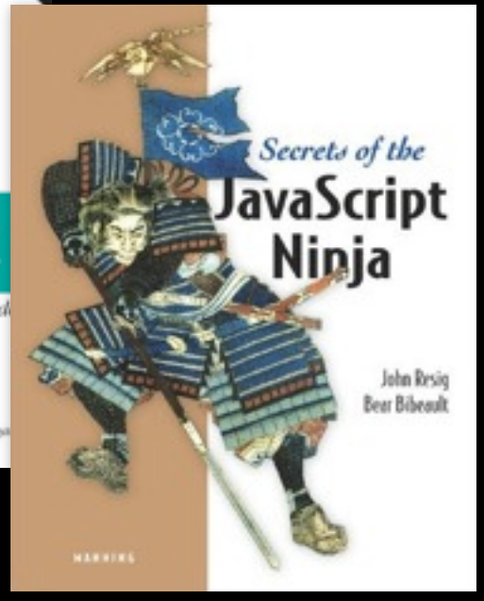
SeaJS 那些事儿

seajs.org

2012

About Me

- 王保平 / lifesinger / 玉伯 / 射雕
- 支付宝 - 前端开发部 - 基础技术组
- KISSY、SeaJS、Arale、布道、翻译



Topics

- SeaJS 是什么
- 核心设计与实现
- 谈谈开源
- 未来规划

1. SeaJS 是什么

SeaJS

is

A **Module** **Loader** for the **Web**

SeaJS 是

适用于 Web 端的模块加载器

以 seajs.org 为例

SeaJS 的应用场景

- SeaJS 是更自然的代码组织方式
- 只要项目的 JS 文件超过 3 个，就适合用
- 文件越多，则越适合
- 误解：SeaJS 不适合小项目

以 Jscex 为例

II. 核心设计与实现

模块系统

什么是系统

- 系统由个体组成
- 个体之间有关连，按照规则协同完成任务

https://github.com/seajs/seajs/issues/240

模块系统的基本问题

- 系统成员：模块是什么?
- 系统通讯：模块之间如何交互?

模块定义规范

CommonJS
Modules / 1.1

AMD

CMD

Node Modules

Intel

...

CommonJS
Modules / 2.0

CMD

- CMD - Common Module Definition
- 尽量与 CommonJS Modules/1.1 以及 Node Modules 的规范保持一致
- 同时考虑 Web 特性

CMD

```
define(function(require, exports, module) {

    var $ = require('jquery')
    var math = require('./math')

    exports.doSomething = ...

})
```

https://github.com/seajs/seajs/issues/242

模块加载器

加载器的基本功能

- 模块定义规范的实现，这是模块系统的基础。
- 模块系统的启动与运行。

https://github.com/seajs/seajs/issues/260

Node 的实现

```
var math = require('./math')
```

Step 1: resolveFilename

Step 2: load

Step 3: compile

Step 4: return module.exports

https://github.com/joyent/node/blob/master/lib/module.js

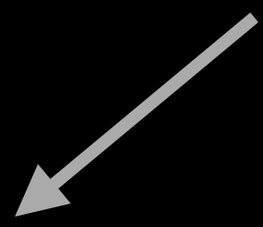
从 Server 到 Web

- node_modules 查找不适合 Web 端
- 文件的同步读取不适合 Web 端
- Web 端的依赖需要提前获取

SeaJS 的实现

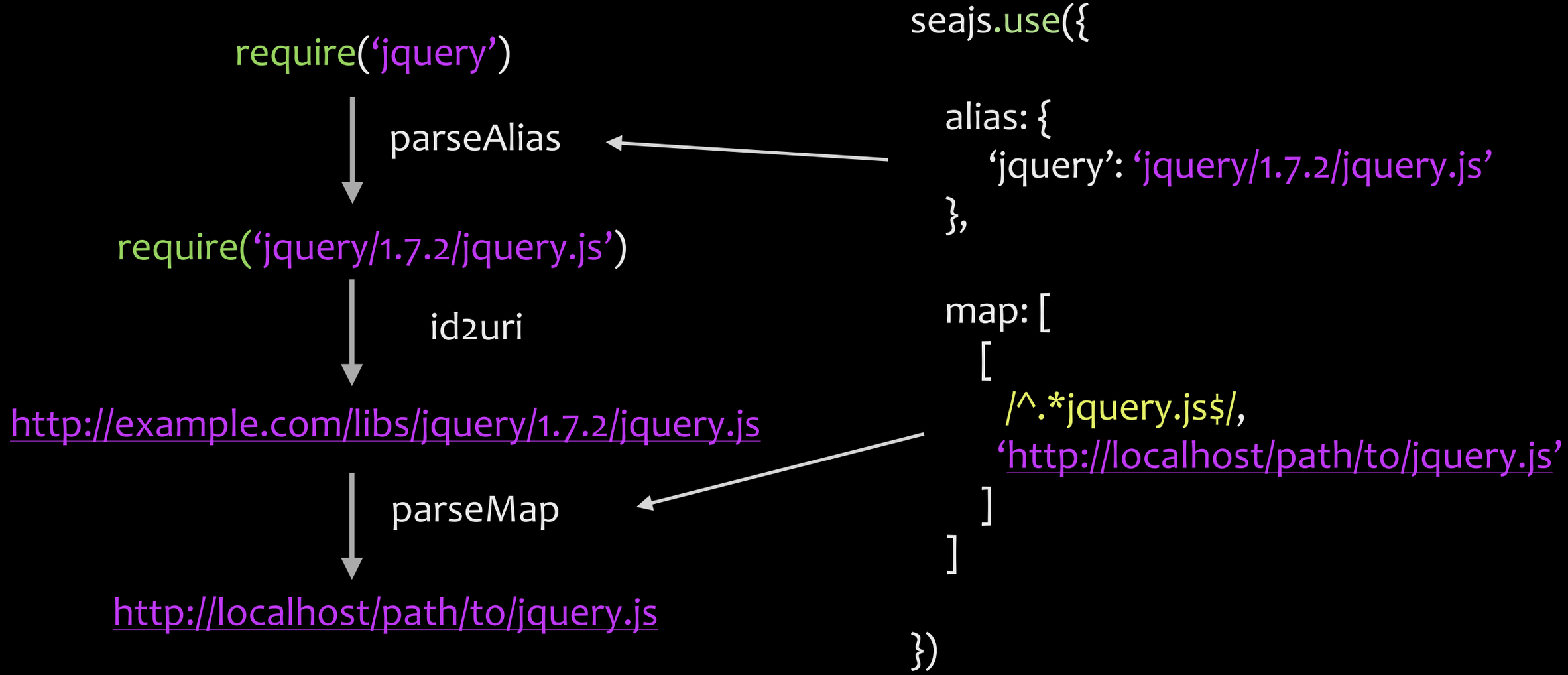
```
/* a.js */
define(function(require, exports, module) {
    var b = require('./b')
    var c = require('./c')
    // ...
})
```

```
/* main.js */
seajs.use('./a')
```

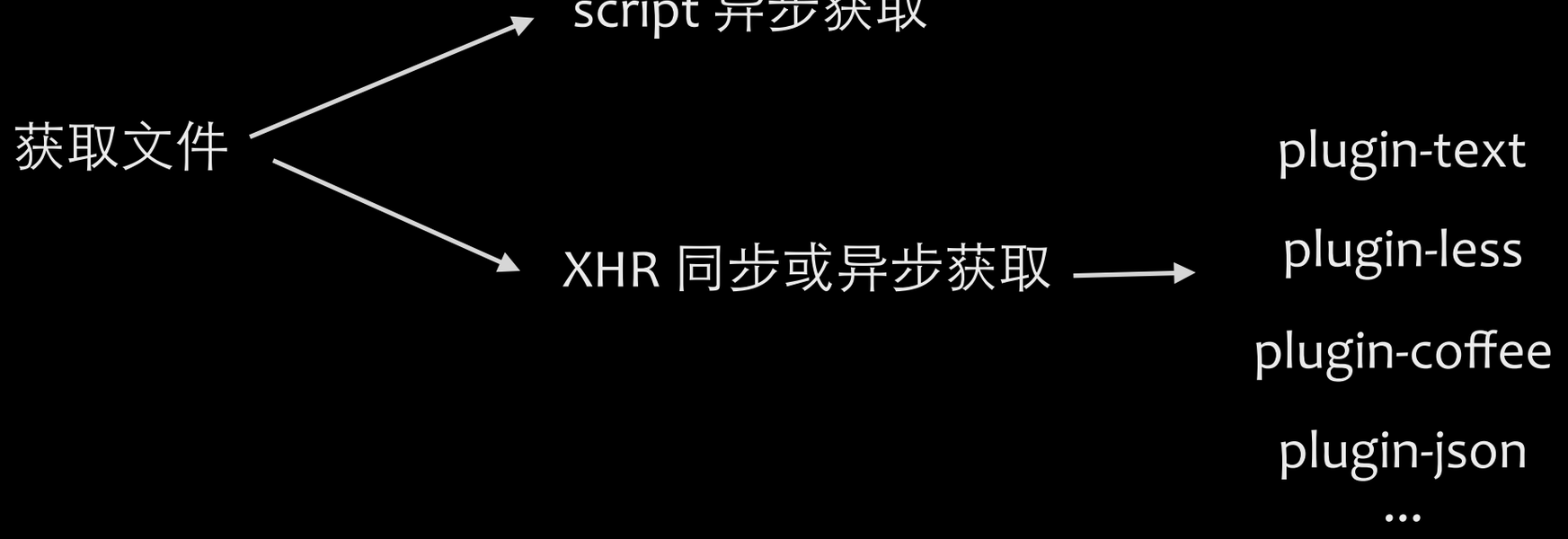


Step 1:　解析 './a'
Step 2.1:　下载 a
Step 2.2:　执行 define，保存 a 的 factory
Step 2.3:　得到依赖 b 和 c
Step 2.4:　加载 b 和 c
Step 3:　执行 a 的 factory，得到 a 的 module.exports

Step 1: 路径解析


require('jquery')
parseAlias
require('jquery/1.7.2/jquery.js')
id2uri
http://example.com/libs/jquery/1.7.2/jquery.js
parseMap
http://localhost/path/to/jquery.js
seajs.use({
alias: {
'jquery': 'jquery/1.7.2/jquery.js'
},
map: [
[
/^.*jquery.js$/,
'http://localhost/path/to/jquery.js'
]
]
})

Step 2: 模块加载


script 异步获取
获取文件
plugin-text
XHR 同步或异步获取
plugin-less
plugin-coffee
plugin-json
...

如何得到依赖?

factory.toString() + 正则匹配

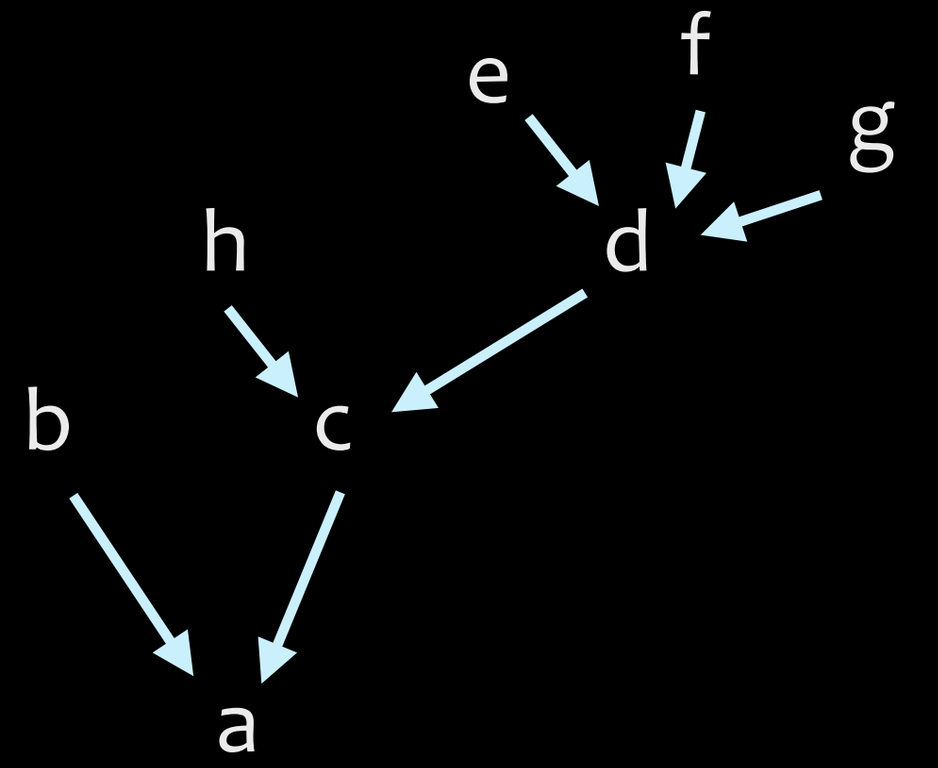
https://github.com/seajs/seajs/blob/master/src/util-deps.js

require('./xxx')

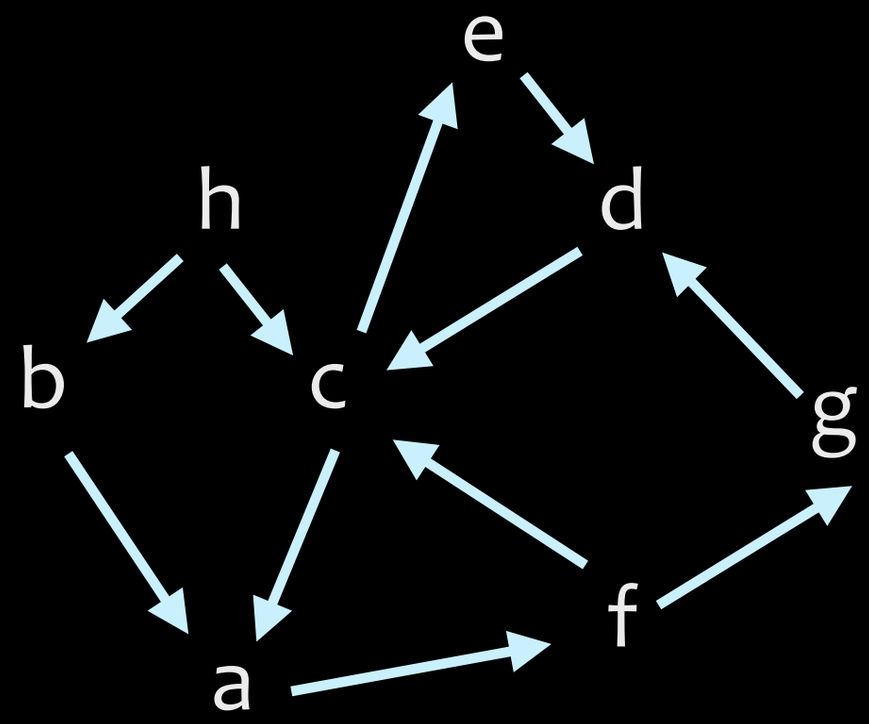
Rule 1: factory 第一个参数的命名必须是 require
Rule 2: require 函数只能接收字符串值
Rule 3: 不要覆盖 require

http://seajs.org/docs/zh-cn/rules.html

依赖的回调树



循环依赖



加载时的循环等待

isCircularWaiting(module, uri)

https://github.com/seajs/seajs/blob/master/src/module.js

编译时的循环等待

```
if (module.status === STATUS.COMPILING) {
    return module.exports
}
```

https://github.com/seajs/seajs/blob/master/src/module.js

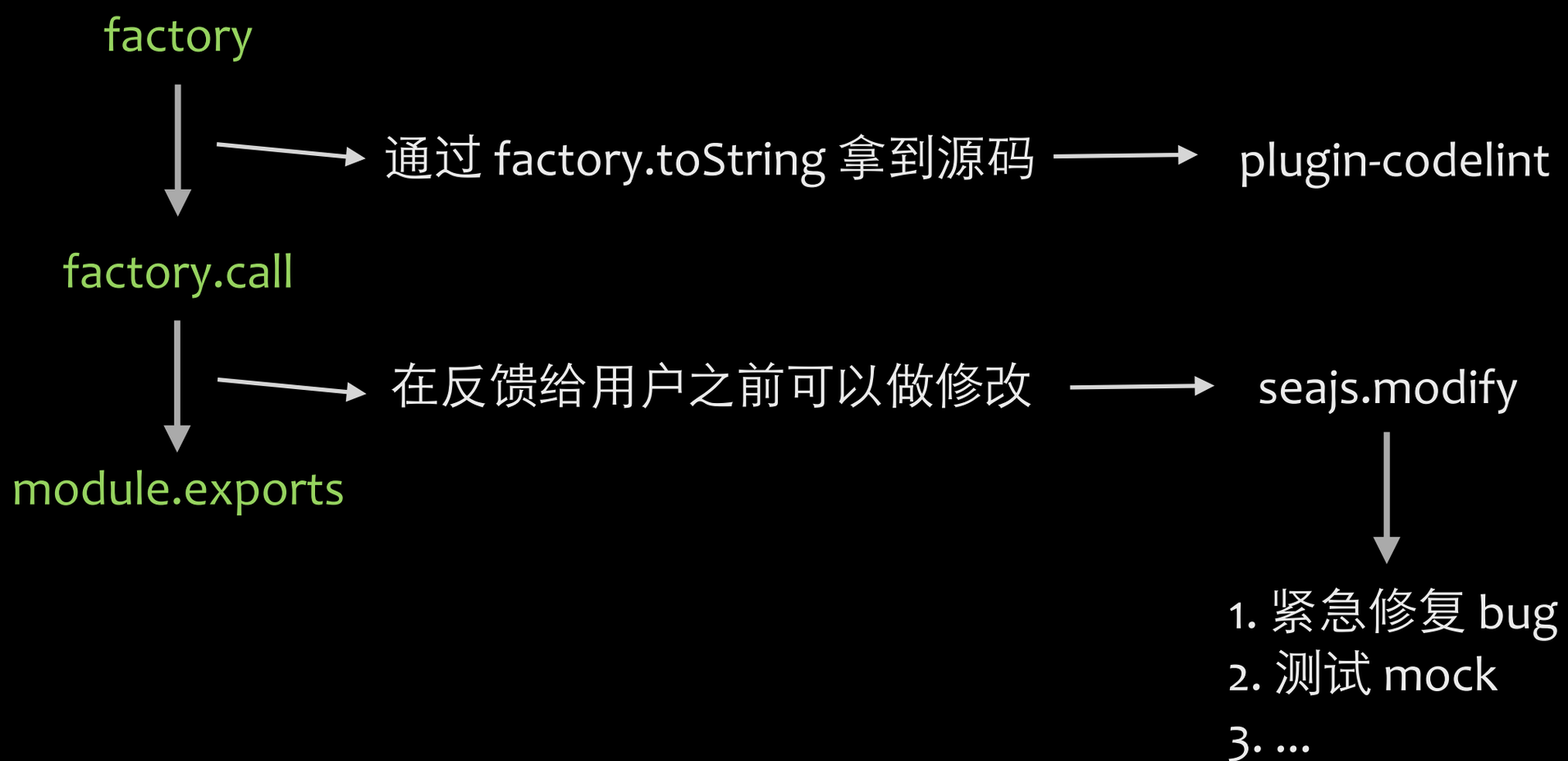
Step 3: 代码编译

```
/* a.js */
define(function(require, exports, module) {
    var b = require('./b')
    var c = require('./c')
    // ...
})
```

```
module.require = require
module.exports = {}

module.factory.call(
        window,
        module.require,
        module.exports,
        module
)
```

编译前后


factory
通过 factory.toString 拿到源码
plugin-codelint
factory.call
在反馈给用户之前可以做修改
seajs.modify
module.exports
1. 紧急修复 bug
2. 测试 mock
3. ...

实现小结

- 路径解析： id -- uri
- 依赖加载： toString / onload / ...
- 代码编译： factory.call

SeaJS 的可靠性

SeaJS 的基本假设

A --- 表示 a.js 执行时的时间
a --- 表示 a.js 的 onload / onerror 时的时间

开发时，SeaJS 要求：A 与 a 紧相邻
上线后，SeaJS 要求：A < a

http://seajs.org/test/research/onload-order/test.html

https://github.com/seajs/seajs/issues/130

疯狂的测试用例

http://seajs.org/test/runner.html

PC、Mobile

理论上是个浏览器就应该可以跑

已有哪些公司在用


支付宝
淘宝网
一淘 etao.com
来往
朋友网 pengyou.com
百姓网
发现啦!
好知
网易微博 t.163.com
Clicki.cn
雪球
医道网 www.eDoctor.cn
Sodao搜道网 www.sodao.com
DNSPOD

More

- seajs.log / seajs.cache / seajs.find / seajs.modify
- plugin-text / plugin-json / plugin-combo / plugin-coffee / plugin-less / plugin-livereload / plugin-codelint / ...
- seajs.pluginSDK

III. 谈谈开源

开源的目的

- 把好的东西分享出来
- 让好的东西变得更好
- 其他一切皆是浮云

开源中最重要的

- 一个优秀、靠谱的想法
- 疯狂而持久的坚持

开源项目起步时，梦想都很丰满，但现实都很骨感。很多人等不到后天的太阳，经常离开于明天的晚上。

IV. 未来规划

SeaJS v1.2.0

https://github.com/seajs/seajs/issues/190

SeaJS v1.2 will release SOON !

SeaJS v1.3.0

https://github.com/seajs/seajs/issues/225

SPM 1.0

will release at Jul 30

No 跳票 again!

不仅仅是模块加载器

- JavaScript 模块生态圈之梦
- Arale 的尝试
- SPM 仓库

Questions?

与 RequireJS 对比

- API 设计上比 RequireJS 更优秀
- 实现上比 RequireJS 更优秀
- 理念上比 RequireJS 更优秀
- 更懂中国人

向 CommonJS / NodeJS /
UnCommonJS / FlyScript /
RequireJS / ... 致敬！！！

:)

seajs.org