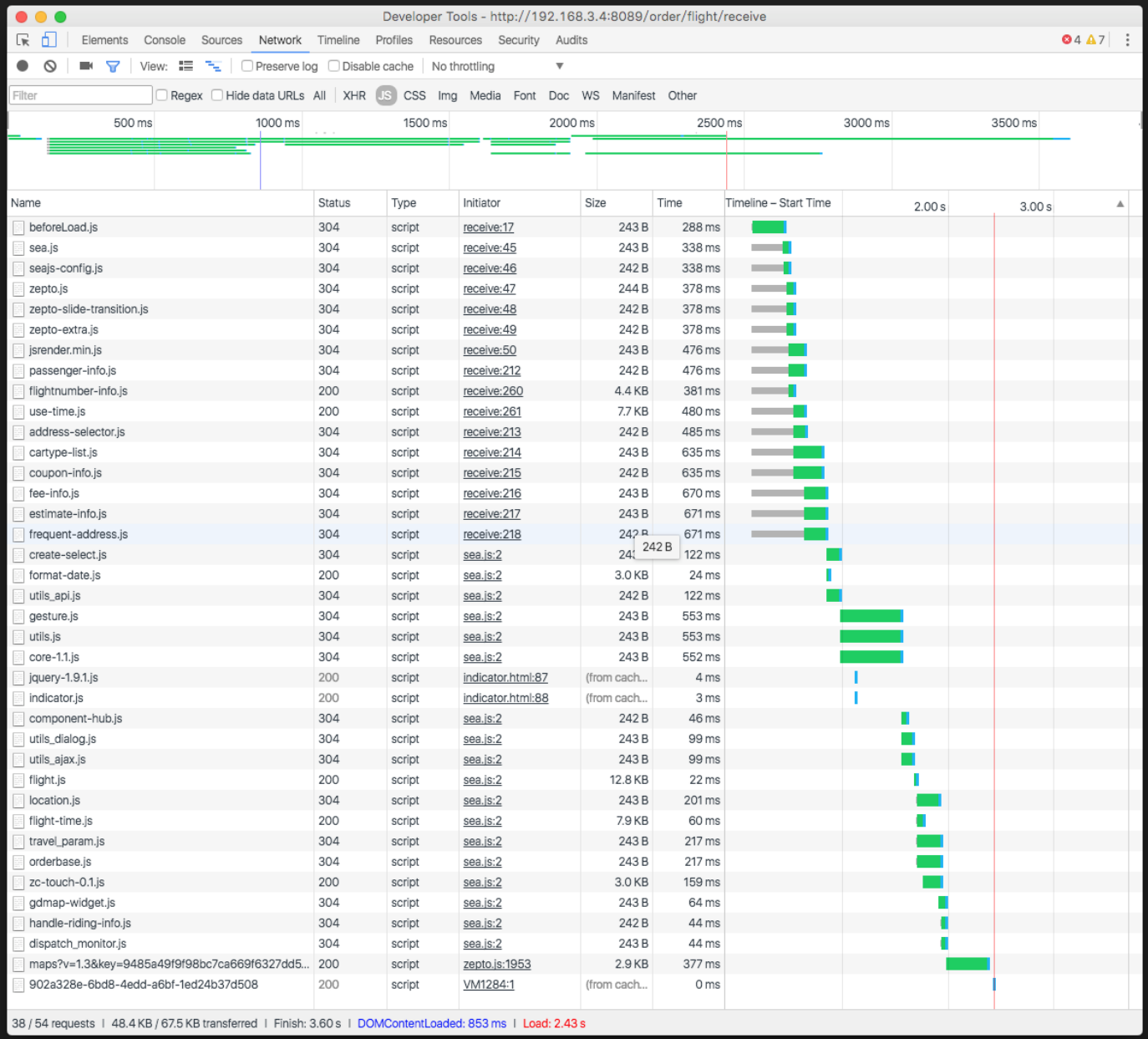
Tiler

Pack up static resources.

Why

我们面临的主要问题：

- Module/Loader 采用 SeaJS，串行加载效率低
- 模块化、组件化深入推进，请求量较多

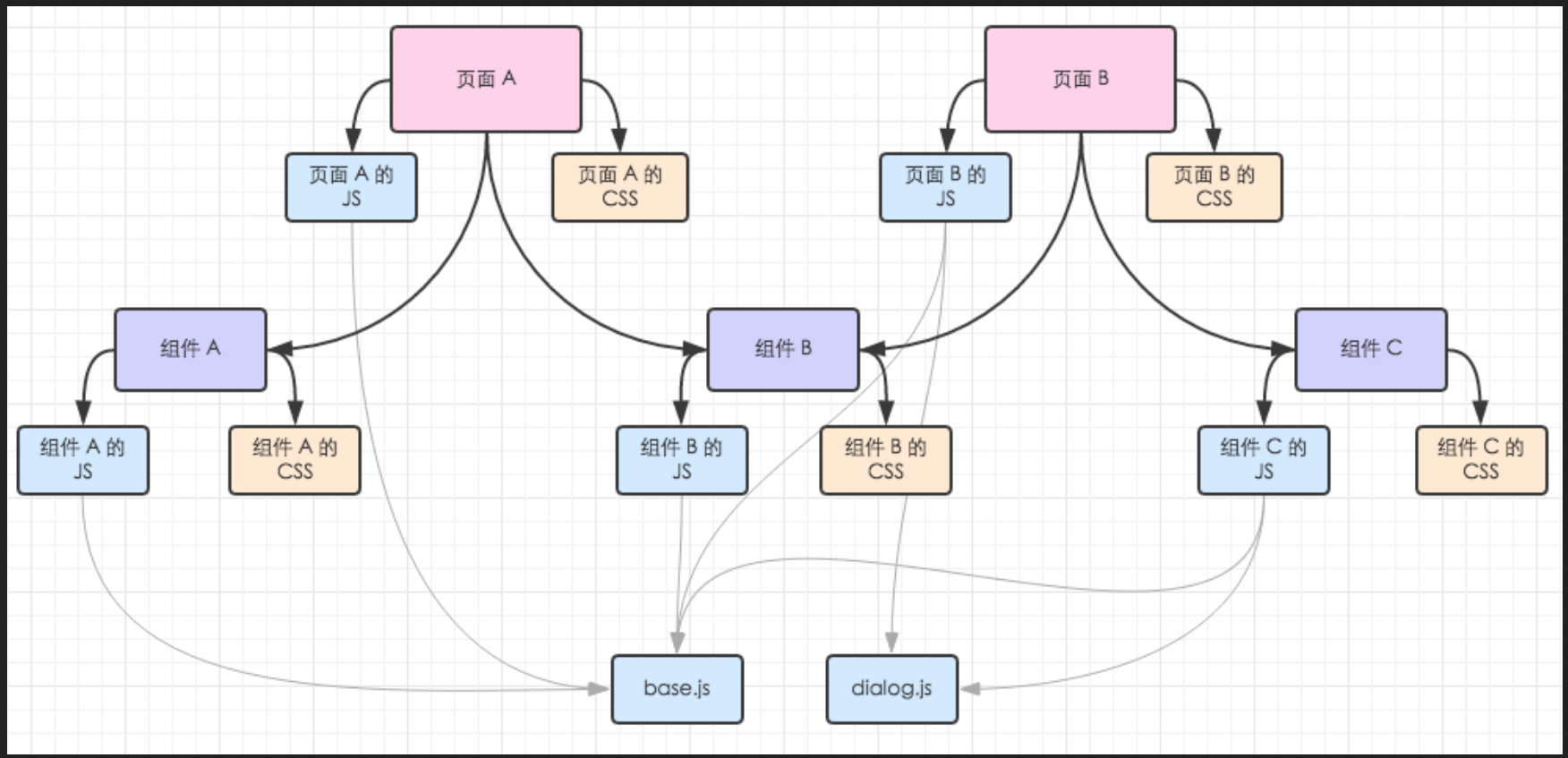

Developer Tools - http://192.168.3.4:8089/order/flight/receive
Elements Console Sources Network Timeline Profiles Resources Security Audits
View: Preserve log Disable cache No throttling
Filter Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms
Name Status Type Initiator Size Time Timeline – Start Time 2.00 s 3.00 s
beforeLoad.js 304 script receive:17 243 B 288 ms
sea.js 304 script receive:45 243 B 338 ms
seajs-config.js 304 script receive:46 242 B 338 ms
zepto.js 304 script receive:47 244 B 378 ms
zepto-slide-transition.js 304 script receive:48 242 B 378 ms
zepto-extra.js 304 script receive:49 242 B 378 ms
jsrender.min.js 304 script receive:50 243 B 476 ms
passenger-info.js 304 script receive:212 242 B 476 ms
flightnumber-info.js 200 script receive:260 4.4 KB 381 ms
use-time.js 200 script receive:261 7.7 KB 480 ms
address-selector.js 304 script receive:213 242 B 485 ms
cartype-list.js 304 script receive:214 243 B 635 ms
coupon-info.js 304 script receive:215 242 B 635 ms
fee-info.js 304 script receive:216 243 B 670 ms
estimate-info.js 304 script receive:217 243 B 671 ms
frequent-address.js 304 script receive:218 242 B 671 ms
242 B
create-select.js 304 script sea.js:2 122 ms
format-date.js 200 script sea.js:2 3.0 KB 24 ms
utils_api.js 304 script sea.js:2 242 B 122 ms
gesture.js 304 script sea.js:2 243 B 553 ms
utils.js 304 script sea.js:2 243 B 553 ms
core-1.1.js 304 script sea.js:2 243 B 552 ms
jquery-1.9.1.js 200 script indicator.html:87 (from cach... 4 ms
indicator.js 200 script indicator.html:88 (from cach... 3 ms
component-hub.js 304 script sea.js:2 242 B 46 ms
utils_dialog.js 304 script sea.js:2 243 B 99 ms
utils_ajax.js 304 script sea.js:2 243 B 99 ms
flight.js 200 script sea.js:2 12.8 KB 22 ms
location.js 304 script sea.js:2 243 B 201 ms
flight-time.js 200 script sea.js:2 7.9 KB 60 ms
travel_param.js 304 script sea.js:2 243 B 217 ms
orderbase.js 304 script sea.js:2 243 B 217 ms
zc-touch-0.1.js 200 script sea.js:2 3.0 KB 159 ms
gdmap-widget.js 304 script sea.js:2 243 B 64 ms
handle-riding-info.js 304 script sea.js:2 242 B 44 ms
dispatch_monitor.js 304 script sea.js:2 243 B 44 ms
maps?v=1.3&key=9485a49f9f98bc7ca669f6327dd5... 200 script zepto.js:1953 2.9 KB 377 ms
902a328e-6bd8-4edd-a6bf-1ed24b37d508 200 script VM1284:1 (from cach... 0 ms
38 / 54 requests | 48.4 KB / 67.5 KB transferred | Finish: 3.60 s | DOMContentLoaded: 853 ms | Load: 2.43 s

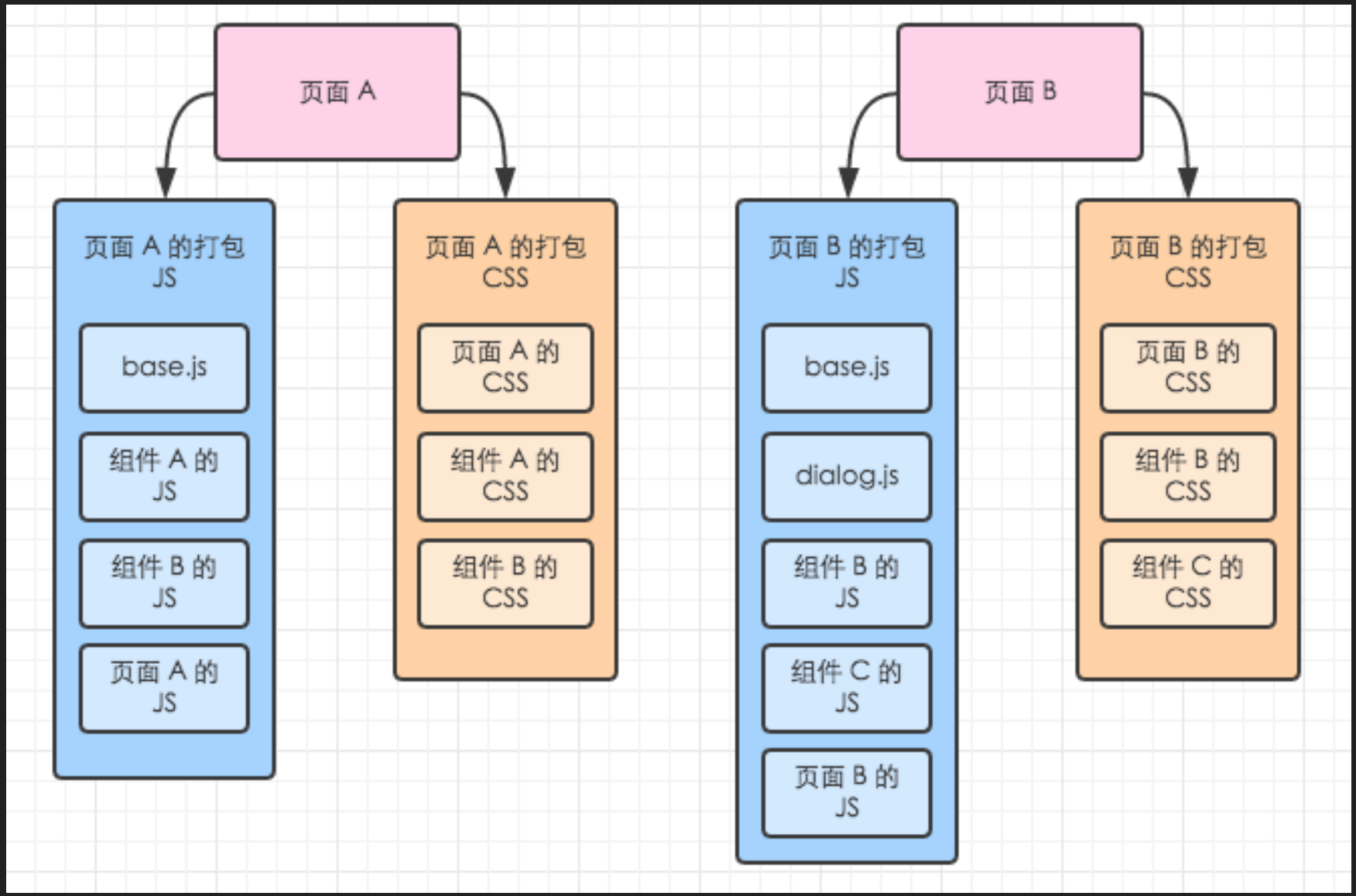
What

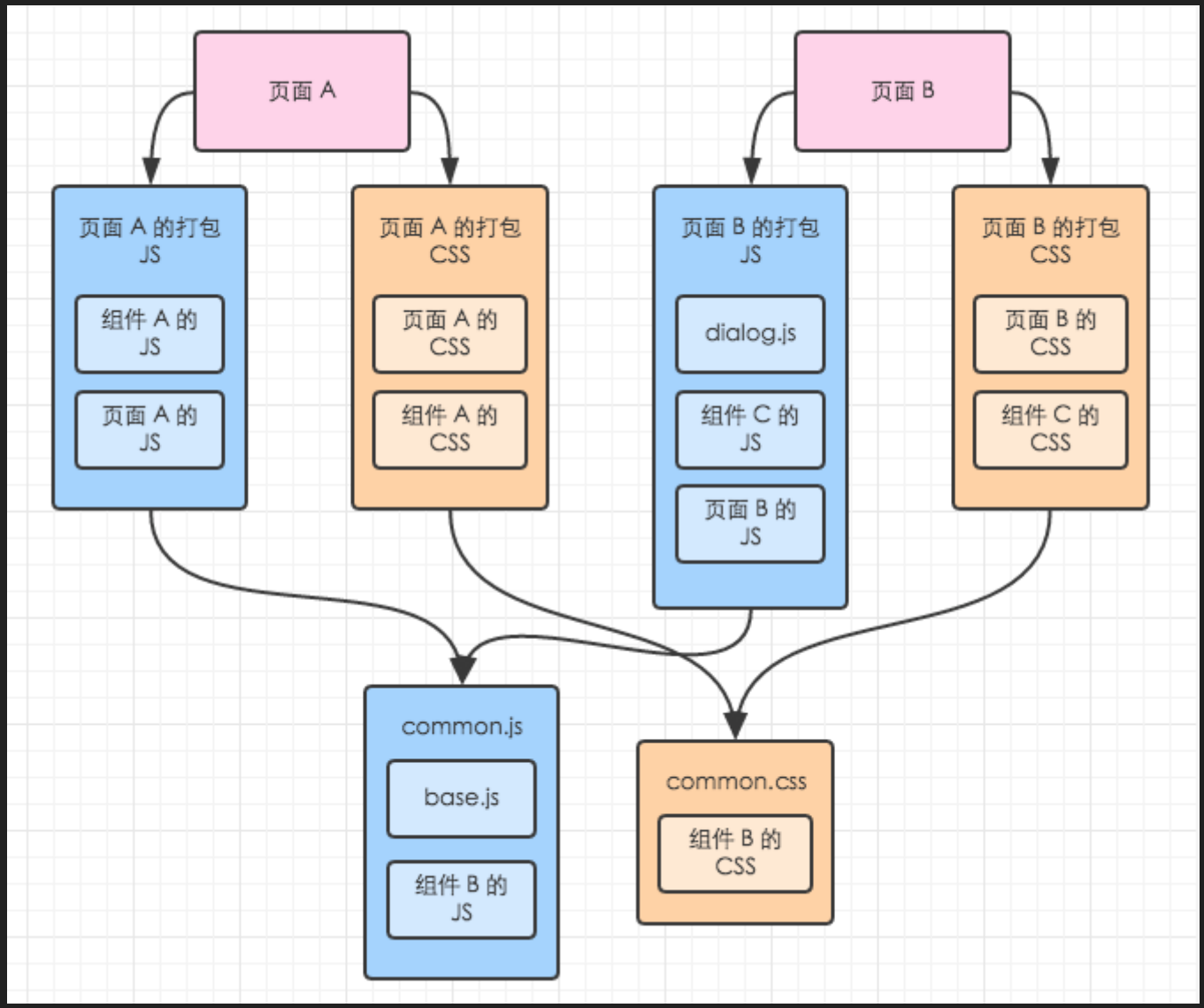
具体需求：

- 支持按照页面进行 JS、CSS 文件的打包
- 支持公共包的配置，优化加载速度
- 支持页面组件打包
- 支持 CSS 外联资源的处理
- 与 Gulp 较好集成

一个栗子


页面 A
页面 A 的 JS
页面 A 的 CSS
页面 B
页面 B 的 JS
页面 B 的 CSS
组件 A
组件 A 的 JS
组件 A 的 CSS
组件 B
组件 B 的 JS
组件 B 的 CSS
组件 C
组件 C 的 JS
组件 C 的 CSS
base.js
dialog.js


页面 A
页面 A 的打包 JS
base.js
组件 A 的 JS
组件 B 的 JS
页面 A 的 JS
页面 A 的打包 CSS
页面 A 的 CSS
组件 A 的 CSS
组件 B 的 CSS
页面 B
页面 B 的打包 JS
base.js
dialog.js
组件 B 的 JS
组件 C 的 JS
页面 B 的 JS
页面 B 的打包 CSS
页面 B 的 CSS
组件 B 的 CSS
组件 C 的 CSS


页面 A
页面 B
页面 A 的打包 JS
组件 A 的 JS
页面 A 的 JS
页面 A 的打包 CSS
页面 A 的 CSS
组件 A 的 CSS
页面 B 的打包 JS
dialog.js
组件 C 的 JS
页面 B 的 JS
页面 B 的打包 CSS
页面 B 的 CSS
组件 C 的 CSS
common.js
base.js
组件 B 的 JS
common.css
组件 B 的 CSS

How

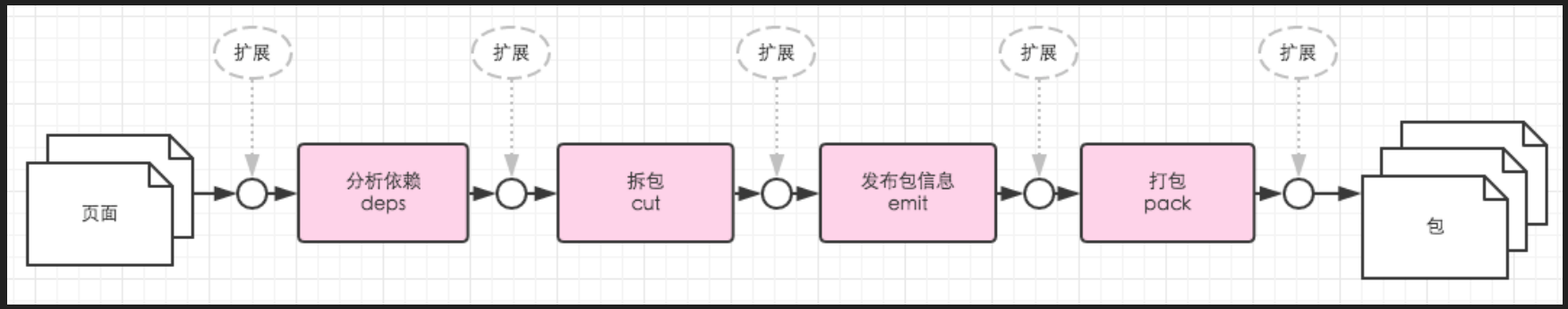
- 借鉴 Browserify 思想，基于 Stream
- 将复杂的逻辑拆分为多个串行的 Transform
- 处理对象采用和 Gulp 兼容的 Vinyl

Stream


```
var gulp = require('gulp');
var coffee = require('gulp-coffee');
var uglify = require('gulp-uglify');

gulp.task('js', function(){
    return gulp.src('./js/src/*.coffee')
        .pipe(coffee())
        .pipe(uglify())
        .pipe(gulp.dest('./js/'));
});
gulp.task('watch', function(){
    gulp.watch('./js/src/*.coffee', ['js']);
});
```

主要流程



核心代码

```
const pipeline = splicer.obj([
  'record', [this._recorder()],
  'deps', [this._deps],
  'sort', [this._sort()],
  'cut', [this._cut()],
  'emit', [this._emitDeps()],
  'pack', [this._pack()],
  'append', [this._append()],
  'write', [this._write()],
]);
```

Deps

分析页面依赖

```
{{StyleLink 'order.css'}}

{{Component 'address-selector'}}

<script>
  seajs.use('jsBasePath/order/immediate');
</script>
```

Deps

分析页面依赖

```
{{StyleLink 'order.css'}}

{{Component 'address-selector'}}

<script>
  seajs.use('jsBasePath/order/immediate');
</script>
```

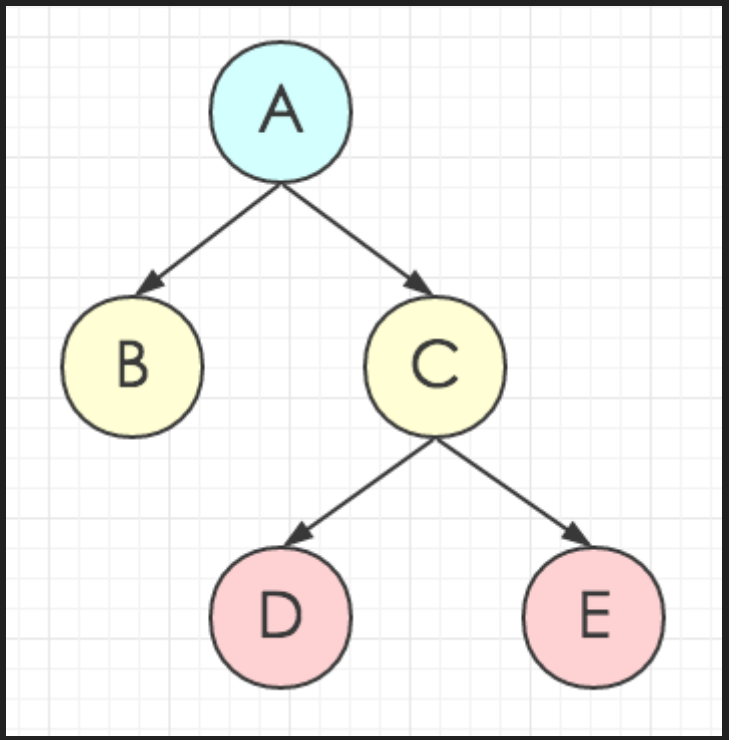
```
# JS
js/order/immediate-entry.js
├──component/address-selector/address-selector.js
└──js/order/immediate.js

# CSS
css/order/immediate-entry.css
└──css/order.css
```

配置的公共层

```
{
  script: [
    {
      path: getClientPath('js/common/seed.js'),
      includes: resource.COMMON_SCRIPTS.map(script => {
        return getClientPath('js/' + script);
      }),
    },
    getClientPath('js/common/core-1.1.js'),
  ],
  style: [
    {
      path: getClientPath('css/seed.css'),
      includes: resource.COMMON_STYLES.map(style => {
        return getClientPath('css/' + style);
      }),
    },
  ],
};
```

递归分析依赖

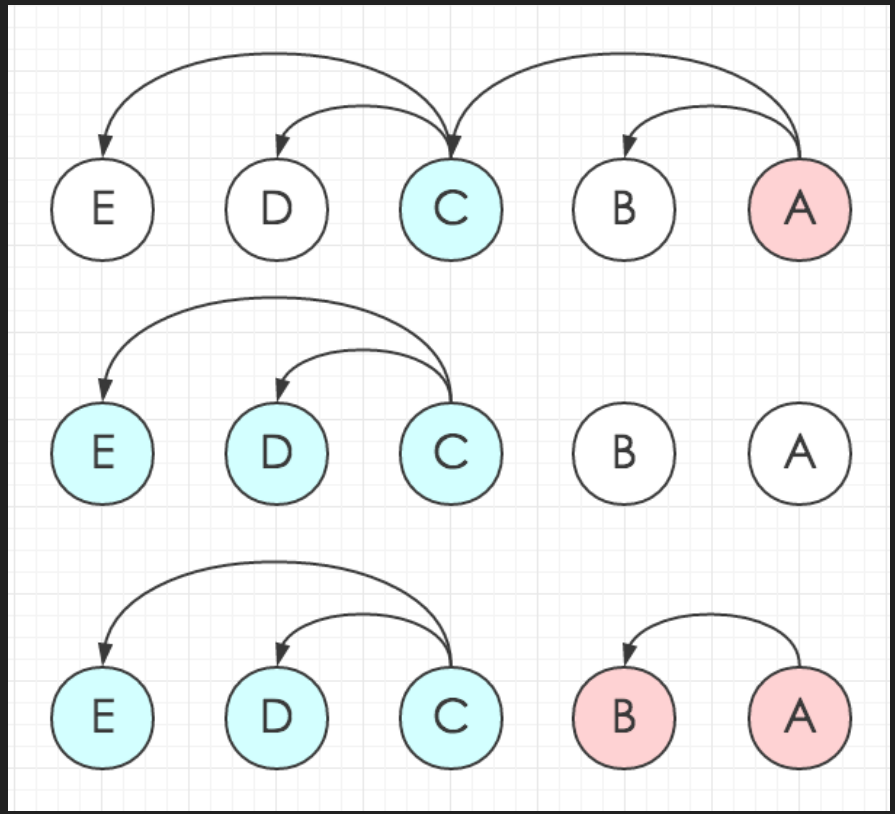


小结

- 输入页面
- 分析页面资源、页面组件资源
- 递归分析依赖关系
- 根据依赖关系进行拓扑排序
- 输出页面依赖的所有 JS、CSS

Cut

资源分组，为接下来的打包提供依据



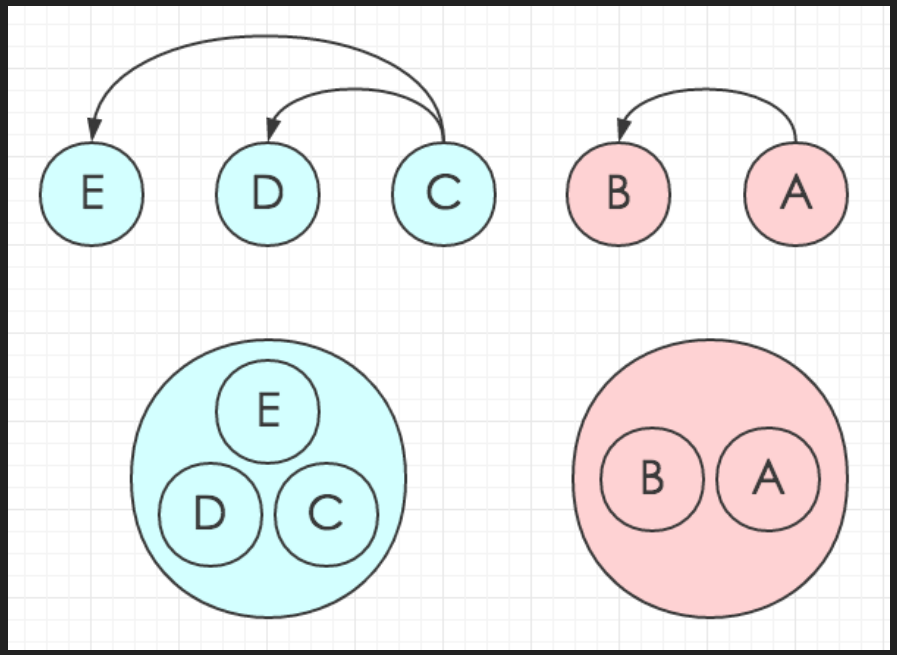
Emit

发送依赖和分组信息给外部，方便进行数据记录

```
const self = this;
return new Transform({
  objectMode: true,
  transform(row, enc, next) {
    self.emit('dep', row);
    this.push(row);
    next();
  },
});
```

Pack

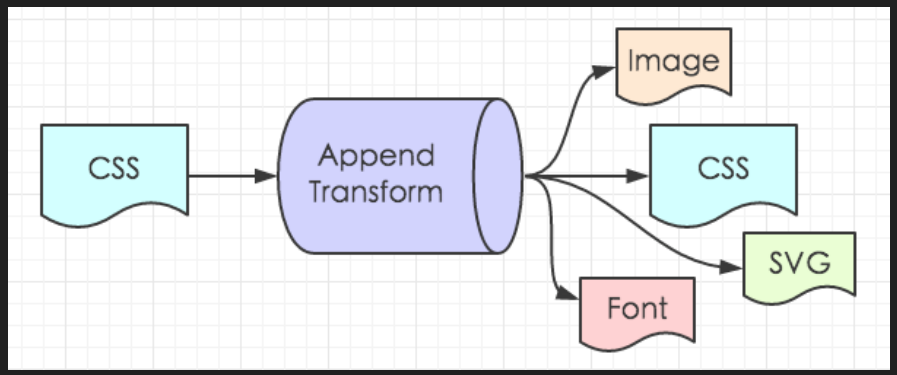
根据分组信息进行打包



Append

添加 CSS 外联资源

```
.travel-info .true-show span:before {
    position: absolute;
    content: '';
    width: 22px;
    height: 20px;
    background: url('../img/usericon.png') -80px 0 no-repeat;
}
```



与 Gulp 整合

```
var stream = tiler.bundle()
  .pipe(addsrc(standaloneFiles, { base: clientDir }))
  .pipe(dedupe())
  .pipe(hold())
  .pipe(gulp.dest(staticDir))
  // stamp version
  .pipe(imagemin())
  .pipe(stampImageVersion)
  .pipe(hold())
  .pipe(rewriteStyleUrl)
  .pipe(stampOtherVersion)
  .pipe(gulp.dest(staticDir))
  // compress
  .pipe(stampMin)
  .pipe(cssmin())
  .pipe(filterScript)
  .pipe(uglify())
  .pipe(filterScript.restore)
  .pipe(gulp.dest(staticDir));
```

TODO

- 支持按内容去重
- 支持更加灵活、自动的公共层打包
- 支持 CommonJS 规范的模块
- 支持更高性能要求的即时打包

参考

- browserify
- gulp vinyl
- depsify

Welcome Contribute!

@ucar/tiler

Thanks