
7     var item = _.findWhere(store, { id: id });
8     console.log(item); // undefined
9     return item;
10 }
11
12 var store = [
13     {id: 1, name: 'Anton'},
14     {id: 2, name: 'Sasha'}
15 ];
```

```
1  function dostuff() {
2      var item = findById(getId());
3      setActive(item);
4  }
5
6  function findById(id) {
7      var item = _.findWhere(store, { id: id });
8      console.log(id); // '1'
9      return item;
10 }
11
12 var store = [
13     {id: 1, name: 'Anton'},
14     {id: 2, name: 'Sasha'}
15 ];
```

```
1 function dostuff() {
2     var item = findById(getId());
3     setActive(item);
4 }
5
6 function findById(id) {
7     id = id|0;
8     var item = _.findWhere(store, { id: id });
9     console.log(item); // {id: 1, name: 'Anton'}
10    return item;
11 }
12
13 var store = [
14     {id: 1, name: 'Anton'},
15     {id: 2, name: 'Sasha'}
16 ];
```

Что не так с console.log()

- Не точно
- Не подробно
- Медленно



console.log — не отладчик



HO...

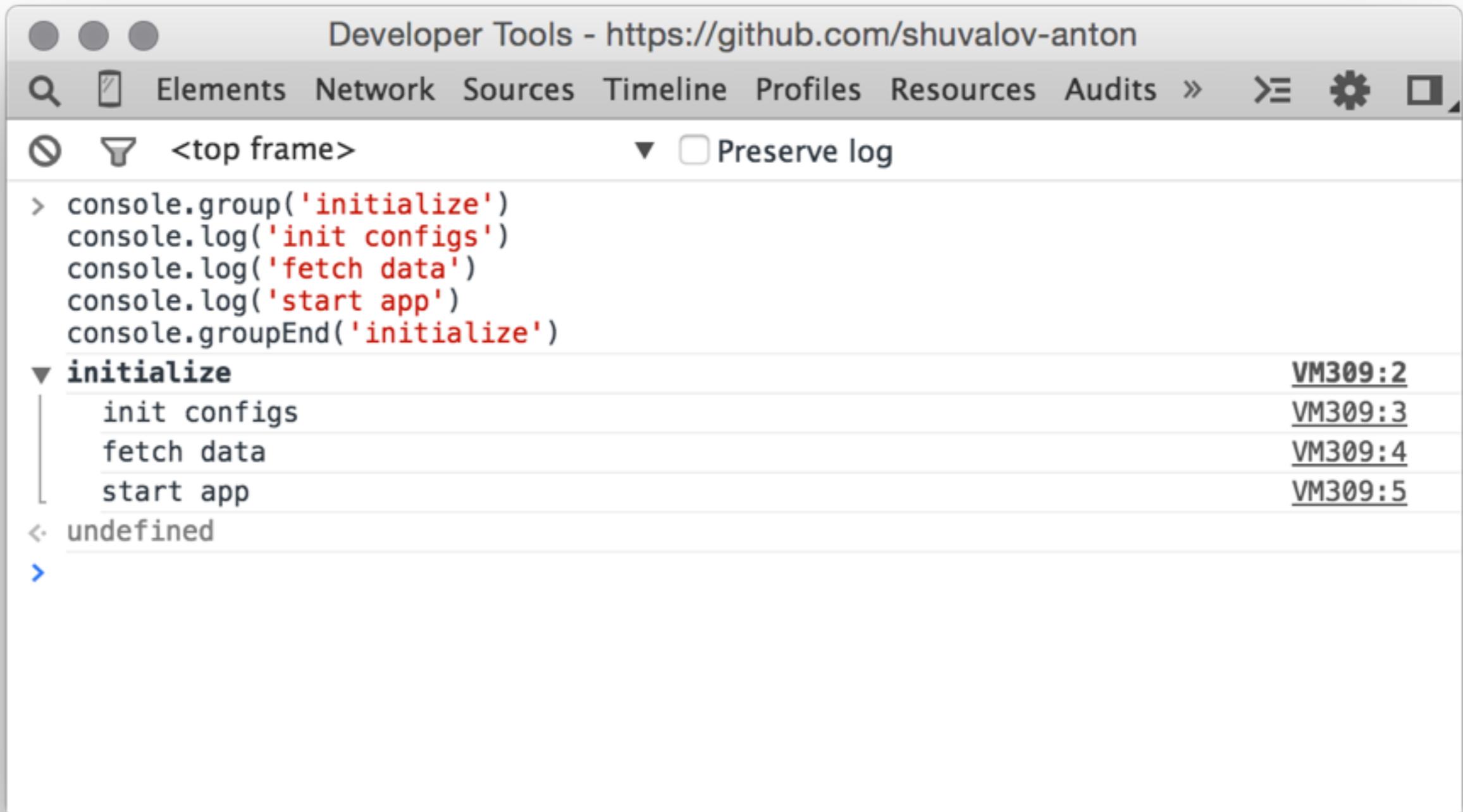


зачем нам console?

The screenshot shows the Chrome Developer Tools console for the URL `https://github.com/shuvalov-anton`. The console is set to the <top frame> and has the 'Preserve log' option checked. The log contains the following entries:

- A log message: `> console.log('value: %d', 17)` with the output `value: 17` and source `VM134:2`.
- An undefined return value: `< undefined`.
- A warning message: `> console.warn('method is deprecated')` with the output `method is deprecated` and source `VM175:2`.
- Another undefined return value: `< undefined`.
- An error message: `> console.error('You shall not pass!11')` with the output `▶ You shall not pass!11` and source `VM212:2`.
- A final undefined return value: `< undefined`.
- A blue prompt character: `>`.

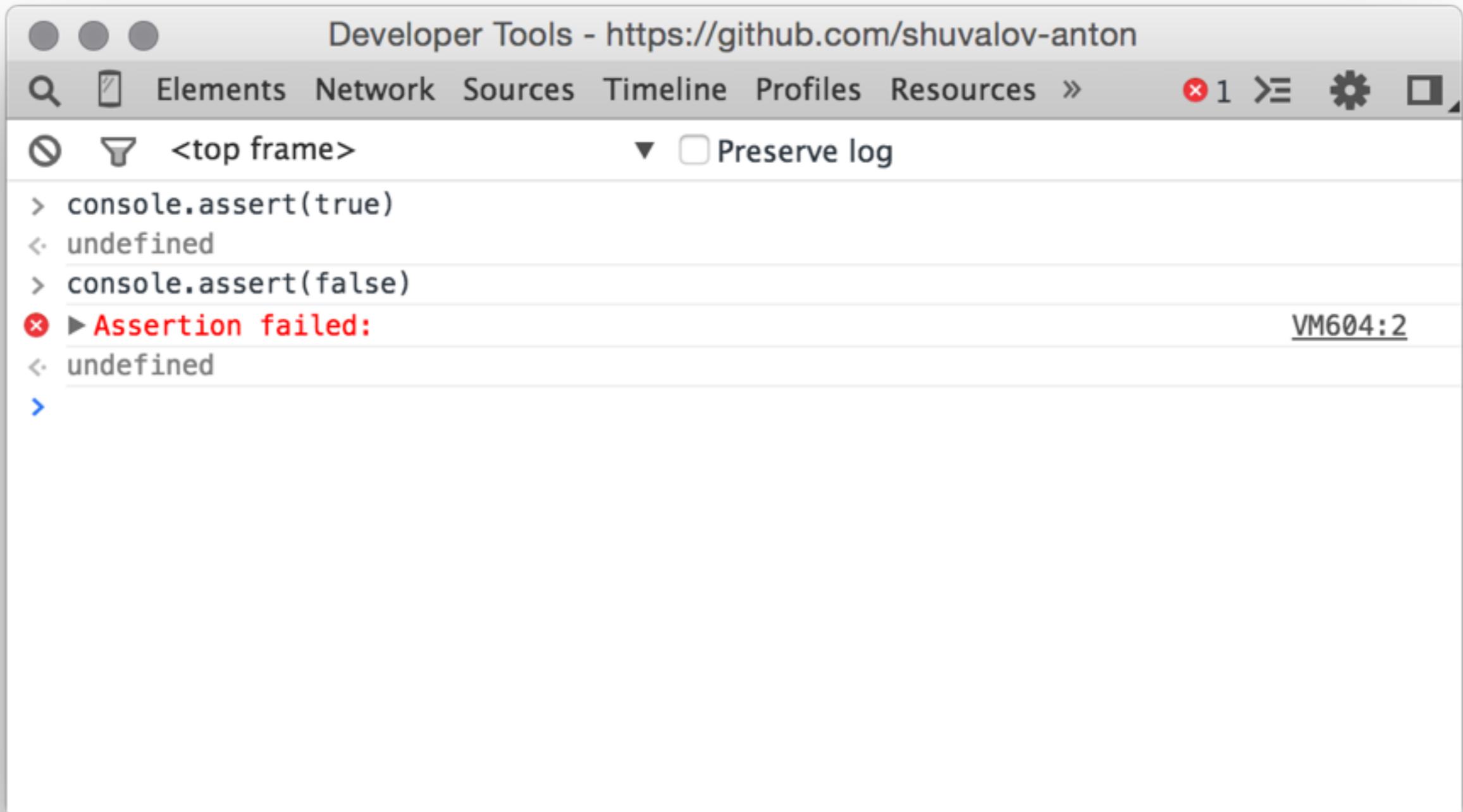
console



console.group

```
Developer Tools - https://github.com/shuvalov-anton
Elements Network Sources Timeline Profiles Resources Audits >> >≡ ⚙️ □
<top frame> ▼  Preserve log
> console.log($('body')[0])
▶ <body class="logged_in env-production macintosh page-profile mine">...</body> VM459:2
< undefined
> console.dir($('body')[0])
▼ body.logged_in.env-production.macintosh.page-profile.mine ⓘ VM461:2
  aLink: ""
  accessKey: ""
  ▶ attributes: NamedNodeMap
  background: ""
  baseURI: (...)
  bgColor: ""
  childElementCount: 7
  childNodes: (...)
  ▶ children: HTMLCollection[7]
  ▶ classList: DOMTokenList[5]
  className: "logged_in env-production macintosh page-profile mine"
```

console.dir



console.assert



The screenshot shows the Chrome Developer Tools console. The browser tab is titled "Developer Tools - https://github.com/shuvalov-anton". The console toolbar includes a search icon, a mobile device icon, and tabs for "Elements", "Network", "Sources", "Timeline", "Profiles", "Resources", and "Audits". The console content area shows the following code and output:

```
> console.time('initialize');  
  setTimeout(function () { console.timeEnd('initialize'); }, 1000);  
< 511  
initialize: 1001.260ms VM729:3  
>
```

console.time



The screenshot shows the Chrome Developer Tools console. The browser tab is titled "Developer Tools - https://github.com/shuvalov-anton". The console toolbar includes a search icon, a filter icon, and a "Preserve log" checkbox. The console content shows the following:

```
> console.profile('find element');  
$('body')  
console.profileEnd()  
< undefined  
Profile 'find element' started. VM817:2  
Profile 'find element' finished. VM817:4  
>
```

console.profile

Developer Tools - https://github.com/shuvalov-anton

Elements Network Sources Timeline Profiles Resources Audits »

Heavy (Bottom Up) ▾

Profiles	Self		Total		Function
	ms	%	ms	%	
CPU PROFILES	0.0	0.00	0.0	0.00	Inject... VM84:679
find element Save	0.0	0.00	0.0	0.00	▶ Injected... VM84:813
	0.0	0.00	0.0	0.00	▶ ⚠ Inject... VM84:850
	0.0	0.00	0.0	0.00	▶ evaluateWithExcep...
	0.0	0.00	0.0	0.00	▶ ⚠ (anony... VM817:1

console.profile

Возможности console

- Сообщения, ошибки предупреждения
 - `log()`, `info()`, `warn()`, `error()`, `assert()`, `group()`, `dir()`
- Профилирование
 - `time()`, `profile()`

Отладчик

ОСНОВЫ ОТЛАДКИ

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

Line 26, Column 1

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

Uncaught Error: no item [index.js:15](#)

▶ Watch Expressions + ↻
▶ Call Stack Async
▶ Scope Variables
▶ Breakpoints
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

Line 26, Column 1

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

Uncaught Error: no item index.js:15

Watch Expressions + ↻

Call Stack Async

Scope Variables

Breakpoints

index.js:2
var item = findById(getId());

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

▶ Watch Expressions + ↻
▶ Call Stack Async
▶ Scope Variables
▼ Breakpoints
 index.js:2
var item = findById(getId());
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

{ } Line 2, Column 3

Console Search Emulation Rendering

<top frame> ▼ Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

>

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

undefined

- ▶ Watch Expressions + ↻
- ▶ Call Stack Async
- ▶ Scope Variables
- ▼ Breakpoints
 - index.js:2
var item = findById(getId());
 - ▶ DOM Breakpoints
 - ▶ XHR Breakpoints +
 - ▶ Event Listener Breakpoints

{ } 18 characters selected

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

>

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

Line 2, Column 3

Watch Expressions + ↻

Call Stack Async

Scope Variables

Breakpoints

index.js:2
var item = findById(getId());

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

Console Search Emulation Rendering

Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

>

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('setActive', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Item 1'},
21   {id: 2, name: 'Item 2'}
22 ];
23
24
25 dostuff();
26
```

▶ Watch Expressions + ↻
▶ Call Stack Async
▶ Scope Variables
▼ Breakpoints
 index.js:2
var item = findById(getId());
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

Array[2]
▶ 0: Object
▶ 1: Object
length: 2
▶ __proto__: Array[0]

{ } Line 11, Column 23

Console Search Emula

<top frame>

Filter Regex All Errors warnings info Logs Debug Hide network messages

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %s', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   {id: 2, name: 'Sasha'}
22 ];
23
24
25 dostuff();
26
```

Line 10, Column 24

Watch Expressions + ↻

Call Stack Async

Scope Variables

Breakpoints

index.js:2
var item = findById(getId());

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

>

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
```

{ } 5 characters selected

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

> id=id|0
< 1
>

▶ Watch Expressions + ↻
▶ Call Stack Async
▶ Scope Variables
▼ Breakpoints
☑ index.js:2
var item = findById(getId());
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return $.findWhere(store, { id: id });
12 }
13
14 funct
15   f
16   on
17 }
18
19 var s
20   id
31 ch
```

▶ Watch Expressions + ↻
▶ Call Stack Async
▶ Scope Variables
▼ Breakpoints
 index.js:2
var item = findById(getId());
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

Object

```
id: 1
name: "Anton"
▶ __proto__: Object
```

Console

Filter

```
> id=id|e
< 1
>
```

Debug Hide network messages

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return $.findWhere(store, { id: id });
12 }
13
14 funct
15   if
16   con
17 }
18
19 var s
20 {id
31 ch
```

Object

- id: 1
- name: "Anton"
- __proto__: Object

Watch Expressions

Call Stack

Scope Variables

Breakpoints

- index.js:2
var item = findById(getId());

DOM Breakpoints

XHR Breakpoints

Event Listener Breakpoints

Console

Filter

> id=id|e
< 1
>

Debug Hide network messages

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

index.js x

```
1 function dostuff() {
2   var item = findById(getId());
3   setActive(item);
4 }
5
6 function getId() {
7   return $('#input').val();
8 }
9
10 function findById(id) {
11   return _.findWhere(store, { id: id });
12 }
13
14 function setActive(item) {
15   if (!item) throw new Error('no item');
16   console.log('set %o', item);
17 }
18
19 var store = [
20   {id: 1, name: 'Anton'},
21   ...
22 ]
```

31 characters selected

Watch Expressions + ↻

Call Stack Async

Scope Variables

Breakpoints

index.js:2
var item = findById(getId());

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** Errors Warnings Info Logs Debug Hide network messages

```
> id=id|0
< 1
set ▼ Object 1 index.js:16
  id: 1
  name: "Anton"
  ▶ __proto__: Object
> |
```

НИЧОСИ



Нет постоянных
перезагрузок страницы!



Управление ходом исполнения программы!



Доступ к стеку вызовов!



Отлично работающий
REPL!



Работает с SourceMaps!



НИЧОСИ



Что ещё интересного
в DevTools?

Watch Expressions

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8 |
```

Watch Expressions + ↻

No Watch Expressions

Call Stack Async

Scope Variables

Breakpoints

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

{ } Line 8, Column 1

The image shows a screenshot of the Chrome Developer Tools interface. The top bar indicates the page is 'http://localhost:8001/'. The 'Sources' panel is active, showing a file named 'watch-expressions.js'. The code is as follows:
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2 return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6 return a + b;
7 }
8 |
Line 2 is highlighted in blue. To the right, the 'Watch Expressions' panel is open, showing 'No Watch Expressions'. Below it are other panels: 'Call Stack', 'Scope Variables', 'Breakpoints', 'DOM Breakpoints', 'XHR Breakpoints', and 'Event Listener Breakpoints'. A red rectangle highlights the 'Watch Expressions' panel and the toolbar icons above it. The status bar at the bottom shows '{ } Line 8, Column 1'.

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

sum(m,v)

Call Stack Async

Scope Variables

Breakpoints

- watch-expressions.js:2
return sum(m, v);

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

{ } Line 2, Column 10

The image shows the Chrome DevTools interface. The 'Sources' panel is active, displaying the file 'watch-expressions.js'. The code is as follows:
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2 return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6 return a + b;
7 }
8
The second line of code is highlighted in blue. On the right side, the 'Watch Expressions' panel is open, showing a single watch expression 'sum(m,v)'. Below this, the 'Breakpoints' section is expanded, and a breakpoint is set on line 2 of 'watch-expressions.js' at the 'return sum(m, v);' statement. A red box highlights the Watch Expressions and Breakpoints sections.

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

▼ Watch Expressions + ↻

sum(m,v): 1

▶ Call Stack Async

▶ Scope Variables

▶ Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 8, Column 1

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

▼ Watch Expressions + ↻

sum(m,v): 3

▶ Call Stack Async

▶ Scope Variables

▶ Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 2, Column 10

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

▼ Watch Expressions + ↻

sum(m, v): 6

▶ Call Stack Async

▶ Scope Variables

▶ Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 2, Column 10

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

Watch Expressions + ↻

sum(m,v): 10

Call Stack Async

Scope Variables

Breakpoints

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

{ } Line 2, Column 10

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

watch-expressions.js x

```
1 var x = [1, 2, 3, 4, 5].reduce(function(m, v) {
2   return sum(m, v);
3 }, 0);
4
5 function sum(a, b) {
6   return a + b;
7 }
8
```

Watch Expressions + ↻

sum(m, v): 15

Call Stack Async

Scope Variables

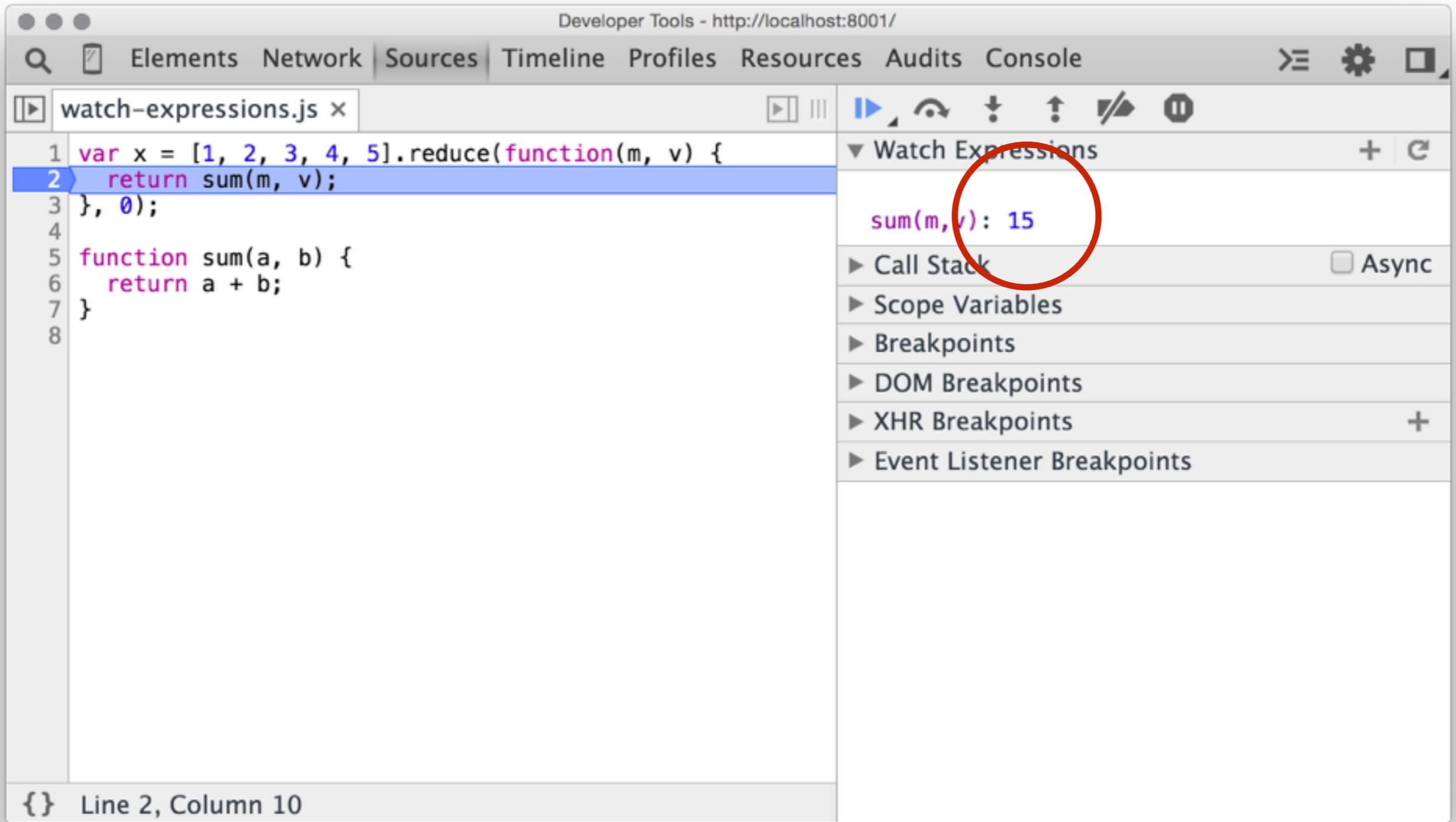
Breakpoints

DOM Breakpoints

XHR Breakpoints +

Event Listener Breakpoints

{ } Line 2, Column 10



Работа со стеком

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... stack.js x

(no domain) code.jquery.c localhost:800 (index) stack.js underscorejs.

```
1 var stack = [  
2   function() { return 1; },  
3   function() { return 2; },  
4   function() { return 3; },  
5   function() { return 4; },  
6   function() { debugger; },  
7 ];  
8  
9 function dispatch(ctx) {  
10  var i = 0;  
11  function next() {  
12    var fn = stack[i++];  
13    if (fn) fn() && next()  
14  }  
15  next();  
16 }  
17  
18 dispatch();  
19
```

fn(): <not available>

Call Stack Async

stack	stack.js:6
next	stack.js:13
dispatch	stack.js:15
(anonymous function)	stack.js:18

Scope Variables

Breakpoints

No Breakpoints

DOM Breakpoints

XHR Breakpoints

{ } Line 6, Column 16

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... stack.js x

(no domain) code.jquery.c localhost:800 (index) stack.js underscorejs.

```

1 var stack = [
2   function() { return 1; },
3   function() { return 2; },
4   function() { return 3; },
5   function() { return 4; },
6   function() { debugger; },
7 ];
8
9 function dispatch(ctx) {
10  var i = 0;
11  function next() {
12    var fn = stack[i++];
13    if (fn) fn() && next()
14  }
15  next();
16 }
17
18 dispatch();
19

```

Watch Expressions

fn(): 4

Call Stack Async

stack	stack.js:6
next	stack.js:13
dispatch	stack.js:15
(anonymous function)	stack.js:18

Scope Variables

Breakpoints

No Breakpoints

DOM Breakpoints

XHR Breakpoints

{ } Line 13, Column 21

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... stack.js x

```
1 var stack = [  
2   function() { return 1; },  
3   function() { return 2; },  
4   function() { return 3; },  
5   function() { return 4; },  
6   function() { debugger; },  
7 ];  
8  
9 function dispatch(ctx) {  
10  var i = 0;  
11  function next() {  
12    var fn = stack[i++];  
13    if (fn) fn() && next()  
14  }  
15  next();  
16 }  
17  
18 dispatch();  
19
```

Watch Expressions
fn(): 2

Call Stack Async
stack stack.js:6
next stack.js:13
next stack.js:13
next stack.js:13
next stack.js:13
next stack.js:13
next stack.js:13
dispatch stack.js:15
(anonymous function) stack.js:18

Scope Variables
Breakpoints
No Breakpoints
DOM Breakpoints
XHR Breakpoints

{ } Line 13, Column 27

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... stack.js x

```
6 function() { debugger; },
7 ];
8
9 function dispatch(ctx) {
10   var i = 0;
11   function next() {
12     var fn = stack[i++];
13     if (fn) fn() && next()
14   }
15   next();

```

Line 13, Column 21

Watch Expressions

fn(): 3

Call Stack Async

stack	stack.js:6
next	stack.js:13
next	stack.js:13
next	stack.js:13

Console Search Emulation Rendering

<top frame> Preserve log

Filter Regex **All** | Errors Warnings Info Logs Debug Hide network messages

```
> fn()
< 3
> i
< 5
> |
```

Отладка асинхронного кода

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... async.js x jquery-2.1.4.min.js

▶ (no domain)
▶ code.jquery.c
▼ localhost:800
 (index)
 async.js
▶ underscorejs.

```
1 $.get('/404').error(onError);  
2  
3 function onError() {  
4   debugger;  
5 }  
6
```

▶ Watch Expressions
▼ Call Stack Async
onError async.js:4
n.Callbacks.j jquery-2.1.4.min.js:2
n.Callbacks.k.fireWith jquery-2.1.4.min.js:2
x jquery-2.1.4.min.js:4
jquery-2.1.4.min.js:4
n.ajaxTransport.k.cors.a.crossDomain.send.b

▶ Scope Variables
▶ Breakpoints
▶ DOM Breakpoints
▶ XHR Breakpoints +
▶ Event Listener Breakpoints

{ } Line 6, Column 1

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... async.js x jquery-2.1.4.min.js

```
1 $.get('/404').error(onError);
2
3 function onError() {
4   debugger;
5 }
6
```

▶ Watch Expressions + ↻

▼ Call Stack Async

onError async.js:4

n.Callbacks.j jquery-2.1.4.min.js:2

n.Callbacks.k.fireWith jquery-2.1.4.min.js:2

x jquery-2.1.4.min.js:4

jquery-2.1.4.min.js:4

n.ajaxTransport.k.cors.a.crossDomain.send.b

▶ Scope Variables

▶ Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 6, Column 1

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... async.js x jquery-2.1.4.min.js

▶ (no domain) 1 \$.get('/404').error(onError);
▶ code.jquery.c 2
▼ localhost:800 3 function onError() {
(index) 4 debugger;
async.js 5 }
underscorejs. 6

Call Stack Async

onError	async.js:4
n.Callbacks.j	jquery-2.1.4.min.js:2
n.Callbacks.k.fireWith	jquery-2.1.4.min.js:2
x	jquery-2.1.4.min.js:4
	jquery-2.1.4.min.js:4
n.ajaxTransport.k.cors.a.crossDomain.send.b	
<i>XMLHttpRequest.send (async)</i>	
	jquery-2.1.4.min.js:4
n.ajaxTransport.k.cors.a.crossDomain.send	
n.extend.ajax	jquery-2.1.4.min.js:4
n.each.n.(anonymous function)	jquery-2.1.4.min.js:4
(anonymous function)	async.js:1

Scope Variables
Breakpoints
DOM Breakpoints

{ } Line 4, Column 3

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

S... C... S... async.js x jquery-2.1.4.min.js

(no domain) code.jquery.c localhost:800 (index) async.js underscorejs.

```
1 $.get('/404').error(onError);
2
3 function onError() {
4     debugger;
5 }
6
```

Watch Expressions

Call Stack Async

- onError async.js:4
- n.Callbacks.j jquery-2.1.4.min.js:2
- n.Callbacks.k.fireWith jquery-2.1.4.min.js:2
- x jquery-2.1.4.min.js:4
- jquery-2.1.4.min.js:4
- n.ajaxTransport.k.cors.a.crossDomain.send.b
- XMLHttpRequest.send (async)*
- jquery-2.1.4.min.js:4
- n.ajaxTransport.k.cors.a.crossDomain.send
- n.extend.ajax jquery-2.1.4.min.js:4
- n.each.n.(anonymous function) jquery-2.1.4.min.js:4
- (anonymous function) async.js:1**

Scope Variables

Breakpoints

DOM Breakpoints

{ } Line 1, Column 3

Developer Tools - http://localhost:8001/

Elements Network Sources Timeline Profiles Resources Audits Console

async.js x jquery 2.1.4.min.js

```
1 $.get('/404').error(onError);
2
3 function onError() {
4     debugger;
5 }
6
```

Watch Expressions +

Call Stack Async

- onError async.js:4
- n.Callbacks.j jquery-2.1.4.min.js:2
- n.Callbacks.k.fireWith jquery-2.1.4.min.js:2
- x jquery-2.1.4.min.js:4
- jquery-2.1.4.min.js:4
- n.ajaxTransport.k.cors.a.crossDomain.send.b
- XMLHttpRequest.send (async)*
- jquery-2.1.4.min.js:4
- n.ajaxTransport.k.cors.a.crossDomain.send
- n.extend.ajax jquery-2.1.4.min.js:4
- n.each.n.(anonymous function) jquery-2.1.4.min.js:4
- (anonymous function) async.js:1

Scope Variables

Breakpoints

DOM Breakpoints

{ } Line 1, Column 3

Карты кода

Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

app.js x

```
1 const plugins = [];  
2  
3 class App {  
4  
5   static use(...args) {  
6     return plugins.push(...args);  
7   }  
8  
9   init() {  
10    plugins.map(plugin => plugin.init(this));  
11  }  
12  
13 }  
14
```

- ▶ Watch + ↻
- ▶ Call Stack Async
- ▶ Scope
- ▶ Breakpoints
- ▶ DOM Breakpoints
- ▶ XHR Breakpoints +
- ▶ Event Listener Breakpoints

{ } Line 5, Column 15

Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

app.bundle.js x

```
1 (function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require <
2 "use strict";
3
4 var _createClass = (function () { function defineProperties(target, props) {
5
6 function _classCallCheck(instance, Constructor) { if (!(instance instanceof
7
8 var plugins = [];
9
10 var App = (function () {
11   function App() {
12     _classCallCheck(this, App);
13   }
14
15   _createClass(App, [{
16     key: "init",
17     value: function init() {
18       var _this = this;
19
20       plugins.map(function (plugin) {
21         return plugin.init(_this);
22       });
23     }
24   }], [{
25     key: "use",
26     value: function use() {
27       for (var _len = arguments.length, args = Array(_len), _key = 0; _ke
28         args[_key] = arguments[_key];
29     }
30
31     return plugins.push.apply(plugins, args);
32
33 }
34 }
```

Watch + ↻
Call Stack Async
Scope
Breakpoints
DOM Breakpoints
XHR Breakpoints +
Event Listener Breakpoints

{ } Line 18, Column 24

Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

app.js x

```
1 const plugins = [];  
2  
3 class App {  
4  
5   static use(...args) {  
6     return plugins.push(...args);  
7   }  
8  
9   init() {  
10    plugins.map(plugin => plugin.init(this));  
11  }  
12  
13 }  
14
```

- ▶ Watch + ↻
- ▶ Call Stack Async
- ▶ Scope
- ▶ Breakpoints
- ▶ DOM Breakpoints
- ▶ XHR Breakpoints +
- ▶ Event Listener Breakpoints

{ } Line 5, Column 15

Карты Кода

- browserify --debug
- gulp-sourcemaps
- webpack { devtool: 'inline-source-map' }
- ...
- profit!



`_this`

__this

`_store2['default']`

Не работают в REPL



Blackboxing

Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources >> >_ ⚙️ 🗨️

debug.js app.bundle.js app.js x

```
1 var debug = require('debug')('app');
2
3 debug('do things!');
4
```

▶ Watch + ↺

▶ Call Stack Async

▶ Scope

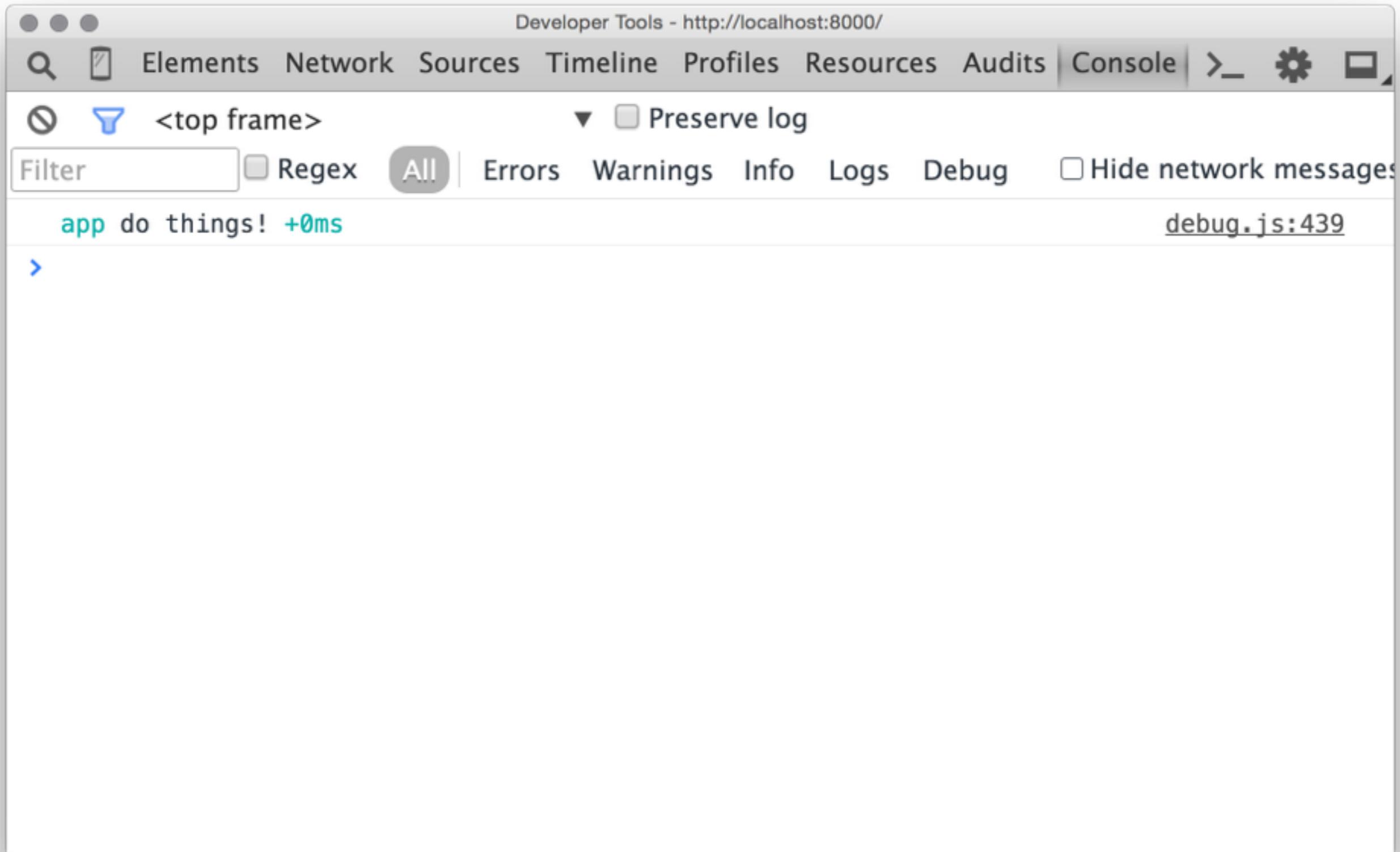
▶ Breakpoints

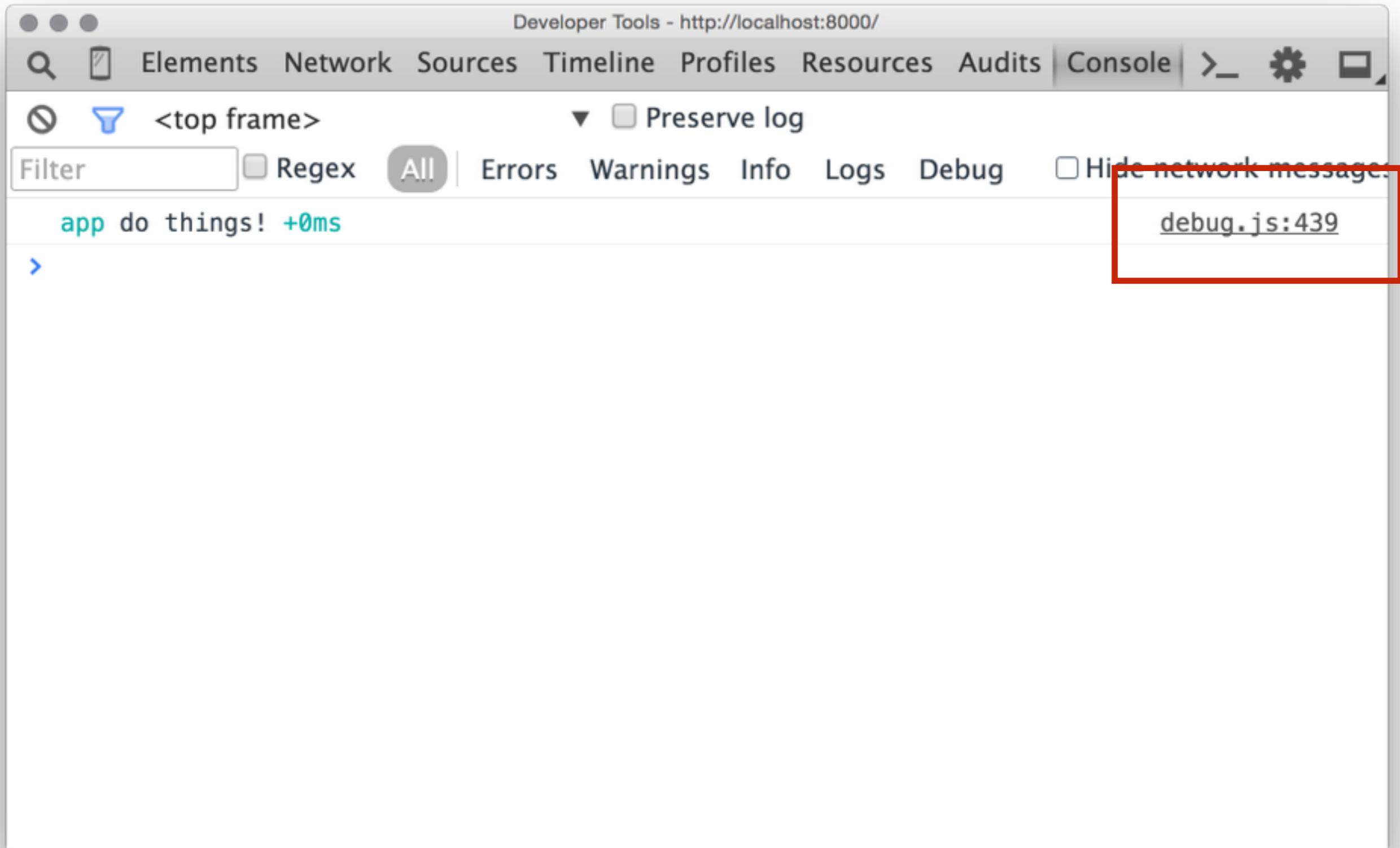
▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 4, Column 1





Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

So... Co... Sn... debug.js x app.bundle.js app.js

(no domain) 428 * INVOKES console.log() when available.
429 * No-op when `console.log` is not a "func
localhost:8000 430 *
431 * @api public
node_module 432 */
(index) 433
app.bundle.j 434 function log() {
435 // this hackery is required for IE8/9, w
app.js 436 // the `console.log` function doesn't ha
437 return 'object' === typeof console
d console.log
Function.prototype.apply.call(conso

Watch +
Call Stack Async
Scope
Breakpoints
DOM Breakpoints
XHR Breakpoints +
Event Listener Breakpoints

Add Folder to Workspace
Local Modifications
Blackbox Script
Reveal in Network Panel
Open Link in New Tab
Copy Link Address
Save
Save As...

ve `namespaces`.
aram {String} namespaces
oi private
449 function save(namespaces) {
450 try {
451 if (null == namespaces) {
{ } Line 439, Column 33

Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

So... Co... Sr... debug.js x app.bundle.js app.js

(no domain) localhost:8000

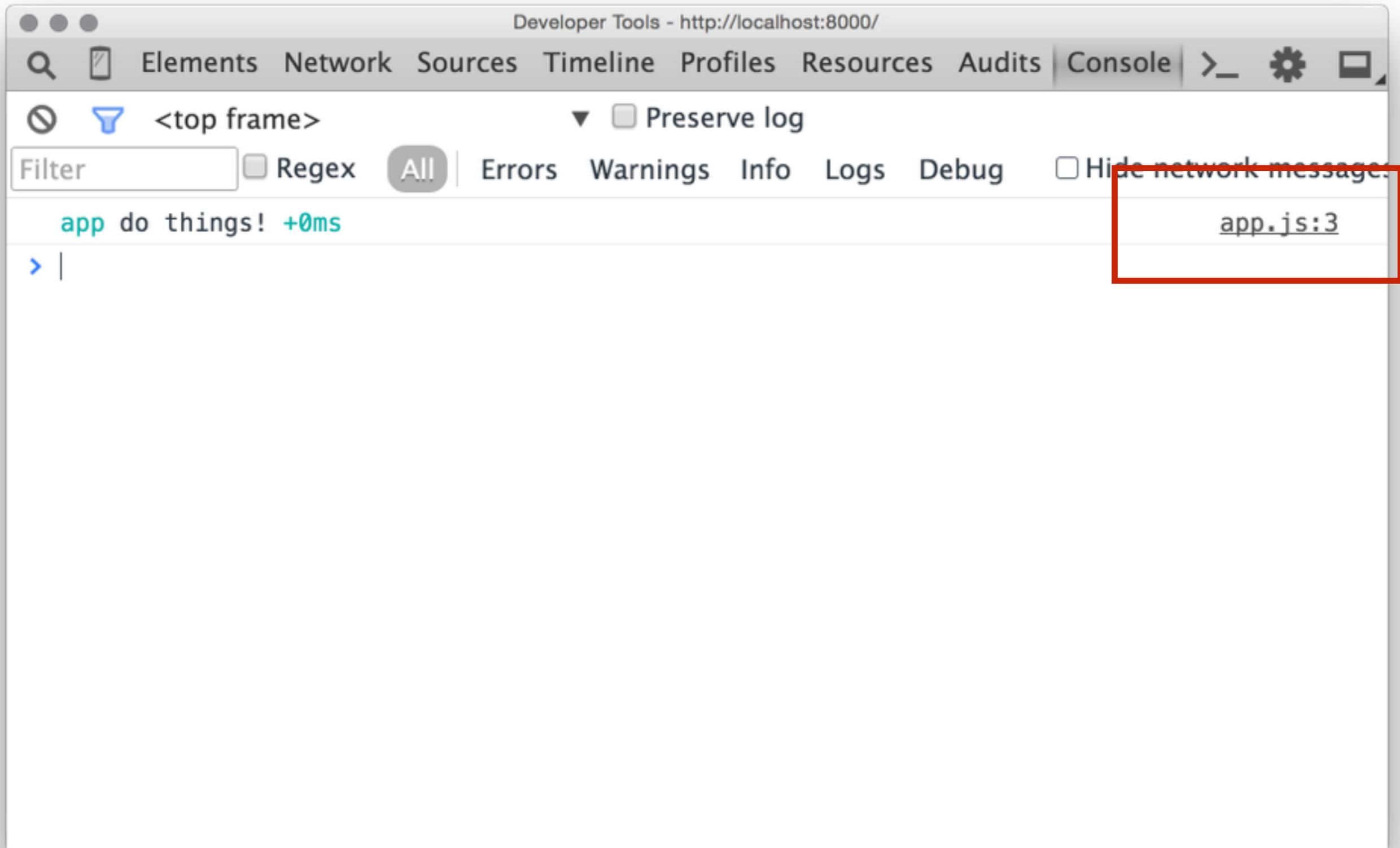
node_modules (index) app.bundle.js app.js debug.js

This script is blackboxed in debugger [more](#) x

```
428 * INVOKES console.log() when available.
429 * No-op when `console.log` is not a "func
430 *
431 * @api public
432 */
433
434 function log() {
435   // this hackery is required for IE8/9, w
436   // the `console.log` function doesn't ha
437   return 'object' === typeof console
438     && console.log
439     && Function.prototype.apply.call(conso
440 }
441
442 /**
443  * Save `namespaces`.
444  *
445  * @param {String} namespaces
446  * @api private
447  */
448
449 function save(namespaces) {
450 }
```

Line 439, Column 33

Watch Call Stack Scope Breakpoints DOM Breakpoints XHR Breakpoints Event Listener Breakpoints



Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

So... Co... Sn... debug.js app.bundle.js app.js x

(no domain)
localhost:8000
node_modules
(index)
app.bundle.js
app.js
debug.js

```
1 var debug = require('debug')('app');  
2  
3 debug('do things!');  
4
```

Watch +
Call Stack Async
Scope
Breakpoints
DOM Breakpoints
XHR Breakpoints +
Event Listener Breakpoints

{ } Line 3, Column 1

Отладчик не заходит
внутри черного ящика



Идеально подходит
к оберткам над логгерами



Работает только
с файлами



Console — для логов

DevTools — для отладки



Не забивайте гвозди
ИЗОЛЕНТОЙ

Шувалов Антон

github.com/shuvalov-anton

twitter.com/shuvalov_anton