

JavaScript

现代化排错实践

赵劼 - 2012.9

关于我

- 赵劼 / 老赵 / Jeffrey Zhao / 赵姐夫
- 日写代码三百行，不辞长作程序员
- 博客：<http://blog.zhaojie.me/>
- 微博：@老赵
- F#, JavaScript, Scala, C#, Python, .NET, Mono...
- 痛恨Java语言

内容提纲

- 代码调试检查
- 调用堆栈分析
- 调试混淆代码
 - Source Map

代码调试

代码调试

- 很久很久以前: `alert`
- 过了一段时间: `console.log`
- 其实从IE 5时代开始便可以调试代码

两个问题

- Node.js可以调试吗？
- eval出来的代码可以调试吗？

调试Node.js代码

- Eclipse
- Eclipse Debugger Plugin for V8
- 启动程序
 - `node --debug[=port] app.js`
 - `node --debug-brk[=port] app.js`

调试eval的代码

- 代码末尾: `//@ sourceMappingURL=<path>`
- 支持浏览器
 - Chrome
 - Safari
 - Firefox (Firebug)

调用堆栈分析

调用堆栈

✖ ▼ Uncaught Error: Something got wrong! 1-clean-stack.js:13
 MyClass.baz 1-clean-stack.js:13
 MyClass.bar 1-clean-stack.js:9
 MyClass.foo 1-clean-stack.js:5
 (anonymous function) 1-clean-stack.js:18

▼ Call Stack

MyClass.baz	1-clean-stack.js:13
MyClass.bar	1-clean-stack.js:9
MyClass.foo	1-clean-stack.js:5
(anonymous function)	1-clean-stack.js:18

命名函数表达式

// 函数定义

```
function Identifier(FormalParameterListopt) { FunctionBody }
```

// 函数表达式

```
function Identifieropt(FormalParameterListopt) { FunctionBody }
```

分辨一下

// 声明：程序的一部分
function foo() { };

// 表达式：分组操作符中只能包含表达式
(function foo() { });

// 表达式：赋值表达式 (AssignmentExpression) 的一部分
var bar = function foo() { };

// 表达式：New表达式 (NewExpression) 的一部分
new function bar() { };

```
(function () {  
    // 声明：函数体 (FunctionBody) 的一部分  
    function bar() { };  
})();
```

调试混淆代码

调试混淆代码

```
1 (function(){var g,j,h=function(b){return"[object Array]"===Object.prototype.toString.call(b)},m=function(b,a,c){if(h(b)){for(var
2 arguments[d];return l(b,c)}return b.replace(/\{\{1,2\}\d+\}\{1,2\}/g,function(b){if(0===b.indexOf("{")&&b.indexOf("}")===b.length-2)r
3 e))return e},isEmpty:function(b){return h(b)?0===b.length:!j.each(b,function(){return!0}}),map:m,clone:function(b){return m(b)},v
4 n.prototype={log:function(b,a){if(this.level<=b)try{console.log(a)}catch(c){}},trace:function(b){this.log(i.TRACE,b)},debug:funct
5 [],function(a,b){var c=g.modules[a];if(!c)throw Error(j.format("Missing required module: \"{0}\" (expected version: \"{1}\").",a,b));
6 d=!("function"===typeof require&&"function"===typeof define&&define.amd),f=function(b){var e=b.autoloads||[];if(a){var f=b.requi
7 module.exports,e();else if(d)define("wind-core",function(){g={};e();return g});else{var o=Function("return this")();if(o.Wind)thr
8 c){if("normal"===b||"continue"===b)e(f+1);else if("throw"===b||"return"===b)a(b,c);else if("break"===b)a("normal");else throw Error('I
9 Do: function(i,g){return{next: function(c,a){var d=function(){i.next(c,function(f,e){if("normal"===f||"continue"===f)try{g.call(c)?d(
10 g)}else a(d,f,e)}}},Return: function(g){return{next: function(h,c){c("return",g)}},Normal: function(){return{next: function(g,h){h
11 h.call(a,e)}catch(i){c?c.next(a,function(a,b,c){c("normal"===a?d("throw",i):d(a,b,c)):d("throw",i)}b&&b.next(a,function(b,e,f){c?c.
12 init: function(){m.BuilderBase=g}});if(j){try{m=require("./wind-core")}catch(l){m=require("wind-core")}k()}else if(h)require(["wi
13 a.prototype={_typeId:"4a73efb8-c2e2-4305-a05c-72385288650a",message:"This is an error contains sub-errors, please check the 'chil
14 1)}},cancel: function(){if(!this.isCancellationRequested){this.isCancellationRequested=!0;var a=this._handlers;delete this._handle
15 d||(d=this._listeners[a]=[]);d.push(b)},remove: function(a,b){if(this._firing===a){var c=this;setTimeout(function(){c.remove(a,b)
16 "ready"};e.prototype={start: function(){if("ready"!==this.status)throw Error("Task can only be started in \"ready\" status.");this.st
17 var c=this._eventManager;this._eventManager=null;"success"===a?(this.status="succeeded",g?this._result=b:this._result=b,c.fire("su
18 if(!g)return this.error;var a=this._unobservedTimeoutToken;a&&(clearTimeout(a),this._unobservedTimeoutToken=null);this._errorObse
19 this.error);var b;try{b=a(this._result)}catch(e){return c.complete("failure",e)}"ready"===b.status&&b.start();if("running"===b.statu
20 this._eventManager;if(!a)throw Error("All the event listeners have been removed when the task was complete.");a.remove.apply(a,ar
21 b){f.remove.apply(f,arguments)};e.unobservedTimeout=1E4;var b=e.isTask=function(a){return a&&typeof a.start==="function"&&typeof
22 new a(e));else{var b=l.map(c,function(a){return a._result});d.complete("success",b)},g=0;l.each(c,function(a,d){if(d){b(d)||c[a]
23 else{c=l.isArray(d);l.each(d,function(c,d){b(d)&&(a[c]=d)}))}else for(d=0;d<arguments.length;d++){var e=arguments[d];b(e)&&(a[d]=
24 c)}var d=function(){this.observeError();var c=this,e;l.each(a,function(a,b){if(c===b)e=a;else{b.off("complete",d);b.on("failure",
25 i.onEvent=function(a,b,d){return e.create(function(e){d&&d.isCancellationRequested&&e.complete("failure",new c);var f=function(){
26 p.prototype={Start: function(a,b){return e.create(function(c){b.next(a,function(a,b){if(a==="normal"||a==="return")c.complete("succe
27 e):e(a)}}});var h=i.Binding={},s=function(a,b){for(var c=[],d=0;d<a.length;d++)c.push(a[d]);for(;c.length<b;)c.push(void 0);retu
28 d)}e.complete("success",c)});a.apply(c,d)});h.fromCallback=function(a){var b=t(arguments);return function(){var c=this,d=s(arg
29 version:"0.7.0",require:u&&require,autoloads:["builderbase"],dependencies:{builderbase:"~0.7.0"},init: function(){l=k._;l.each(k.B
30 SortingAnimations=function(g){var j=g.getContext?g.getContext("2d"):null,h=function(c,a){j.clearRect(0,0,g.width,g.height);j.line
31 Wind.builders.async;return f.Start(this,f.Delay(function(){var e=c[a];c[a]=c[d];c[d]=e;h(c,[a,d]);return f.Bind(Wind.Async.sleep(
32 0;return a.For(function(){return f<c.length-d-1},function(){f++;a.Delay(function(){return a.Bind(m(c[f],c[f+1]),function(d){retu
33 b.Delay(function(){return b.Bind(m(c[f],h),function(a){if(0>a)f++;else return b.Break();return b.Normal()}})),b.Combine(b.While(
34 c),function(d){return b.Combine(b.Delay(function(){return a<d-1?b.Bind(f(a,d-1),function(){return b.Normal()})}:b.Normal()),b.Del
35 d+1;return a.For(function(){return e<c.length},function(){e++;a.Delay(function(){return a.Bind(m(c[e],c[f]),function(c){0>c&&(f=
36 a.Delay(function(){var e=c[f],g;return a.Combine(a.Delay(function(){g=f;return a.For(function(){return g>=d},function(){g=d},a.D
37 0.5});return c};this.paint=h;this.names=[];for(var n in i)this.names.push(n);this.sortAsync=function(c,a){var d=Wind.builders.asy
```


格式化以后

```
17 ,complete:function(a,b){
    if("success"!==a&&"failure"!==a)throw Error("Unsupported type: "+a);
    if("running"!==this.status)throw Error('The "complete" method can only be called in "running"');
    var c=this._eventManager;
    this._eventManager=null;
    "success"===a?(this.status="succeeded",g?this._result=b:this.result=b,c.fire("success",this))
    :c.fire("complete",this);
    if("failure"===a&&g&&!this._errorObserved){
        var d=this;
        this._unobservedTimeoutToken=setTimeout(function(){
            d._handleUnobservedError(b)
        },e.unobservedTimeout)
    }
}
18 ,observeError:function(){
    if("ready"===this.status||"running"===this.status)throw Error("The method could only be called in 'ready' or 'running' state");
    if(!g)return this.error;
    var a=this._unobservedTimeoutToken;
    a&&(clearTimeout(a),this._unobservedTimeoutToken=null);
    this._errorObserved=!0;
}
```

配合Source Map

```
48 var compareAsync = /* async << function (x, y) { */ (function (x, y) {
49                                     var _builder_$0 = Wind.builders["async"];
50                                     return _builder_$0.Start(this,
51                                     _builder_$0.Delay(function () {
52 /*      $await(Wind.Async.sleep(compareCost)); */      return _builder_$0.Bind(Wind.Async.sleep(compareCost), function () {
53 /*      return x - y; */                                return _builder_$0.Return(x - y);
54                                                         });
55                                                         });
56                                                         });
57 /* } */
58
59 var swapAsync = /* async << function (array, i, j) { */ (function (array, i, j) {
60                                     var _builder_$0 = Wind.builders["async"];
61                                     return _builder_$0.Start(
62                                     _builder_$0.Delay(
63 /*      var t = array[i]; */                                var t = array[i];
64 /*      array[i] = array[j]; */                            array[i] = array[j];
65 /*      array[j] = t; */                                array[j] = t;
66 /*      paint(array, [i, j]); */                          paint(array, [i, j]);
67 /*      $await(Wind.Async.sleep(updateCost)); */          return _builder_$0.Return(
68                                                         return _builder_$0.Return(
69                                                         });
70                                                         });
71                                                         });
72 /* } */
73
74 var assignAsync = /* async << function (array, i, value, updating) { */ (function (array, i, value, updating) {
75                                     var _builder_$0 = Wind.builders["async"];
76                                     return _builder_$0.Start(
77                                     _builder_$0.Delay(
78 /*      array[i] = value; */                                array[i] = value;
79 /*      paint(array, updating); */                        paint(array, updating);
80 /*      $await(Wind.Async.sleep(updateCost)); */          return _builder_$0.Return(
81                                                         return _builder_$0.Return(
82                                                         });
83                                                         });
84                                                         });
85 /* } */
```

Array[30]
0: 15
1: 16
2: 22
3: 19
4: 3
5: 6
6: 27
7: 18
8: 26
9: 12
10: 13
11: 1
12: 2
13: 30
14: 29
15: 5
16: 28
17: 9
18: 4
19: 11

Source Map

- 记录目标代码到源代码的映射
- JSON格式 + 编码后的映射数据
- Source Map V3 Spec

Source Map

- 记录目标代码到源代码的映射
- JSON格式 + 编码后的映射数据
- Source Map V3 Spec

脚本末尾加上: `//@ sourceMappingURL=<path>`, 或
脚本请求头加上: `X-SourceMap: <path>`

深入Source Map

Source Map V3

```
{  
  "version": 3,  
  "file": "all.min.js",  
  "lineCount": 37,  
  "sources": [  
    "wind-core.js",  
    "wind-builderbase.js",  
    "wind-async.js",  
    "sorting-animations.aot.js"  
  ],  
  "names": [ "Wind", "_", "isArray", "obj", ... ],  
  "mappings": "AAAC,SAAS,EAAG,CAGT,IAAIA, ..."  
}
```

解码 mappings 字段

- 用分号区分“行”，逗号区分“段”。
- Base64 Variable-Length Quantity 编码
- 节省空间，比V2节省50%左右

确定代码行号

AABBC;	// 第1行
KAUYM, GAKoEF;	// 第2行
CCDD, ...;	// 第3行
...	
...	

Base64 VLQ解码

KAUYM, GAKoEF

Base64 VLQ解码

KAUYM, GAKoEF

∇
∇

[10, 0, 20, 24, 12], [6, 0, 10, 40, 4, 5]

Base64 VLQ解码

KAUYM, GAKoEF

√
√

[10, 0, 20, 24, 12], [6, 0, 10, 40, 4, 5]

√
√

[001010, 000000, 010100, 011000, 001100],
[000110, 000000, 001010, 101000, 000100, 000101]

Base64 VLQ解码

KAUYM, GAKoEF

√
√

[10, 0, 20, 24, 12], [6, 0, 10, 40, 4, 5]

√
√

[001010, 000000, 010100, 011000, 001100],

[000110, 000000, 001010, 101000, 000100, 000101]

最低N-1位为数据位

最高位表示是否“连接后续数据”

Base64 VLQ解码

KAUYM, GAKoEF

↓
↓

[10, 0, 20, 24, 12], [6, 0, 10, 40, 4, 5]

↓
↓

[001010, 000000, 010100, 011000, 001100],
[000110, 000000, 001010, 101000, 000100, 000101]

最低N-1位为数据位

↓
↓

[1010, 0, 10100, 11000, 1100],
[110, 0, 1010, 10001000, 101]

最高位表示是否“连接后续数据”

数据解码

[1010, 0, 10100, 11000, 1100],
[110, 0, 1010, 10001000, 101]

数据解码

[1010, 0, 10100, 11000, 1100],

[110, 0, 1010, 10001000, 101]

最高N-1位为数据位

最低位为符号位

数据解码

[1010, 0, 10100, 11000, 1100],

[110, 0, 1010, 10001000, 101]

最高N-1位为数据位

最低位为符号位

√
√

[5, 0, 10, 12, 6], [3, 0, 5, 68, -2]

数据解码

[1010, 0, 10100, 11000, 1100],

[110, 0, 1010, 10001000, 101]

最高N-1位为数据位

最低位为符号位

√
√

[5, 0, 10, 12, 6], [3, 0, 5, 68, -2]

√
√

[5, 0, 10, 12, 6], [8, 0, 15, 80, 4]

含义

```
// 已确定行号  
[  
    5,    // 列号  
    0,    // 源文件，从sources查找  
    10,   // 源文件内行号  
    12,   // 源文件内列号  
    6     // 源文件内标示符，从names查找  
]
```


无需手动分析

- Google Closure Compiler可以生成V2版Source Map格式，未编码的明文数据
- 使用Mozilla的SourceMap项目读取或生成Source Map文件

总结

- 使用调试器辅助JavaScript开发
 - 即使是eval生成的代码也没有问题
- 使用命名函数表达式来美化堆栈
- 使用Source Map来调试混淆后的代码
 - 手动使用Source Map亦不是问题

Q & A

谢谢