

基于 SeaJS 的调试实践

玉伯

2012.9.14 @沪JS

关于我

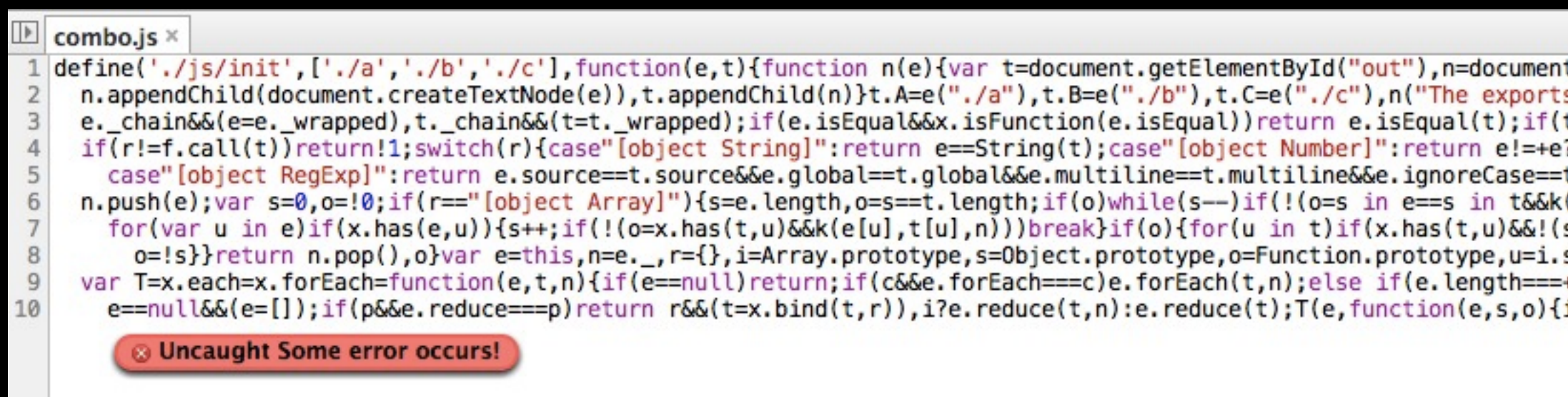
- 王保平 / lifesinger / 玉伯
- 支付宝 - 前端开发部
- KISSY、SeaJS、Arale 2
- 微博: @玉伯也叫射雕



Topics

- I. 解 combo 和源码映射: map
- II. 让继承链清晰: class
- III. bug fix 与 mock: modify
- IV. 释放 F5 键: reload
- V. 原理

Problem 1



```
1 define('./js/init', ['./a', './b', './c'], function(e, t) { function n(e) { var t = document.getElementById("out"), n = document
2   n.appendChild(document.createTextNode(e)), t.appendChild(n) } t.A = e("./a"), t.B = e("./b"), t.C = e("./c"), n("The exports
3   e._chain && (e = e._wrapped), t._chain && (t = t._wrapped); if (e.isEqual && x.isFunction(e.isEqual)) return e.isEqual(t); if (t
4   if (r != f.call(t)) return !1; switch (r) { case "object String": return e == String(t); case "object Number": return e != +e?
5     case "object RegExp": return e.source == t.source && e.global == t.global && e.multiline == t.multiline && e.ignoreCase == t
6   n.push(e); var s = 0, o = !0; if (r == "object Array") { s = e.length, o = s == t.length; if (o) while (s--) if (!(o = s in e == s in t && k
7     for (var u in e) if (x.has(e, u)) { s++; if (!(o = x.has(t, u) && k(e[u], t[u], n))) break } if (o) { for (u in t) if (x.has(t, u) && !(s
8     o = !s } } return n.pop(), o } var e = this, n = e._, r = {}, i = Array.prototype, s = Object.prototype, o = Function.prototype, u = i.s
9   var T = x.each = x.forEach = function(e, t, n) { if (e == null) return; if (c && e.forEach == c) e.forEach(t, n); else if (e.length == +
10  e == null && (e = [])); if (p && e.reduce == p) return r && (t = x.bind(t, r)), i ? e.reduce(t, n) : e.reduce(t); T(e, function(e, s, o) {
```

Uncaught Some error occurs!









1. 压缩导致难以调试
2. 合并导致难以定位到具体文件

Solution



1. 通过 `?seajs-debug` 开启调试
2. 通过 `map config` 解 `combo` 并映射到本地文件

Result

Name	Path	Method
	/~lifesinger/lifesinger/lifesinger.github.c	GET
	sea.js	GET
	plugin-debug.js	GET
	map-config.js	GET
	init.js	GET
	a.js	GET
	b.js	GET
	c.js	GET

sea.js plugin-debug.js a.js x

```
1035     wrapper.prototype[name] = function() {
1036         var wrapped = this._wrapped;
1037         method.apply(wrapped, arguments);
1038         var length = wrapped.length;
1039         if ((name == 'shift' || name == 'splice') && length === 0) d
1040         return result(wrapped, this._chain);
1041     };
1042 });
1043
1044 // Add all accessor Array functions to the wrapper.
1045 each(['concat', 'join', 'slice'], function(name) {
1046     var method = ArrayProto[name];
1047     wrapper.prototype[name] = function() {
1048         return result(method.apply(this._wrapped, arguments), this._
1049     };
1050 });
1051
1052     throw 'Some error occurs!';
1053
1054 // Start chaining a wrapped Underscore object.
1055 wrapper.prototype.chain = function() {
1056     this._chain = true;
1057     return this;
1058 };
1059
```

Uncaught Some error occurs!

<https://github.com/seajs/seajs/issues/270>

Problem 2

```
> var lifesinger = new NodeJSProgrammer()
  console.dir(lifesinger)
▼ Object
  $caller: null
  caller: null
  ▼ __proto__: Object
    ▶ $constructor: function () { e(this); if (g.$prototyping) { return this; } this.$caller = null; }
    ▶ constructor: function () { e(this); if (g.$prototyping) { return this; } this.$caller = null; }
    ▶ parent: function () { if (!this.$caller) { throw new Error('The method "parent" cannot be called'); } }
    ▼ __proto__: Object
      ▶ __defineGetter__: function __defineGetter__() { [native code] }
      ▶ __defineSetter__: function __defineSetter__() { [native code] }
      ▶ __lookupGetter__: function __lookupGetter__() { [native code] }
      ▶ __lookupSetter__: function __lookupSetter__() { [native code] }
      ▶ constructor: function Object() { [native code] }
      ▶ hasOwnProperty: function hasOwnProperty() { [native code] }
      ▶ isPrototypeOf: function isPrototypeOf() { [native code] }
      ▶ propertyIsEnumerable: function propertyIsEnumerable() { [native code] }
      ▶ toLocaleString: function toLocaleString() { [native code] }
      ▶ toString: function toString() { [native code] }
      ▶ valueOf: function valueOf() { [native code] }
```

看不懂的继承信息

Solution

```
Programmer.js x JavaScriptProgrammer.js x NodeJSProgrammer.js x init.js
1 define(function(require, exports, module) {
2
3   var Class = require('class')
4
5   module.exports = Class.create({})
6
7 })
```

```
Programmer.js x JavaScriptProgrammer.js x NodeJSProgrammer.js x init.js x index.html
1 define(function(require, exports, module) {
2
3   module.exports = require('./Programmer').extend({})
4
5 })
```

1. 通过 aralejs/class 来创建类
2. 自动与 seajs 打通，获取关联的模块信息

Result

```
> var lifesinger = new NodeJSProgrammer()
  console.dir(lifesinger)
▼ n
  __proto__: n
    __filename: "NodeJSProgrammer.js"
    __module: b
    constructor: function n(){e.apply(this,arguments),this.constructor===n&&this.initialize&&this.initialize.apply(this,arguments)}
    __proto__: n
      __filename: "JavaScriptProgrammer.js"
      __module: b
      constructor: function n(){e.apply(this,arguments),this.constructor===n&&this.initialize&&this.initialize.apply(this,arguments)}
      __proto__: n
        __filename: "Programmer.js"
        __module: b
        dependencies: Array[1]
        exports: function n(){e.apply(this,arguments),this.constructor===n&&this.initialize&&this.initialize.apply(this,arguments)}
        factory: function (require, exports, module) {
          id: "http://localhost/~lifesinger/lifesinger/lifesinger.github.com/lab/2012/debug/demo2/seajs/src/Programmer.js"
        }
        parent: b
        require: function a(c){c=e(c,b.uri);c=f[c];if(!c)return null;if(c.status===h.COMPILE)return c.exports;
          status: 6
          uri: "http://localhost/~lifesinger/lifesinger/lifesinger.github.com/lab/2012/debug/demo2/seajs/src/Programmer.js"
        }
        __proto__: b
        constructor: function n(){e.apply(this,arguments),this.constructor===n&&this.initialize&&this.initialize.apply(this,arguments)}
        __proto__: r
```

继承信息一目了然

Problem 3

```
require.async('http://cool.com/api/bonus.js', function(data) {  
  
    if (data.status == '1') {  
        alert('中奖啦')  
    }  
    else {  
        alert('很抱歉，你的运气不是很好，再试一次吧')  
    }  
  
})
```

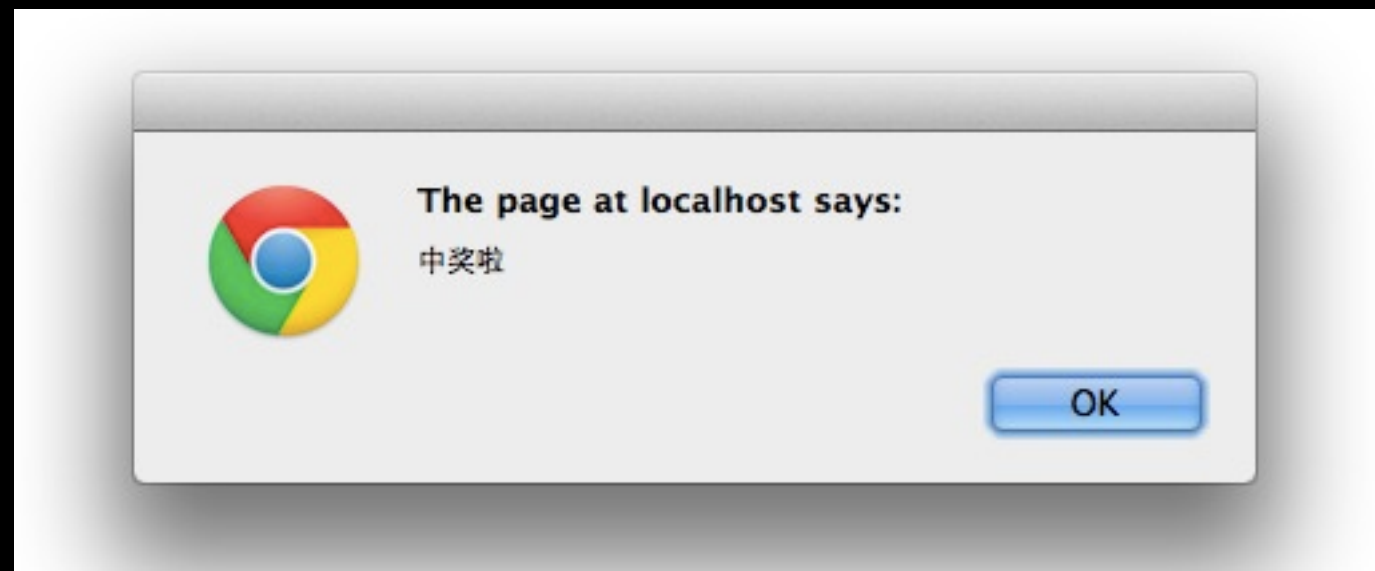
依赖服务接口才能调试

Solution

```
seajs.modify('http://cool.com/api/bonus.js', function(require, exports) {  
  exports.status = 1  
})
```

通过 modify 可以 mock 接口

Result



这个可以用来骗 iPad

Bonus

```
<script src="../../libs/seajs/1.3.0/sea.js"></script>  
<script src="../../libs/seajs/1.3.0/plugin-reload.js"></script>
```

```
$ reload-server
```

不用 F5 刷新了

原理

- SeaJS 是一个模块加载器
- 控制了入口和出口
- So, 随心所欲

RequireJS is good,
but SeaJS is better.

<http://seajs.org/>

Q ? A : Thanks