

webpack支撑大规模应用开 发最佳实践

个人简介

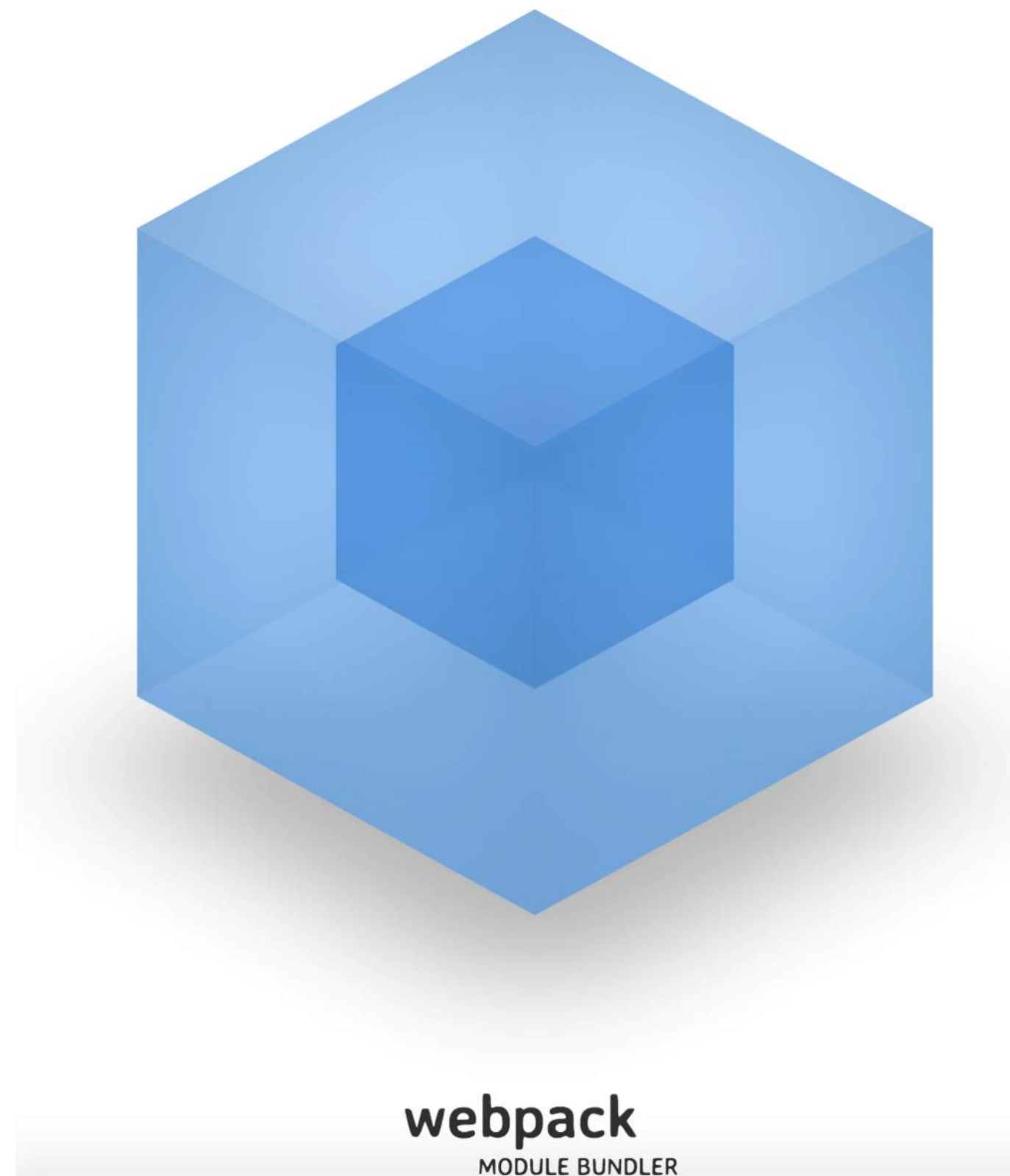
- 个人信息

- 李成熙, AlloyTeam
- 2014年底加入腾讯
- 群活动、家校群、互动直播
- Github: <https://github.com/lcxfs1991>

- 工作方向

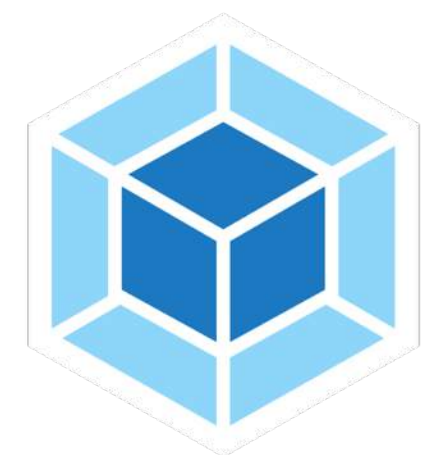
- 前端工程化、自动化
- 性能优化
- Hybrid, React Native

From webpack1 to webpack2



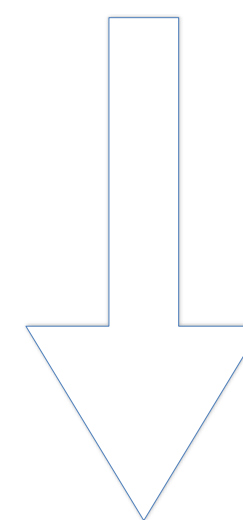
webpack.github.io

From webpack1 to webpack2



webpack

webpack.io



webpack.js.org

Gulp vs Grunt vs Browserify vs webpack

	Gulp	Grunt	Browserify	webpack
Github Stars	23668	11105	10350	19684
Weekly Download	65W+	46W+	50W+	66W+

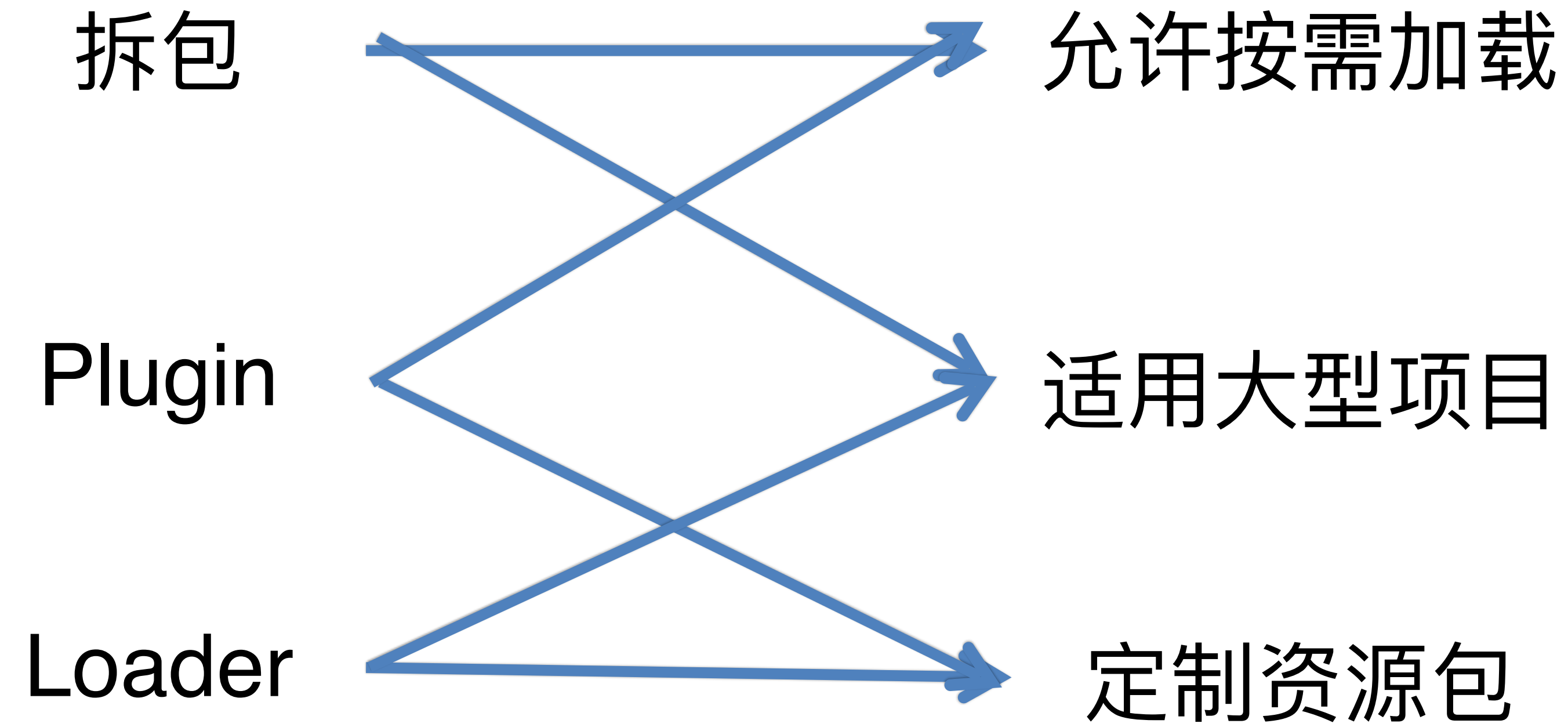
为什么要用webpack

why webpack

为什么要用webpack

1. 与react一类模块化开发的框架搭配着用比较好
2. 属于配置型的构建工具，比较容易上手，160行代码可大致实现gulp300多行才能实现的功能

为什么要用webpack



webpack核心概念

module & chunk & positioning

Webpack核心概念 -- Module

- Module 模块

1. js

2. css, images, fonts...

```
// 函数写法
```

```
var mod1 = function() {
```

```
};
```

```
// 对象写法
```

```
var mod2 = {
```

```
  init: function() {
```

```
  }
```

```
};
```

```
// 立即执行函数
```

```
var mod3 = (function(win) {
```

```
})(window);
```

Webpack核心概念 -- Chunk

- Chunk 块
 1. Many small modules combined
 2. Entry Chunk, Output Chunk, Common Chunk

Webpack核心概念 -- 定位

- 模块加载器(module loader)

1. requirejs

2. sea.js

特点:

1. 浏览器中加载入口及依赖

2. 执行入口及依赖代码

Webpack核心概念 -- 定位

- 模块打包工具(module bundler)

1. browserify

2. webpack

特点：

1. 浏览器中加载入口及依赖

2. 执行入口及依赖代码

3. 分析及合成模块系统

4. 生成文件束 (bundle/chunk)

不仅仅是bundler

more than a module bundler

现在被当成构建工具使用

Stronger as the community flourish

简略回顾Webpack配置

```
var devConfig = {
  entry: {
    common: ['jquery', 'bootstrap', 'net', 'spin'],
    index: ["webpack-dev-server/client?http://localhost:9000",
      "webpack/hot/only-dev-server",
      path.join(config.path.src, "/page/index/main.js")],
  },
  output: {
    publicPath: config.defaultPath,
    path: path.join(config.path.dist),
    filename: "js/[name]" + config.hash + ".js"
  },
  module: {
    noParse: [path.join(nodeModulesPath, '/react/dist/react.min')],
    loaders: [
      {
        test: /\.js?$/,
        loaders: ['react-hot', 'babel?presets[]=es2015&presets[]=react'],
        exclude: /node_modules/,
      },
      {
        test: /\.css$/,
        loader: ExtractTextPlugin.extract("style-loader", "css-loader"),
        include: path.resolve(config.path.src)
      },
    ],
  },
  resolve: {
    extensions: ['', '.js', '.jsx', '.es6', '.css', '.scss', '.png', '.jpg', '.jpeg'],
    alias: {
      'react-dom': path.join(nodeModulesPath, '/react-dom/dist/react-dom'),
      'redux': path.join(nodeModulesPath, '/redux/dist/redux'),
    }
  },
  plugins: [
    new webpack.HotModuleReplacementPlugin(), // 如果server.js好像需要开启?
    new webpack.optimize.OccurenceOrderPlugin(),
    new HtmlWebpackPlugin({
      inject: true,
      excludeChunks: ['test'],
      template: 'src/index.html',
      favicon: 'src/img/favicon.ico'
    }),
    new ExtractTextPlugin("./css/[name]" + config.hash + ".css"),
    new webpack.NoErrorsPlugin()
  ],
};
```

1. entry: 源文件

2. output: 生成文件

3. loader: 编译文件

4. plugins: 插件，比loader更强大，能使用更多webpack的api

5. resolve: 文件路径的指向

webpack基础构建

a basic building tool

Webpack基础构建

1. 架构选型
2. 编译资源 (js, css, image等)
3. 生成文件

Webpack基础构建 -- 架构选型

Gulp为核心
Webpack仅作为
js打包工具



Webpack为核心
Gulp为辅助
Script协助流程



Webpack基础构建 -- 架构选型

- 不建议使用webpack-stream，一处修改，全量更新

gulp.watch

增量更新
13秒

```
[18:09:45] Version: webpack 1.12.0
      Asset      Size  Chunks  Chunk Names
      js/user.js  121 kB    15  [emitted] user
js/question_add.js  132 kB     0  [emitted] question_add
      js/ask_add.js  137 kB     2  [emitted] ask_add
      js/answer_add.js  133 kB     3  [emitted] answer_add
js/mentor_apply.js  142 kB     4  [emitted] mentor_apply
js/mentor_book.js  146 kB     5  [emitted] mentor_book
      js/mentor.js  138 kB     6  [emitted] mentor
js/ask_detail.js  145 kB     7  [emitted] ask_detail
      js/ask.js  135 kB     8  [emitted] ask
js/withdraw.js  126 kB     9  [emitted] withdraw
js/user_resetpassword.js  123 kB    10  [emitted] user_resetpassword
js/user_profile.js  125 kB    11  [emitted] user_profile
js/user_pic.js  125 kB    12  [emitted] user_pic
js/user_msg.js  120 kB    13  [emitted] user_msg
js/user_invite.js  123 kB    14  [emitted] user_invite
      js/index.js  139 kB     1  [emitted] index
js/relation.js  125 kB    16  [emitted] relation
js/payment.js  122 kB    17  [emitted] payment
js/mentor_student.js  127 kB    18  [emitted] mentor_student
js/mentor_appoint_detail.js  128 kB    19  [emitted] mentor_appoint_detail
js/mentor_appoint.js  124 kB    20  [emitted] mentor_appoint
js/type_manage.js  123 kB    21  [emitted] type_manage
js/money_withdraw.js  122 kB    22  [emitted] money_withdraw
js/mentor_pass.js  122 kB    23  [emitted] mentor_pass
js/mentor_cancel.js  121 kB    24  [emitted] mentor_cancel
js/industry_manage.js  123 kB    25  [emitted] industry_manage
      js/cms.js  117 kB    26  [emitted] cms
js/forget.js  105 kB    27  [emitted] forget
js/result.js  96.2 kB    28  [emitted] result
js/common.js  12.3 kB    29  [emitted] common
[18:09:45] Finished 'webpack' after 13 s
```

Webpack基础构建 – 编译资源

- 将源文件转成浏览器能识别的
 1. es6 => es5
 2. image => base64
 3. sass/less => css
 4. etc ...

Loader & Plugin

常用Loaders介绍

1. 处理样式，转成css，如：less-loader, sass-loader
2. 图片处理，如：url-loader, file-loader。两个都必须用上。否则超过大小限制的图片无法生成到目标文件夹中
3. 处理js，将es6或更高级的代码转成es5的代码。如：
babel-loader, babel-preset-es2015, babel-preset-react, babel, babel-preset-loose
4. 读取html生成模板字符串，如html-loader
5. 将js模块暴露到全局，如：expose-loader
6. 将第三方库注入到模块中，如import-loader
7. [List of loaders](#)

常用Plugins介绍

1. 生成html文件及注入资源, HtmlWebpackPlugin
2. 将css成生文件, 而非内联, ExtractTextPlugin
3. 代码丑化, UglifyJsPlugin, 开发过程中不建议打开
4. [List of Plugins](#)

Webpack基础构建 – 生成文件

1. entry => 入口文件

- Single entry

```
const config = {  
  entry: './path/to/my/entry/file.js'  
};  
  
module.exports = config;
```


Webpack基础构建 – 生成文件

1. entry => 入口文件

- Object Syntax

```
const config = {  
  entry: {  
    app: './src/app.js',  
    vendors: './src/vendors.js'  
  }  
};
```

Webpack基础构建 – 生成文件

1. entry => 入口文件

- Object Syntax

```
const config = {  
  entry: {  
    app1: './src/app.js',  
    app2: ['./src/a.js', 'src/b.js'],  
    vendors: './src/vendors.js'  
  }  
};
```

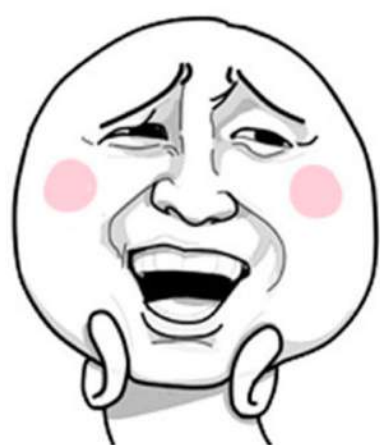
Webpack基础构建 – 生成文件

2. output => 输出文件路径配置等

```
output: {  
  publicPath: "s1.url.cn",  
  path: path.join("/pub/"),  
  filename: "js/[name]-[chunkhash:6].js",  
  chunkFilename: "js/chunk/[name]-[chunkhash:6].js",  
},|
```




老板: React挺好的, 我们尝试用到项目中



我: 没问题



老板: 忘了说, 项目有10个页面



我: 。 。 。 。 。

全量
编译时间
26s

Version: webpack 1.13.2

Time: 26352ms

Asset	Size
./css/spa7-f1177c.css	19.4 kB
js/spa9-615a20.js	1.18 MB
js/spa7-615a20.js	1.18 MB
js/spa6-615a20.js	1.18 MB
js/spa5-615a20.js	1.18 MB
js/spa4-615a20.js	1.18 MB
js/spa3-615a20.js	1.18 MB
js/spa2-615a20.js	1.18 MB
js/spa10-615a20.js	1.18 MB
js/spa1-615a20.js	1.18 MB

Webpack大规模应用开发

Big-scale app

思考：哪些方法提高构建速度？

1. 将大型库外链
2. 将库预先编译
3. 减少构建搜索或编译路径
4. 缓存
5. 并行

实验机器



标准型S1

2核

8G

将大型库外链

- External
常用于引入外链作为library

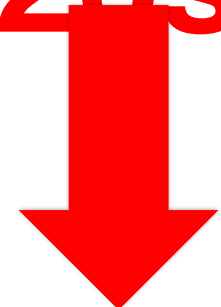
```
<script type="text/javascript" src="//libs/react-e4c416.js"></script>
```

```
// 使用外链
externals: {
  'react': 'React',
  'react-dom': 'ReactDOM',
  'preact': 'preact',
},
```


将大型库外链

全量
编译时间

26s



18s

Version: webpack 1.13.2

Time: 18090ms

Asset	Size
./css/spa7-f1177c.css	19.4 kB
js/spa9-d295b3.js	462 kB
js/spa7-d295b3.js	462 kB
js/spa6-d295b3.js	462 kB
js/spa5-d295b3.js	462 kB
js/spa4-d295b3.js	461 kB
js/spa3-d295b3.js	461 kB
js/spa2-d295b3.js	461 kB
js/spa10-d295b3.js	461 kB
js/spa1-d295b3.js	461 kB

将库预先编译

- DllPlugin

动态连接库，

即事先将包打好，再引用

```
const webpack = require('webpack');
var config = require('./config'),
    configWebpack = config.webpack;

const vendors = [
  'react',
  'react-dom',
  'redux',
  'react-redux',
  'redux-thunk',
  'lodash.merge',
];

module.exports = {
  output: {
    path: configWebpack.path.src + '/libs/',
    filename: '[name].js',
    library: '[name]',
    libraryTarget: 'umd',
  },
  entry: {
    "lib": vendors,
  },
  plugins: [
    new webpack.DllPlugin({
      path: './tools/manifest.json',
      name: '[name]',
      context: __dirname,
    }),
  ],
};
```

将库预先编译

- DllReferencePlugin

```
new webpack.DllReferencePlugin({  
  context: __dirname,  
  manifest: require('./manifest.json'),  
}),
```


将库预先编译

- noParse

将部份库预先编译好为es5

```
noParse: [  
  /utils/,  
  /spin/,  
  /net/,  
  /cgiPath/,  
  /constants/,  
]
```

将库预先编译

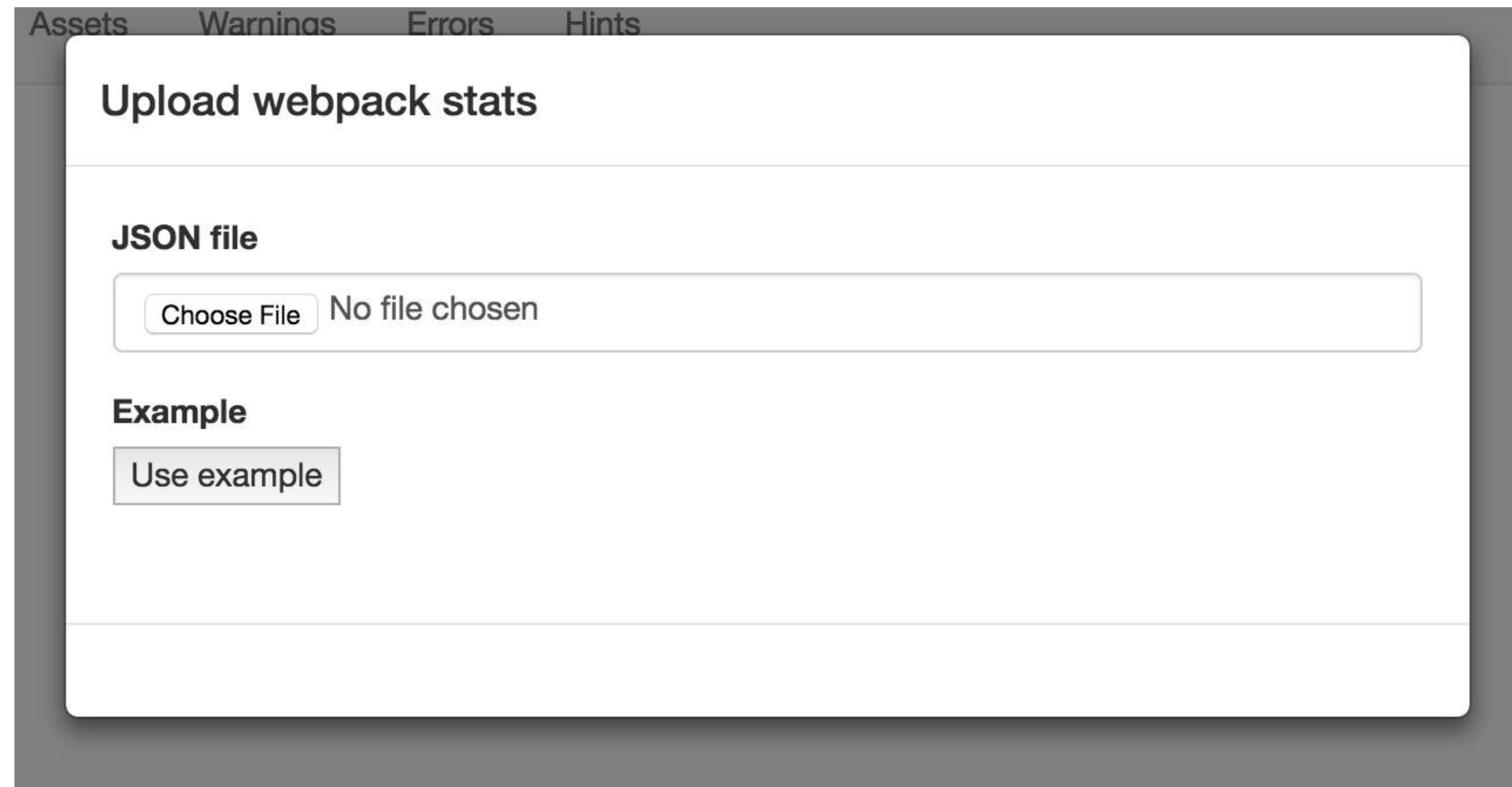
- PrefetchPlugin
stats-webpack-plugin

```
new StatsPlugin('stats.json', {  
  chunkModules: true,  
})
```

将库预先编译

- PrefetchPlugin

<http://webpack.github.io/analyse/>



The image shows a modal dialog box titled "Upload webpack stats". At the top, there are four tabs: "Assets", "Warnings", "Errors", and "Hints". The main content area has a section labeled "JSON file" which contains a file selection interface with a "Choose File" button and the text "No file chosen". Below this is an "Example" section with a "Use example" button.

将库预先编译

- PrefetchPlugin

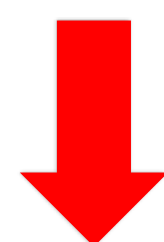
hints => long module build chains

```
new webpack.PrefetchPlugin('redux-devtools-log-monitor/lib/index.js'),  
new webpack.PrefetchPlugin('redux-devtools-dock-monitor/lib/index.js'),  
new webpack.PrefetchPlugin('core-js/library'),  
new webpack.PrefetchPlugin('babel-runtime/core-js'),
```


将库预先编译

启动时间

26s



16.8s

Version: webpack 1.13.2

Time: 16870ms

Asset	Size
./css/spa7-4d7ef5.css	19.3 kB
js/spa9-45548a.js	332 kB
js/spa7-45548a.js	332 kB
js/spa6-45548a.js	332 kB
js/spa5-45548a.js	332 kB
js/spa4-45548a.js	332 kB
js/spa3-45548a.js	332 kB
js/spa2-45548a.js	332 kB
js/spa10-45548a.js	332 kB
js/spa1-45548a.js	332 kB

减少构建搜索或编译路径

- resolve.alias

```
alias: {  
  // 使用压缩版本redux  
  'redux': 'redux/dist/redux.min',  
  'react-redux': 'react-redux/dist/react-redux',  
  'react-router': 'react-router/umd/ReactRouter',  
  'react-router-redux': 'react-router-redux/dist/ReactRouterRedux',  
  'utils': path.join(configWebpack.path.src, '/js/common/utils'),  
  'spin': path.join(configWebpack.path.src, '/js/common/spin'),  
  'spinner': path.join(configWebpack.path.src, '/page/common/components/spinner/index.js'),  
  'net': path.join(configWebpack.path.src, '/js/common/net'),  
  'touch': path.join(configWebpack.path.src, '/page/common/components/touch/index.js'),  
  'scroll': path.join(configWebpack.path.src, '/page/common/components/scroll/index.js'),  
  'pure-render-decorator': path.join(configWebpack.path.src, '/js/common/pure-render-decorator'),  
}
```

减少构建搜索或编译路径

- `resolve.unsafeCache`

保存搜索过的路径，下次启动编译，不再搜索。

减少构建搜索或编译路径

- exclude

```
{
  test: /\.jsx$/,
  loader: 'babel',
  query: {
    "plugins": [
      ["transform-helper-modules"],
      ["transform-decorators-legacy"],
      ["transform-react-jsx", { "pragma":"preact.h" }]
    ],
    presets: [
      'es2015-loose',
    ]
  },
  exclude: /node_modules/,
},
```

减少构建搜索或编译路径

- 慎用
 1. resolve.root
 2. resolve.modulesDirectory
 3. resolve.fallback

减少构建搜索或编译路径

全量
编译时间
18s
↓
15s

Version: webpack 1.13.2

Time: 15145ms

Asset	Size
./css/spa7-4d7ef5.css	19.3 kB
js/spa9y-f9e424.js	379 kB
js/spa7-f9e424.js	379 kB
js/spa6-f9e424.js	379 kB
js/spa5-f9e424.js	379 kB
js/spa4-f9e424.js	379 kB
js/spa3-f9e424.js	379 kB
js/spa2-f9e424.js	379 kB
js/spa10-f9e424.js	379 kB
js/spa1-f9e424.js	379 kB

缓存

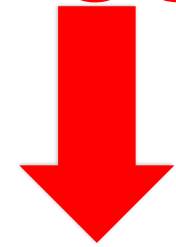
- babel-loader

```
{  
  test: /\.js$/,  
  loader: 'babel',  
  query: {  
    cacheDirectory: './webpack_cache/',  
    plugins: [  
      ["transform-helper-modules"],  
      ['transform-decorators-legacy']  
    ],  
    presets: [  
      'es2015-loose',  
      'react',  
    ]  
  },  
  exclude: /node_modules/,  
},
```


缓存

全量
编译时间

15s



8s

Version: webpack 1.13.2

Time: 8100ms

Asset	Size
./css/spa7-4d7ef5.css	19.3 kB
js/spa9y-f9e424.js	379 kB
js/spa7-f9e424.js	379 kB
js/spa6-f9e424.js	379 kB
js/spa5-f9e424.js	379 kB
js/spa4-f9e424.js	379 kB
js/spa3-f9e424.js	379 kB
js/spa2-f9e424.js	379 kB
js/spa10-f9e424.js	379 kB

并行

- Happypack
- js

```
new HappyPack({
  id: 'jsHappy',
  loaders: [{
    path: 'babel',
    query: {
      cacheDirectory: './webpack_cache/',
      plugins: ['transform-decorators-legacy'],
      presets: [
        'es2015-loose',
        'react',
      ]
    }
  ]
}),
```

并行

- happypack
- css (用不了extract-text-webpack-plugin)

```
{  
  test: /\.less$/,  
  loader: "happypack/loader?id=lessHappy",  
  include: path.resolve(configWebpack.path.src)  
},
```

并行

- happypack
- css (用不了extract-text-webpack-plugin)

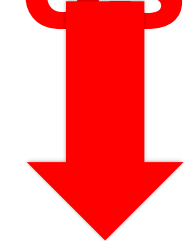
```
new HappyPack({  
  id: 'lessHappy',  
  loaders: ['style', 'css', 'less'],  
}),
```


并行

- happypack

全量
编译时间

8s



5s

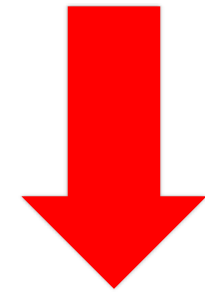
Version: webpack 1.13.2

Time: 5308ms

Asset	Size
libs/react-e4c416.js	147 kB
js/spa9-3efe60.js	415 kB
js/spa7-3efe60.js	415 kB
js/spa6-3efe60.js	417 kB
js/spa5-3efe60.js	417 kB
js/spa4-3efe60.js	417 kB
js/spa3-3efe60.js	417 kB
js/spa2-3efe60.js	417 kB
js/spa10-3efe60.js	417 kB
js/spa1-3efe60.js	417 kB

成果

26s

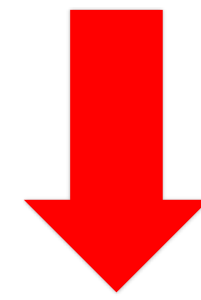


5s

成果

如果有代码丑化，如何？

66s



19.5s

webpack.optimize.
UglifyJsPlugin

webpack-uglify-parallel

成果

如果有代码丑化，如何？

```
Version: webpack 1.13.2
Time: 65885ms
```

Asset	Size
./css/spa7-36cf9e.css	11.3 kB
js/spa9-615a20.js	373 kB
js/spa7-615a20.js	373 kB
js/spa6-615a20.js	373 kB
js/spa5-615a20.js	373 kB
js/spa4-615a20.js	373 kB
js/spa3-615a20.js	373 kB
js/spa2-615a20.js	373 kB
js/spa10-615a20.js	373 kB
js/spa1-615a20.js	373 kB

```
Version: webpack 1.13.2
Time: 19590ms
```

Asset	Size
libs/react-e4c416.js	147 kB
js/spa9-ef7be5.js	165 kB
js/spa7-ef7be5.js	165 kB
js/spa6-ae7ae9.js	165 kB
js/spa5-ae7ae9.js	165 kB
js/spa4-ef7be5.js	165 kB
js/spa3-ef7be5.js	165 kB
js/spa2-ef7be5.js	165 kB
js/spa10-2d4a27.js	165 kB
js/spa1-f7f79b.js	165 kB

思考： 哪些方法提升开发体验

Webpack有无更好的livereload方式？

旧的live reload都是整个页面刷

思考： 哪些方法提升开发体验

热刷新/热替换

1. 自建server
2. React-hot-loader
3. HotModuleReplacementPlugin

热刷新/热替换

自建server

```
var express = require('express');
var app = express();
var webpack = require('webpack');
var webpackDevMiddleware = require("webpack-dev-middleware");
var webpackHotMiddleware = require("webpack-hot-middleware");
var proxy = require('proxy-middleware');

var webpackConfig = require("./webpack.dev.js"),
    config = require("./config.js");
var port = config.server.port;

for (var key in webpackConfig.entry) {
  webpackConfig.entry[key].unshift('webpack-hot-middleware/client');
}

var compiler = webpack(webpackConfig);
app.use(webpackDevMiddleware(compiler, {
  hot: true,
  // historyApiFallback: false,
  noInfo: false,
  stats: {
    colors: true
  },
})));
app.use(webpackHotMiddleware(compiler));
// 前端转发
app.use(config.server.hostDirectory, proxy('http://localhost:' + port));
// 后台转发
app.use('/api/', proxy('http://localhost:3001'));

app.listen(port, function(err) {
  if (err) {
    console.error(err);
  }
  else {
    console.info("Listening on port %s. Open up http://localhost:%s/ in your browser.", port);
  }
});
```

热刷新/热替换

自建server有什么好处

场景： 产品找你配置代理体验

proxy-middleware

热刷新/热替换

react-hot-loader

```
{  
  test: /\.js$/,  
  loaders: ['react-hot'],  
  exclude: /node_modules/,  
},
```

热刷新/热替换

HotModuleReplacementPlugin

```
plugins: [  
  // new Clean(['dist', 'build']),  
  // new webpack.optimize.CommonsChunkPlugin('main', null, false),  
  new webpack.HotModuleReplacementPlugin(), // 如果server.js好像需要开启?  
  new webpack.optimize.OccurenceOrderPlugin(),  
  new HtmlWebpackPlugin({
```

思考： 哪些方法提升开发体验

1个几万行的js bundle报错，如何定位？



The image shows a web browser's developer console with an error message and a corresponding code snippet. The error message is "Uncaught Error: Cannot find module \"../container/index\"", which is highlighted with a red border. Below the error message, the text "[HMR] connected" is visible. To the right of the error message, the file path "index.js:23873" is displayed, and below it, "index.js:11801" is visible. Below the console, a code snippet is shown, with the line "var e = new Error(\"Cannot find module \"../container/index\");" highlighted in red, matching the error message. The code snippet is as follows:

```
1 pack_require__(436);  
2  
3 pack_require__(!(function webpackMissingModule() { var e = new Error("Cannot find module \"../container/index\");  
4  
5 ropRequireDefault(_index);  
6
```


思考： 哪些方法提升开发体验

devtool

devtool	build speed	rebuild speed	production supported	quality
eval	+++	+++	no	generated code
cheap-eval-source-map	+	++	no	transformed code (lines only)
cheap-source-map	+	o	yes	transformed code (lines only)
cheap-module-eval-source-map	o	++	no	original source (lines only)
cheap-module-source-map	o	-	yes	original source (lines only)
eval-source-map	—	+	no	original source
source-map	—	—	yes	original source

思考：哪些方法提升开发体验

devtool



ruanyf

Webpack为什么难用？一个例子是，单单sourcemap的配置就有7种，文档根本没怎么解释！React创始人之一的Pete Hunt说，他也搞不清楚，更别提新手了。

 网页链接

 View less |  Full Size |  Rotate Left |  Rotate Right

devtool

Choose a developer tool to enhance debugging.

`eval` - Each module is executed with `eval` and `//# sourceURL`.

`source-map` - A SourceMap is emitted. See also `output.sourceMapFilename`.

`hidden-source-map` - Same as `source-map`, but doesn't add a reference comment to the bundle.

`inline-source-map` - A SourceMap is added as DataUri to the JavaScript file.

`eval-source-map` - Each module is executed with `eval` and a SourceMap is added as DataUri to the `eval`.

`cheap-source-map` - A SourceMap without column-mappings. SourceMaps from loaders are not used.

`cheap-module-source-map` - A SourceMap without column-mappings. SourceMaps from loaders are simplified to a single mapping per line.

Today 10:13 come from 微博手机版

Favorite

Forward 13

Comment 5

 11

思考： 哪些方法提升开发体验

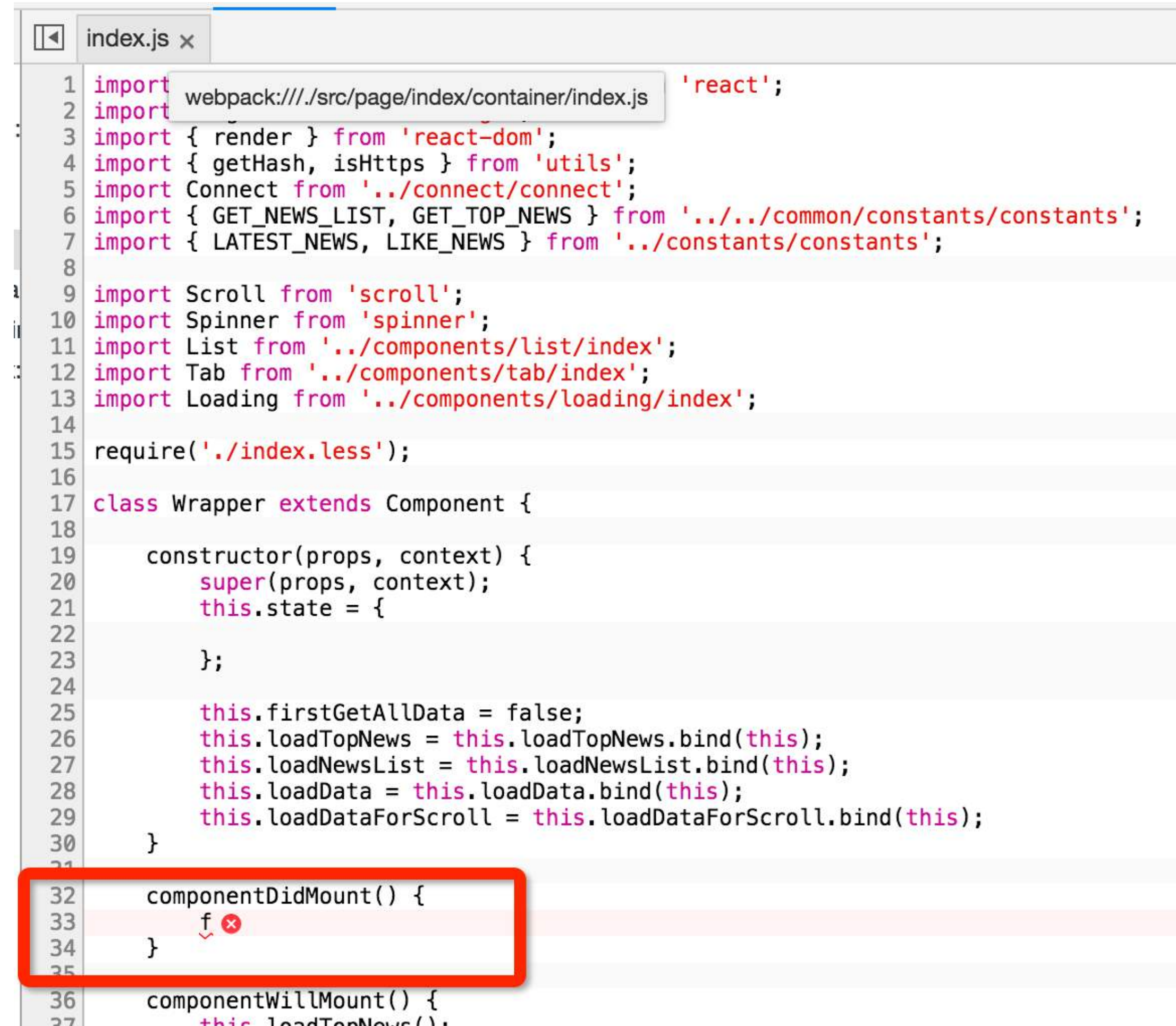
devtool

个人喜欢 inline-source-map, 信息详尽

eval, 速度快

思考： 哪些方法提升开发体验

devtool



```
1 import { render } from 'react-dom';
2 import { getHash, isHttps } from 'utils';
3 import Connect from '../connect/connect';
4 import { GET_NEWS_LIST, GET_TOP_NEWS } from '../../common/constants/constants';
5 import { LATEST_NEWS, LIKE_NEWS } from '../constants/constants';
6
7 import Scroll from 'scroll';
8 import Spinner from 'spinner';
9 import List from '../components/list/index';
10 import Tab from '../components/tab/index';
11 import Loading from '../components/loading/index';
12
13 require('./index.less');
14
15 class Wrapper extends Component {
16   constructor(props, context) {
17     super(props, context);
18     this.state = {
19
20     };
21
22     this.firstGetAllData = false;
23     this.loadTopNews = this.loadTopNews.bind(this);
24     this.loadNewsList = this.loadNewsList.bind(this);
25     this.loadData = this.loadData.bind(this);
26     this.loadDataForScroll = this.loadDataForScroll.bind(this);
27
28   }
29
30   componentDidMount() {
31     f
32   }
33
34   componentWillMount() {
35     this.loadTopNews();
36   }
37 }
```

思考： 哪些方法提升开发体验

区分开发环境与生产环境

1. 在package.json里面的script设置环境变量
2. 在webpack.xxx.js使用process.env.NODE_ENV进行判断

思考： 哪些方法提升开发体验

```
"scripts": {  
  "dev": "node ./tools/gulpfile.js&&node ./tools/webpack.config.js",  
  "pub": "node ./tools/gulpfile.js&&export NODE_ENV=__PROD__&&node ./-  
},  
// in console: f
```

& => 串行

&& => 并行

Mac & Linux => export

Windows => set

思考： 哪些方法提升开发体验

```
new webpack.DefinePlugin({  
  "process.env": {  
    NODE_ENV: JSON.stringify(config.env)  
  }  
}),
```

```
if ("__PROD__" !== process.env.NODE_ENV) {  
  window.console.dev = function(msg) {  
    console.log(msg);  
  };  
  module.exports = require('./Root.dev');  
}  
else {  
  window.console.dev = function(msg) {};  
  module.exports = require('./Root.prod');  
}
```

终于项目发布了



老板：哇靠，首屏时间10秒，年终还要不要？



我：。。。。。。

大规模应用如何借助 **webpack**优化

webpack scalability

webpack的手段

1. cdn & hash
2. 减少首屏包大小
3. 减少总体包大小

cdn与hash

- 为什么用cdn

如果没钱，最直接的使用cdn办法，就是多个域名指向同一台服务器。

```
output: {  
  // ...  
  publicPath: configWebpack.cdn,  
  path: path.join(configWebpack.path.pub),  
  filename: "[name]-" + configWebpack.chunkhash + ".js",  
},  
module: {  
  // ...  
}
```

- `options.publicPath`: string override the `publicPath` setting for this loader

cdn与hash

- Webpack能设置哪些cdn
 1. cdn of js / img => output.publicPath
 2. cdn of css => extract-text-webpack-plugin

cdn与hash

- 为什么给静态资源加hash
 1. 避免覆盖旧文件
 2. 回滚方便，只需要回滚html
 3. 由于文件名唯一，可开启服务器永远缓存

cdn与hash

基本概念

1. hash

2. chunkhash

3. contenthash

(extract-text-webpack-plugin)

cdn与hash

使你的chunk根据内容生成md5

Webpack-md5-hash

拆包

- require
- require.ensure
- router

离线包与提取公共包

- 离线包大影响下载/更新速度

离线包与提取公共包

- CommonsChunkPlugin

1. Entry chunk

```
new webpack.optimize.CommonsChunkPlugin({  
  name: "commons",  
  // (the commons chunk name)  
  filename: "commons.js",  
  // (the filename of the commons chunk)  
  // minChunks: 3,  
  // (Modules must be shared between 3 entries)  
  chunks: ["js/index", "js/spa"],  
  // (Only use these entries)  
}),
```


离线包与提取公共包

- CommonsChunkPlugin

2. Explicit vendor chunk

```
entry: {  
  'js/index': [configWebpack.path.src + '/page/index/main.js'],  
  'js/pindex': [configWebpack.path.src + '/page/pindex/main.jsx'],  
  'js/spa': [configWebpack.path.src + '/page/spa/main.js'],  
  'js/vendor': ['react', 'react-dom'],  
},
```

```
new webpack.optimize.CommonsChunkPlugin({  
  name: "vendor",  
  // (the commons chunk name)  
  filename: "vendor.js",  
  // (the filename of the commons chunk)  
}),
```

离线包与提取公共包

- CommonsChunkPlugin

3. 将被拆包公共部份提取到common chunk中

4. 同3， 但将公共部份提到到单独的文件中

离线包与提取公共包

- CommonsChunkPlugin带来的问题
 1. 包含动态加载chunk, 导致hash频繁变化
chunk-manifest-webapck-plugin
 2. 增量更新, commonchunk与chunk中module id对应不上
recordsPath, recordsInputPath, recordsOutputPath

离线包与提取公共包

- CommonsChunkPlugin带来的问题

```
recordsPath: path.resolve('records.json'),
```

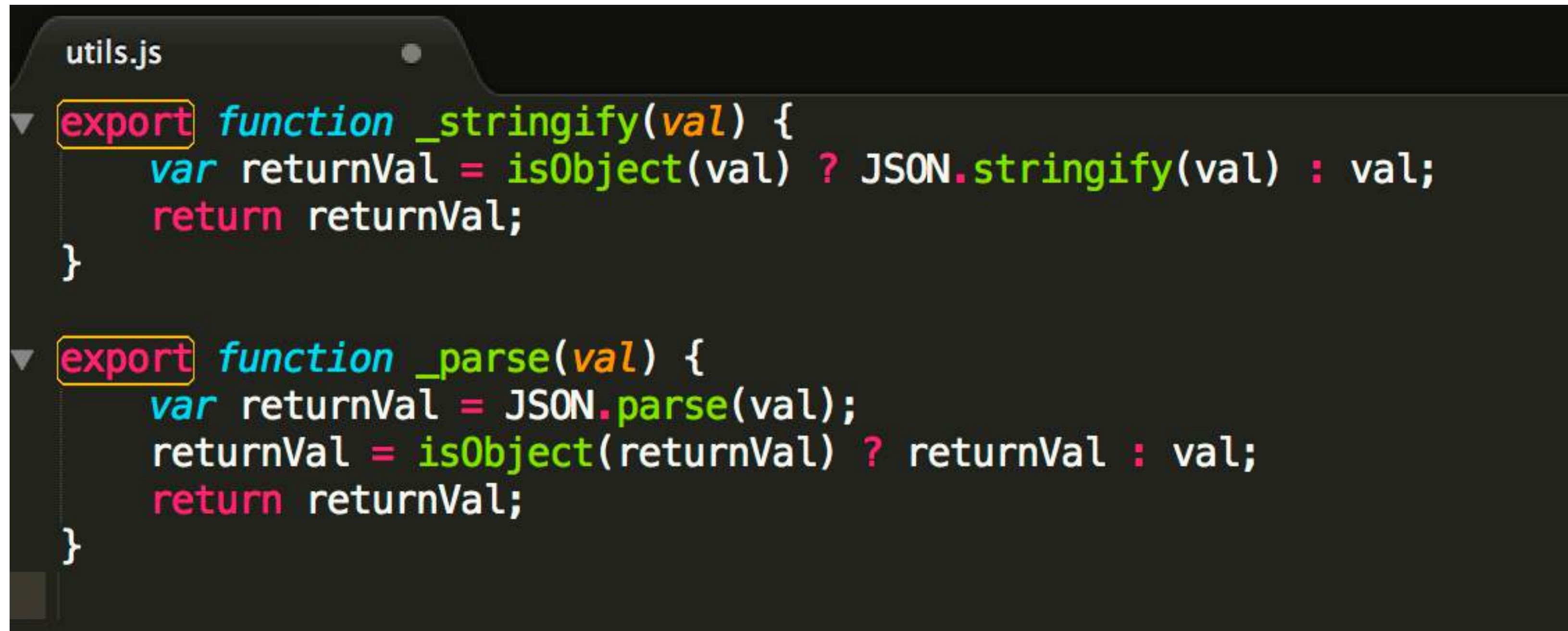
```
ose\", \"react\")}!src/page/common/components/touch/index.js\": 21,  
ose\", \"react\")}!src/page/spa/constants/constants.js\": 22,  
  
ose\", \"react\")}!src/page/common/actions/actions.js\": 28,  
ose\", \"react\")}!src/page/common/components/spinner/index.js\": 29,  
ose\", \"react\")}!src/page/common/middleware/api.js\": 30,  
ose\", \"react\")}!src/page/index/constants/constants.js\": 31,  
ose\", \"react\")}!src/page/pindex/constants/constants.js\": 32,  
ose\", \"react\")}!src/page/spa/connect/connect.js\": 33,
```


webpack2前瞻

webpack future

Tree-shaking

就是把没用的代码，从代码树里摇走



```
utils.js
▼ export function _stringify(val) {
    var returnVal = isObject(val) ? JSON.stringify(val) : val;
    return returnVal;
}

▼ export function _parse(val) {
    var returnVal = JSON.parse(val);
    returnVal = isObject(returnVal) ? returnVal : val;
    return returnVal;
}
```

Tree-shaking

webpack2.0 tree-shaking results in larger bundle size #2451

 **Open** **lcxfs1991** opened this issue on May 8 · 11 comments



lcxfs1991 commented on May 8



Tree-shaking feature really excites me.

However, when I try to use it, it results in larger bundle size. Why? It does eliminate the unused code.

My guess is the translation of import and export in es6.



3



bebraw added **question** **webpack-2** labels on May 8

Tree-shaking

1. babel-preset-es2015-loose-native-modules
2. babel-preset-es2015-native-modules

去掉了transform-es2015-modules-commonjs

Tree-shaking

Webpack2

js/index-932136.js	69.4 kB	1	[emitted]	index
js/pindex-8e6acd.js	66.1 kB	2	[emitted]	pindex
js/spa-66e540.js	165 kB	0	[emitted]	spa

Webpack2 with tree-shaking

js/index-511401.js	65.5 kB	1	[emitted]	index
js/pindex-ec2b41.js	62.4 kB	2	[emitted]	pindex
js/spa-a9c0e3.js	161 kB	0	[emitted]	spa

Webpack1

js/index-fec99b.js	68.8 kB	1	[emitted]	js/index
js/pindex-28565f.js	66.3 kB	2	[emitted]	js/pindex
js/spa-05414c.js	165 kB	0	[emitted]	js/spa

Code Split

- `system.import`
- 返回promise
- 捕获加载error

Code Split

```
export const routeConfig = {
  component: App,
  childRoutes: [
    {
      path: '/',
      getComponent(location, cb) {
        System.import('../container/index')
          .then(loadRoute(cb))
          .catch(errorLoading);
      }
    },
    {
      path: 'detail/:id/:commentid',
      getComponent(location, cb) {
        System.import('../container/detail')
          .then(loadRoute(cb))
          .catch(errorLoading);
      }
    },
    {
      path: 'comment/:id',
      getComponent(location, cb) {
        System.import('../container/comment')
          .then(loadRoute(cb))
          .catch(errorLoading);
      }
    }
  ],
};
```

webpack2 Demo

- <https://github.com/SteamerTeam/steamer-react/tree/dev>

webpack2生成代码分析

- <https://github.com/lcxfs1991/blog/issues/14>

如有谬误， 恳请斧正

Please give a shot