

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 ];
```

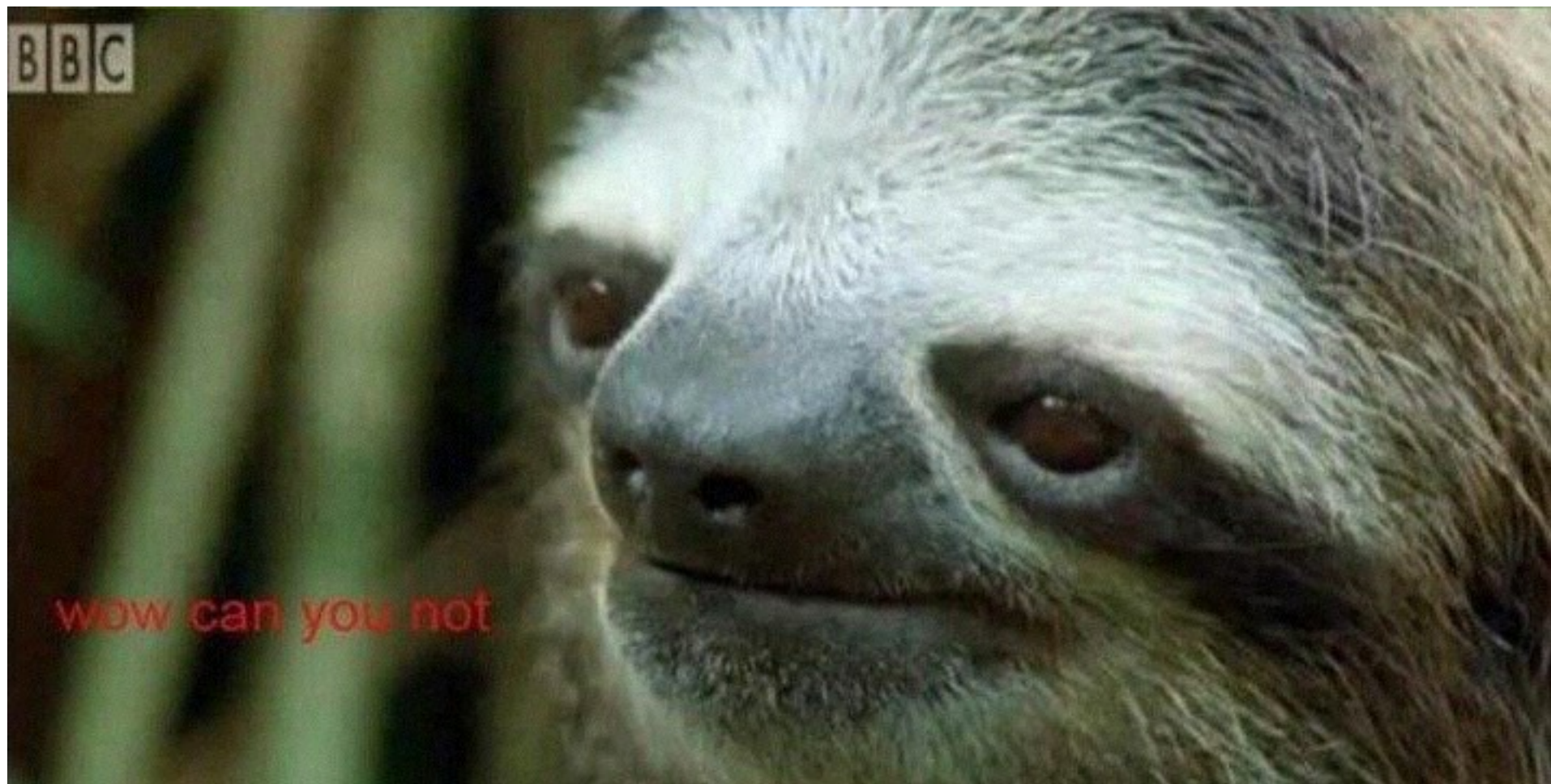


```
1  function dostuff() {
2      var item = findById(getId());
3      setActive(item);
4  }
5
6  function findById(id) {
7      var item = _.findWhere(store, { id: id });
8      console.log(id); // '1'
9      return item;
10 }
11
12 var store = [
13     {id: 1, name: 'Anton'},
14     {id: 2, name: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



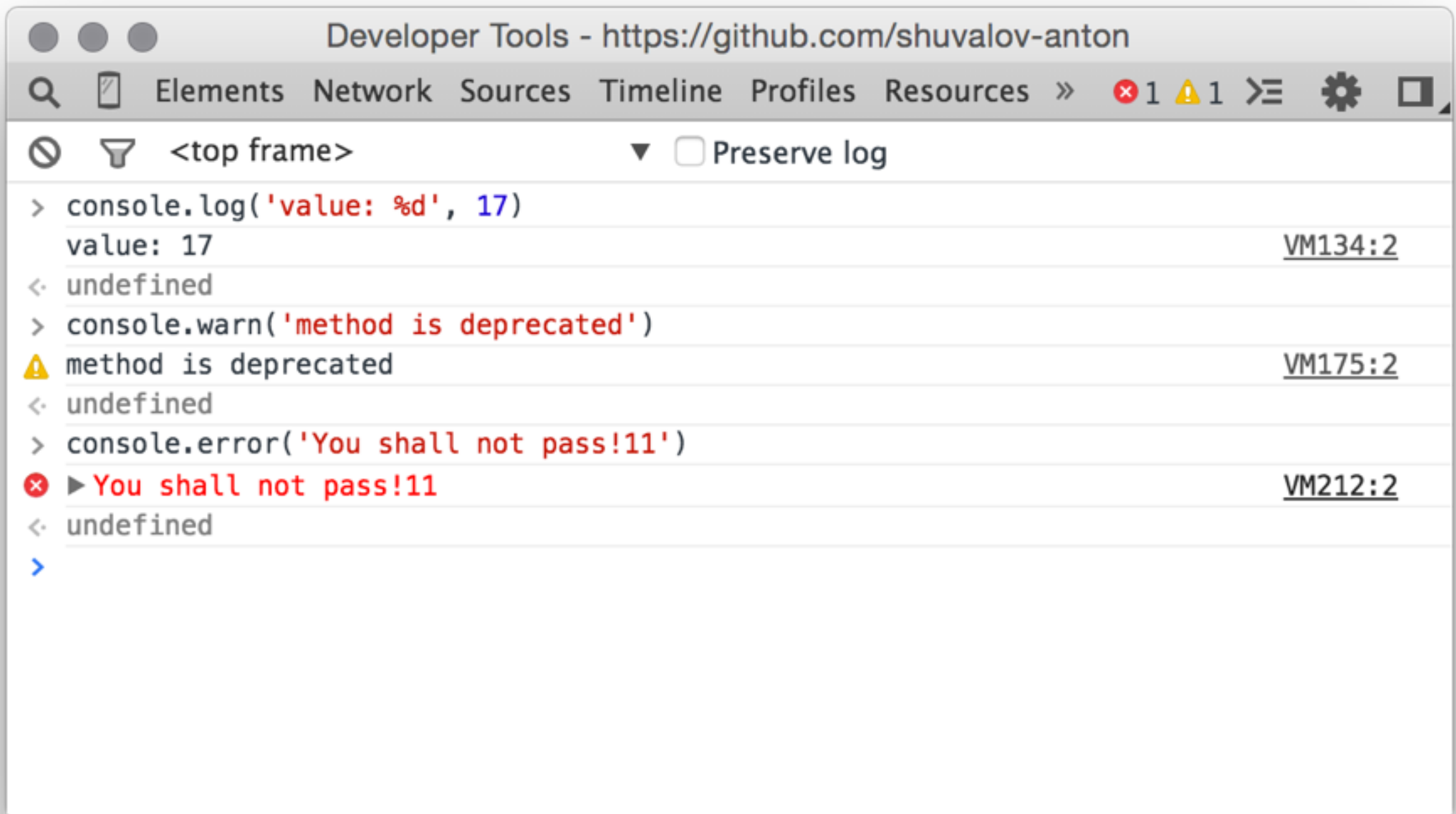
console.log — не отладчик



HO...



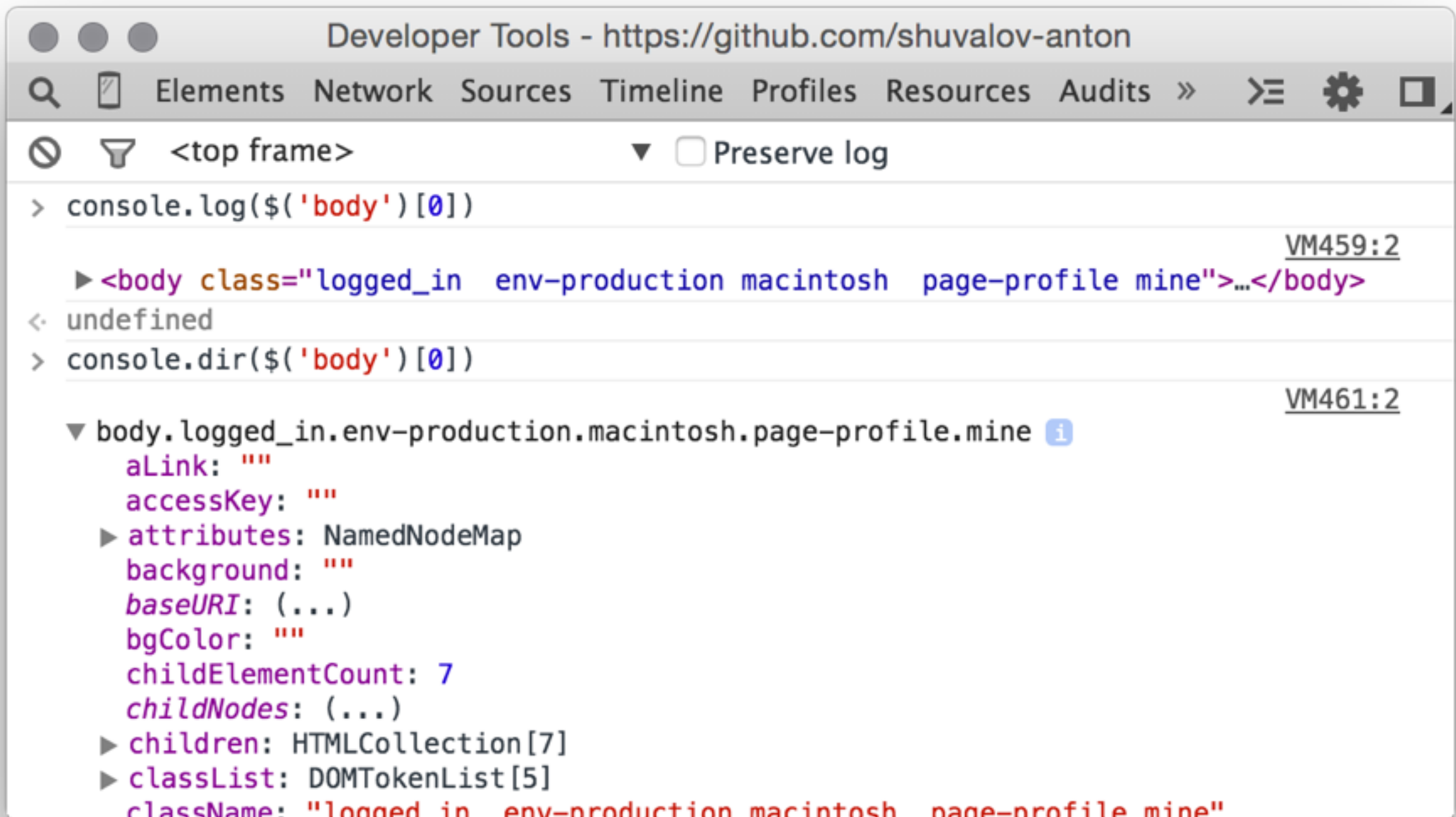
зачем нам console?



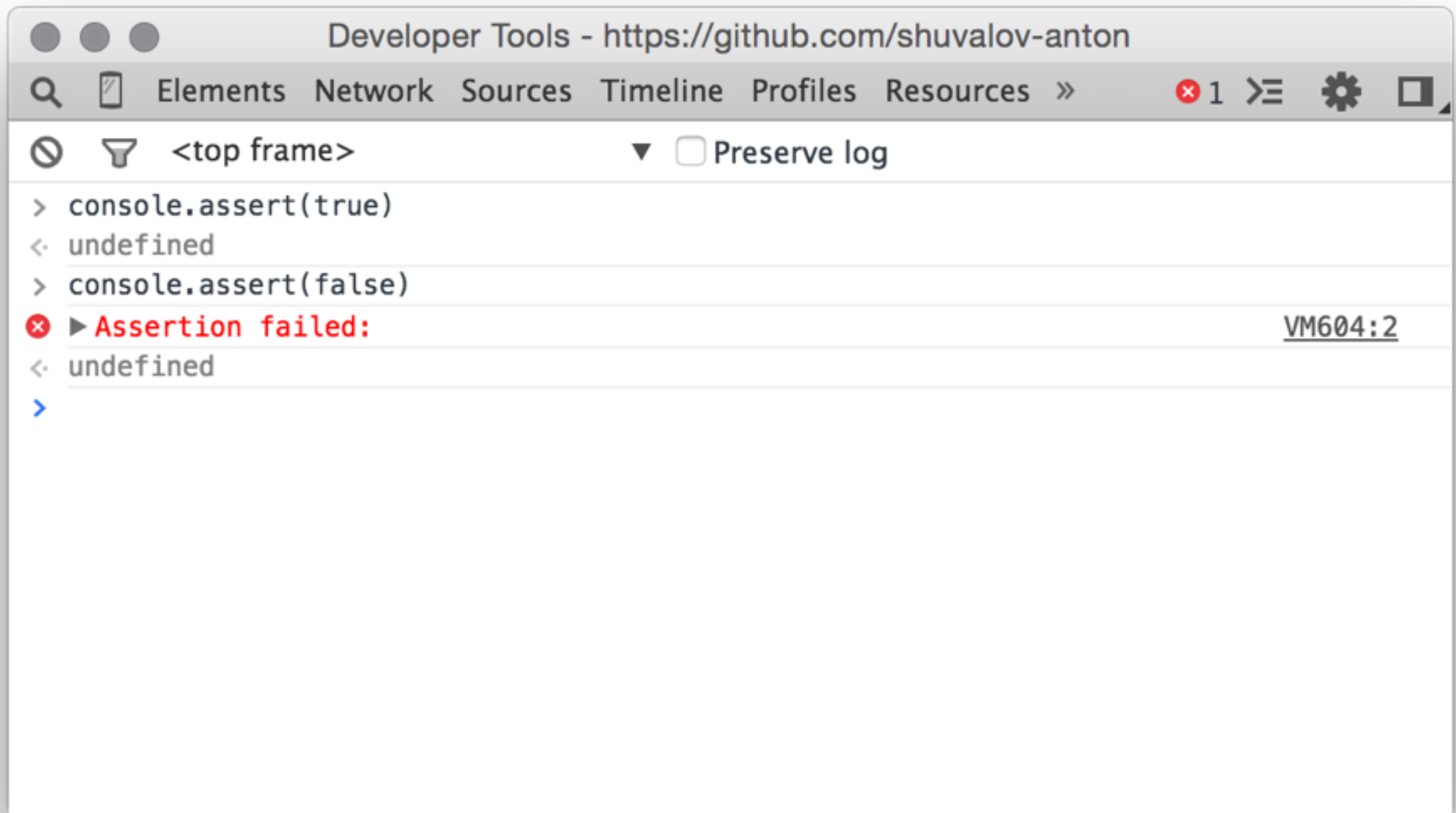
console



console.group



console.dir



console.assert



console.time



console.profile


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

Elements Network Sources Timeline Profiles Resources Audits » ⚙️

Heavy (Bottom Up) ▼ 🔍 ✕ ↺

Profiles

CPU PROFILES

 find element [Save](#)

Self ▼	Total	Function
0.0 ms 0.00%	0.0 ms 0.00%	Inject... VM84:679
0.0 ms 0.00%	0.0 ms 0.00%	▶ Injected... VM84:813
0.0 ms 0.00%	0.0 ms 0.00%	▶ ⚠ Inject... VM84:850
0.0 ms 0.00%	0.0 ms 0.00%	▶ evaluateWithExcep...
0.0 ms 0.00%	0.0 ms 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/

ElementsNetworkSourcesTimelineProfilesResourcesAuditsConsole

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

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

ConsoleSearchEmulationRendering

<top frame> Preserve log

Filter Regex All Errors Warnings Info Logs Debug Hide network messages

Uncaught Error: no item index.js:15

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

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

Uncaught Error: no item index.js:15

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

☐ ☒ <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
15   if (!item) thro
16   console.log('se
17 }
18
19 var store = [
20   {id: 1, name: '
21   {id: 2, name: '
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 +

Enter Breakpoints

Array[2]

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

{ } Line 11, Column 23

Console Search Emula

<top frame>

Filter ☐ Regex ☐ 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) {
15   // ...
16 }
```

5 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
>
```

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
15 if
16 on
17 }
18
19 var s
20 id
31 ch
```

Object

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

Watch Expressions + ↺

Call Stack ☐ Async

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 funct
15 if
16 con
17 }
18
19 var s
20 {id
21
22 }
23
24 {} 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!

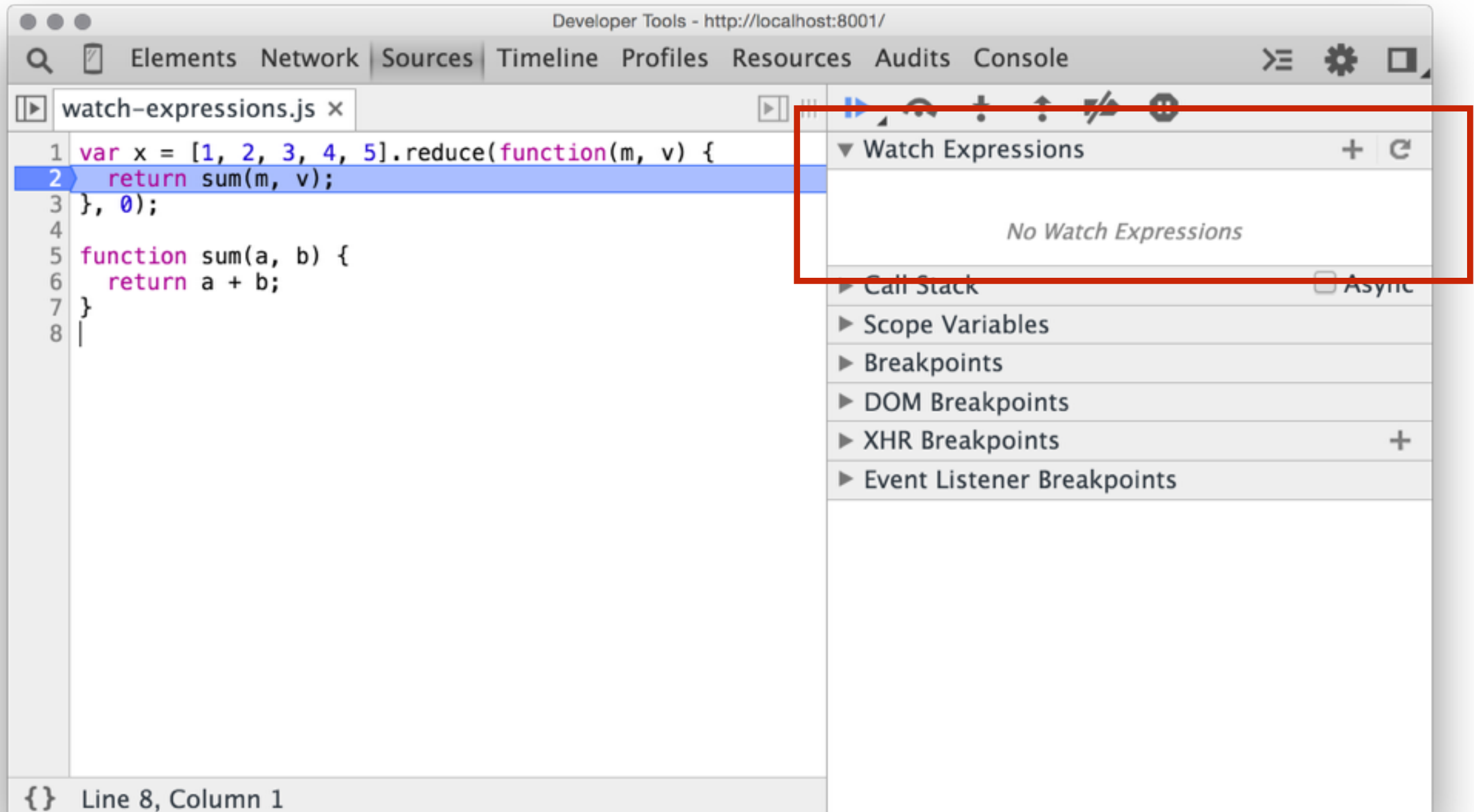


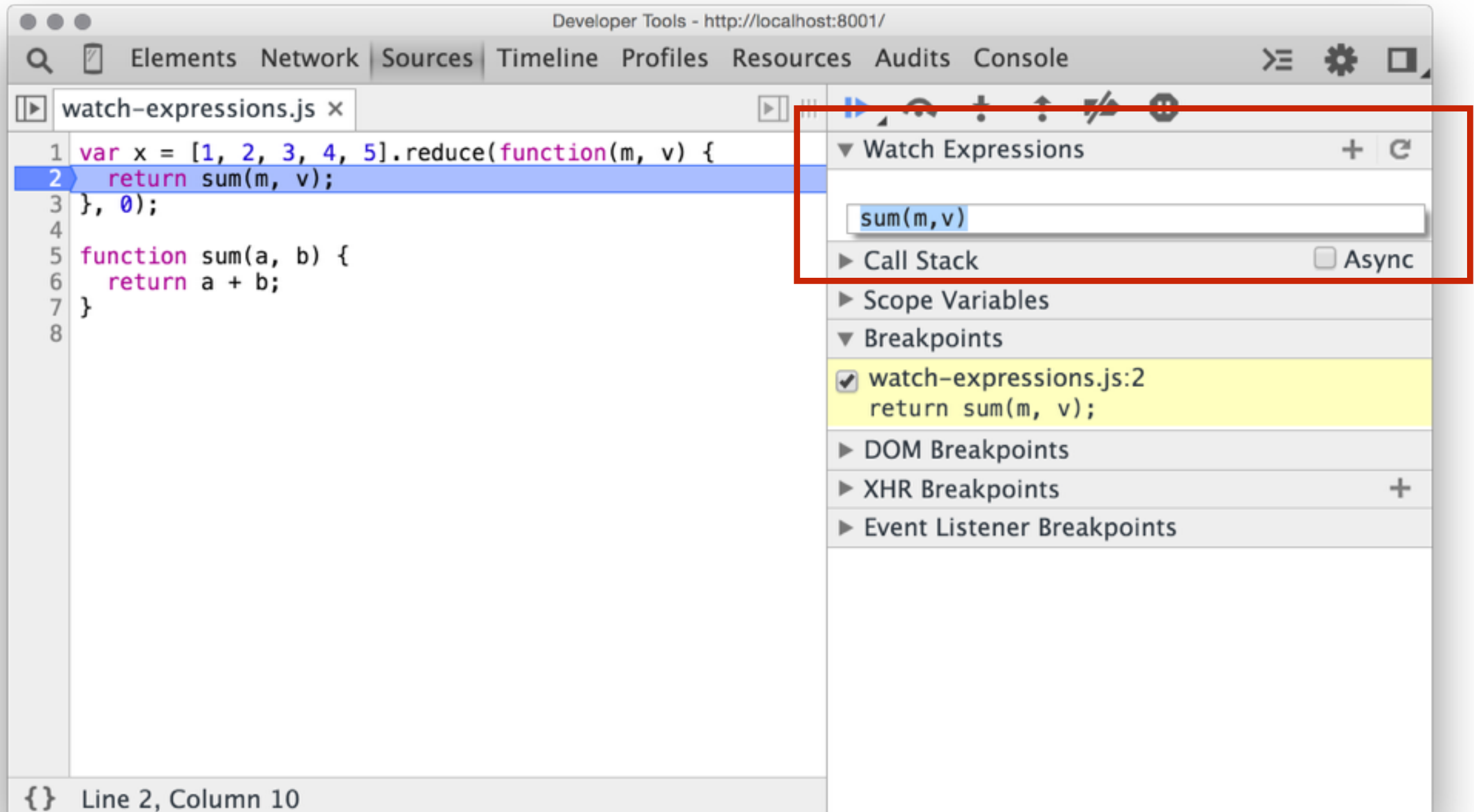
ничоси



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 + ↻

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
```

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:8000
 (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
```

Line 6, Column 16

▼ Watch Expressions + ↺
fn(): <not available>

▼ 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
dispatch	stack.js:15
(anonymous function)	stack.js:18

▶ Scope Variables

▼ Breakpoints

No Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

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

```

Line 13, Column 21

Watch Expressions

fn(): 4

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
dispatch	stack.js:15
(anonymous function)	stack.js:18

Scope Variables

Breakpoints

No Breakpoints

DOM Breakpoints

XHR Breakpoints

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
```

Line 13, Column 27

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
dispatch	stack.js:15
(anonymous function)	stack.js:18

Scope Variables

Breakpoints

No Breakpoints

DOM Breakpoints

XHR Breakpoints

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.

```
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 }
```

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:8001
 (index)
 async.js
▶ underscorejs.

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

Line 6, Column 1

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

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:8001
 (index)
 async.js
▶ underscorejs.

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

▶ Watch Expressions
▼ Call Stack
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)
▶ code.jquery.c
▼ localhost:8001
 (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	
<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:8001
 (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

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

▶ (no domain) 1 \$.get('/404').error(onError);
▶ code.jquery.com 2
▼ localhost:8000 3 function onError() {
(index) 4 debugger;
async.js 5 }
underscorejs. 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	
<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 1, Column 3

Карты кода

Developer Tools - http://localhost:8000/

ElementsNetworkSourcesTimelineProfilesResourcesAuditsConsole

app.js x

1const plugins = [];

2

3class App {

4

5static use(...args) {

6return plugins.push(...args);

7}

8

9init() {

10plugins.map(plugin => plugin.init(this));

11}

12

13}

14

▶ Watch + ↺

▶ Call Stack ☐ Async

▶ Scope

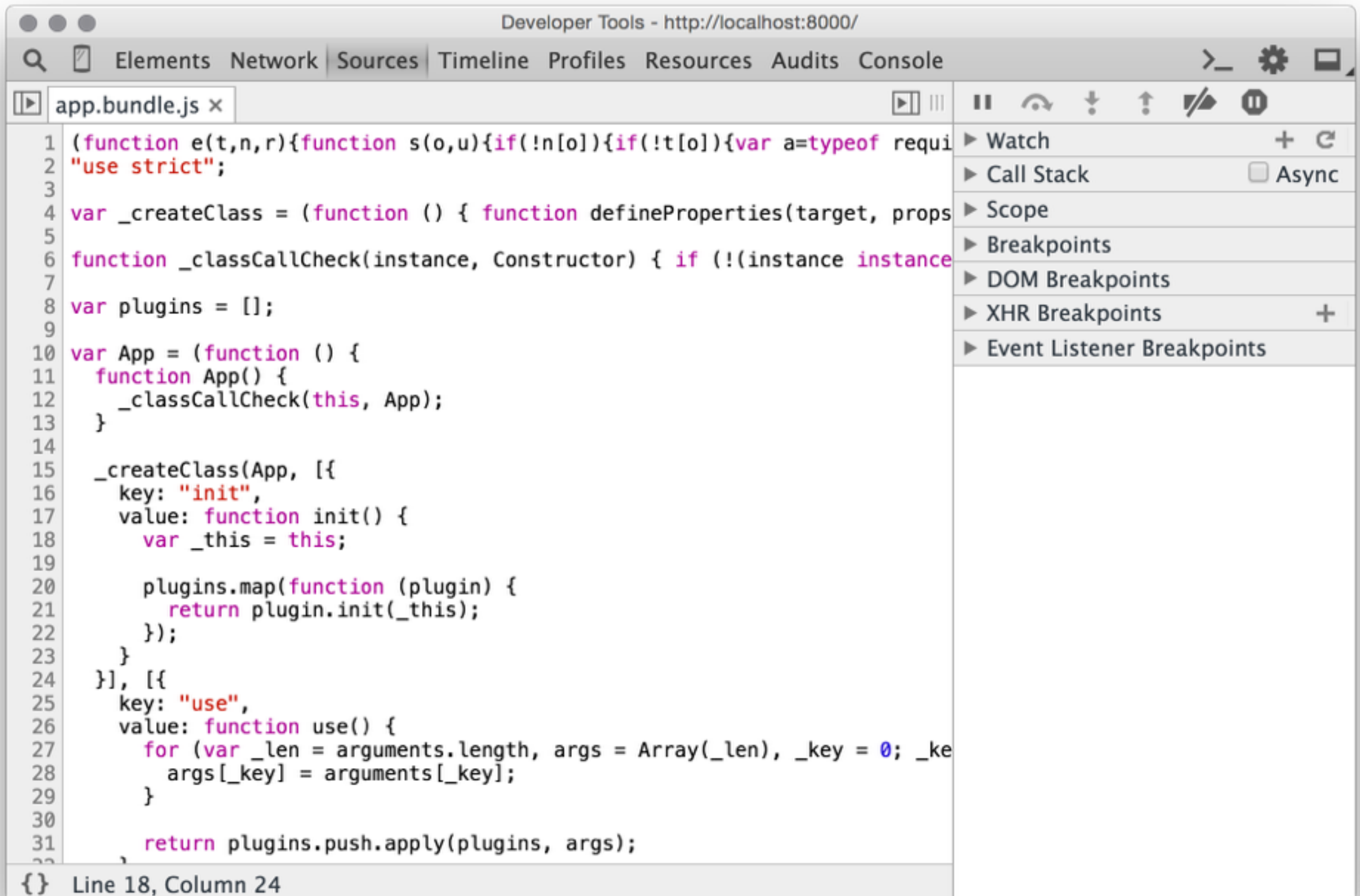
▶ Breakpoints

▶ DOM Breakpoints

▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 5, Column 15



Developer Tools - http://localhost:8000/

ElementsNetworkSourcesTimelineProfilesResourcesAuditsConsole

app.js x

1const plugins = [];

2

3class App {

4

5static use(...args) {

6return plugins.push(...args);

7}

8

9init() {

10plugins.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

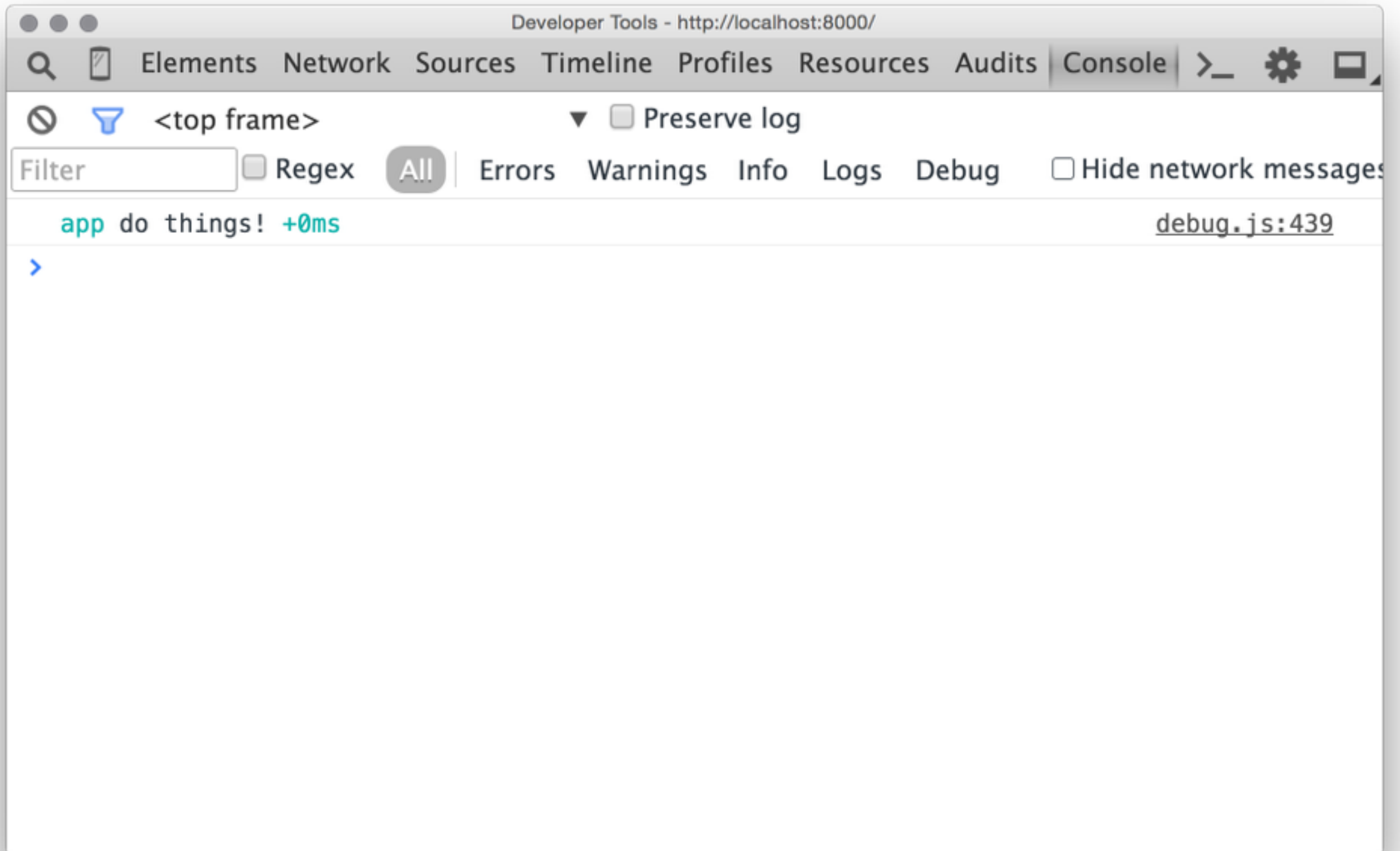
▶ Breakpoints

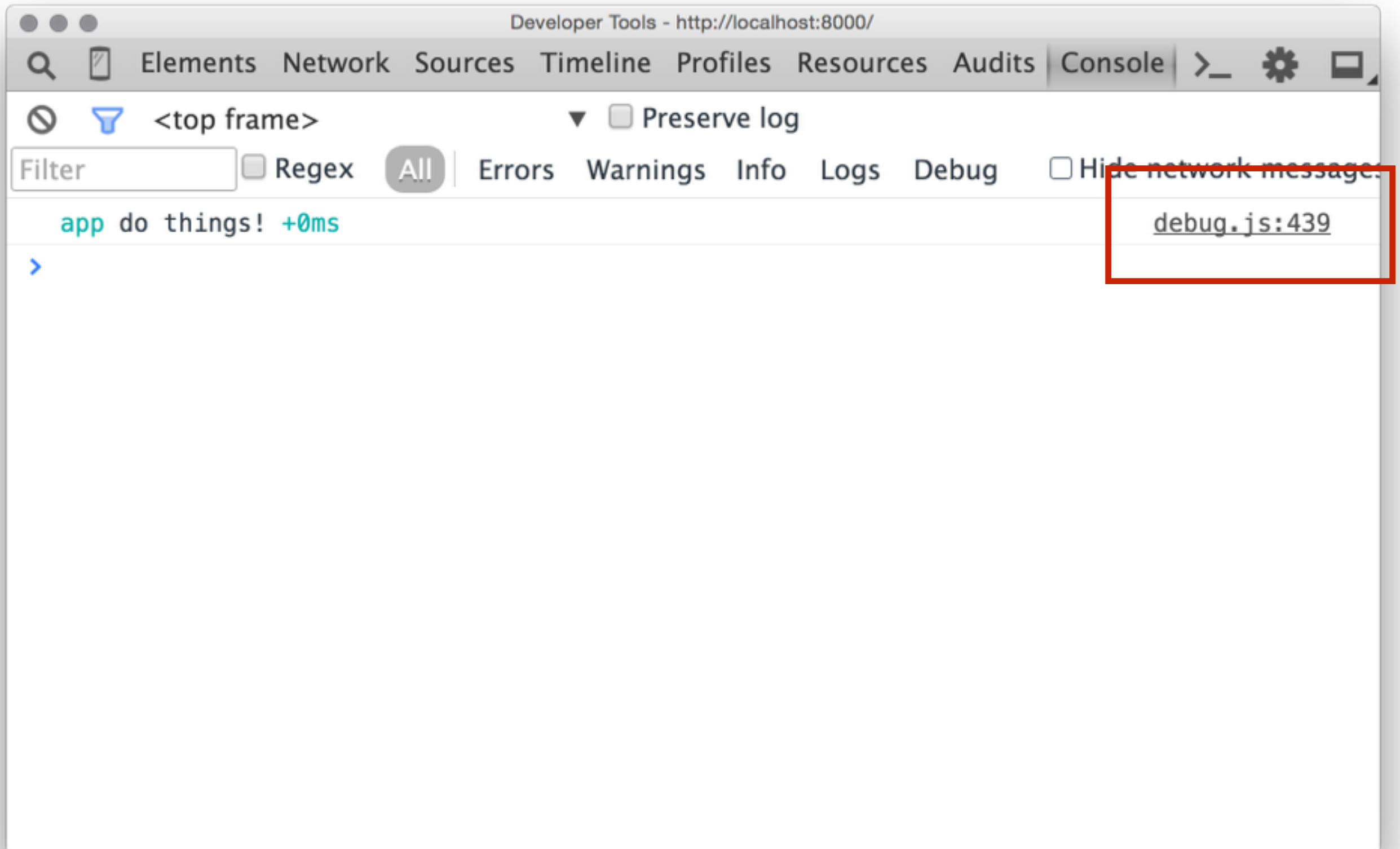
▶ DOM Breakpoints

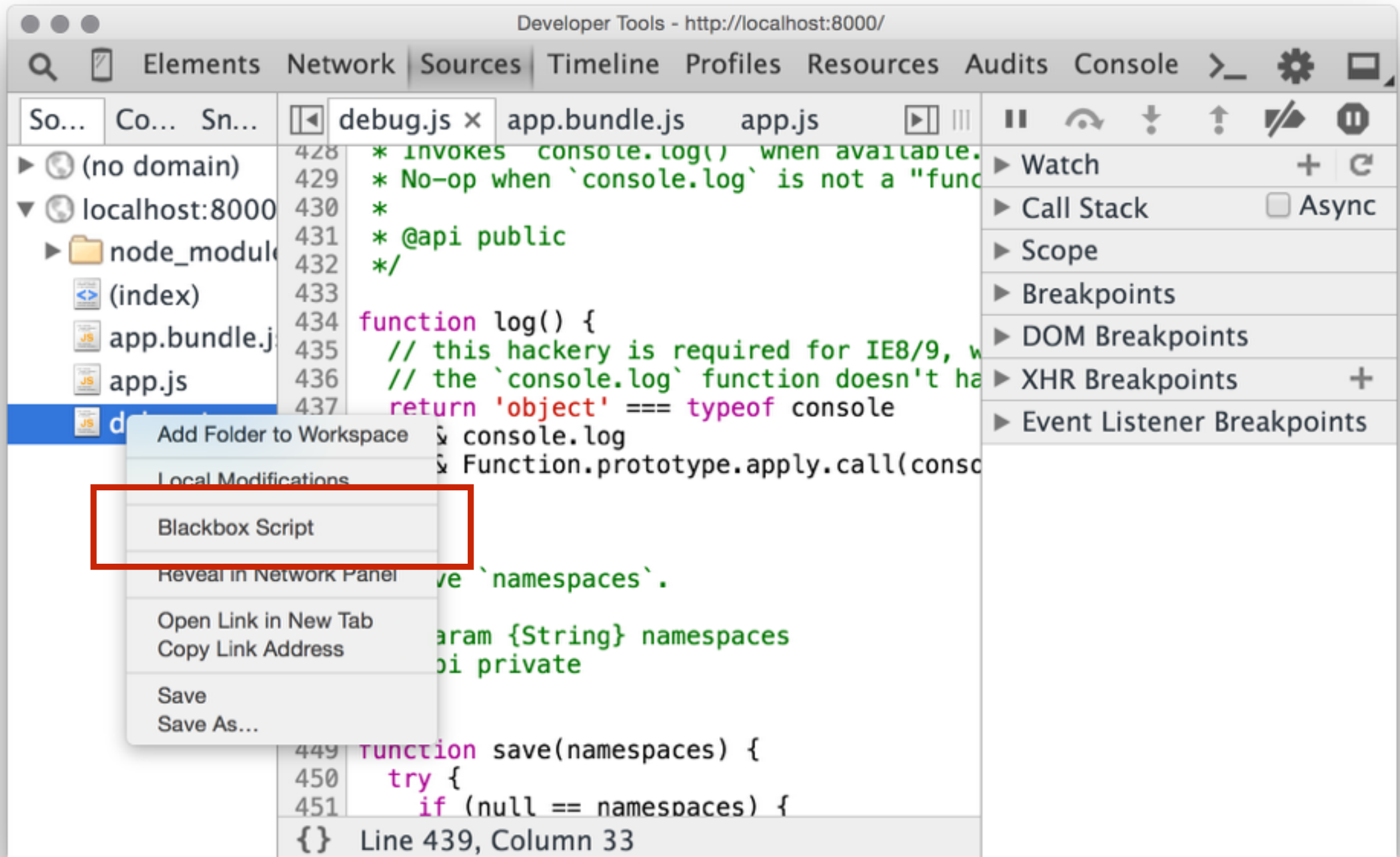
▶ XHR Breakpoints +

▶ Event Listener Breakpoints

{ } Line 4, Column 1







Developer Tools - http://localhost:8000/

Elements Network Sources Timeline Profiles Resources Audits Console

So... Co... 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 ☐ Async

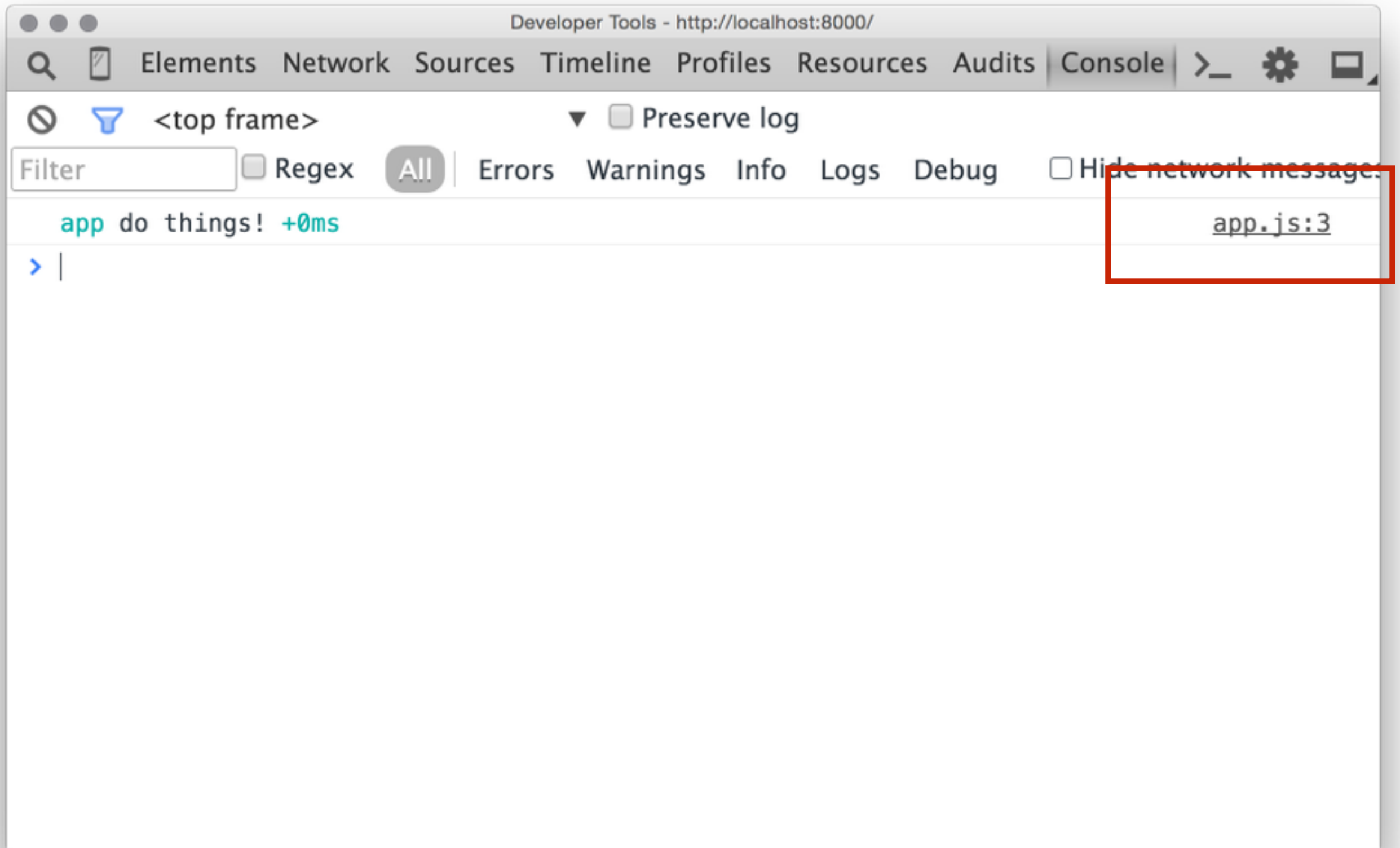
▶ Scope

▶ 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

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

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

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

{ } Line 3, Column 1

Отладчик не заходит
внутрь 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