

link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=114 sentences=203, words=1701

1. sourcemap

1.1 什么是sourceMap

- sourcemap是为了解决开发代码与实际运行代码不一致时帮助我们debug到原始开发代码的技术
- webpack通过配置可以自动给我们 source maps文件，map文件是一种对应编译文件和源文件的方法
- [source-map \(https://github.com/mozilla/source-map\)](https://github.com/mozilla/source-map)

| devtool | build | rebuild | production | quality |
|--|---------|---------|------------|-------------------------------|
| (none) | fastest | fastest | yes | bundled code |
| eval | fastest | fastest | no | generated code |
| eval-cheap-source-map | fast | faster | no | transformed code (lines only) |
| eval-cheap-module-source-map | slow | faster | no | original source (lines only) |
| eval-source-map | slowest | fast | no | original source |
| eval-nosources-source-map | | | | |
| eval-nosources-cheap-source-map | | | | |
| eval-nosources-cheap-module-source-map | | | | |
| cheap-source-map | fast | slow | yes | transformed code (lines only) |
| cheap-module-source-map | slow | slower | yes | original source (lines only) |
| inline-cheap-source-map | fast | slow | no | transformed code (lines only) |
| inline-cheap-module-source-map | slow | slower | no | original source (lines only) |
| inline-source-map | slowest | slowest | no | original source |

类型 含义 **source-map** 原始代码 最好的sourcemap质量有完整的结果,但是会很慢 **eval-source-map** 原始代码 同样道理,但是最高的质量和最低的性能 **cheap-module-eval-source-map** 原始代码(只有行内) 同样道理,但是更高的质量和更低性能 **cheap-eval-source-map** 转换代码(行内) 每个模块被eval执行,并且sourcemap作为eval的一个dataurl eval 生成代码 每个模块都被eval执行,并且存在@sourceURL,带eval的构建模式能cache SourceMap **cheap-source-map** 转换代码(行内) 生成的sourcemap没有列映射,从loaders生成的sourcemap没有被使用 **cheap-module-source-map** 原始代码(只有行内) 与上面一样除了每行特点的从loader中进行映射 **hidden-source-map** 隐藏sourcemap **nosources-source-map** 控制台能正确提示报错的位置而不暴露源码

1.2 配置项

- 配置项其实只是五个关键字eval、source-map、cheap、module和inline的组合

关键字 含义 **source-map** 产生.map文件 **eval** 使用eval包裹模块代码 **cheap** 不包含列信息(关于列信息的解释下面会有详细介绍)也不包含loader的sourcemap **module** 包含loader的sourcemap(比如jsx到js, babel的sourcemap),否则无法定义源文件 **inline** 将.map作为DataURI嵌入,不单独生成.map文件

1.2.1 source-map

src/index.js

```
let a=1;
let b=2;
let c=3;
```

dist/main.js

```
((
  "src/index.js":
  (function (module, exports) {
    let a = 1;
    let b = 2;
    let c = 3;
  })
));
```

1.2.2 eval

- 用 eval 执行代码
- whyeval (<https://github.com/webpack/docs/wiki/build-performance#sourcemaps>)

```
((
  "src/index.js":
  (function (module, exports) {
    eval("let a=1;\r\nlet b=2;\r\nlet c=3;\n\n\n// # sourceURL=webpack:///./src/index.js?");
  })
));
```

- eval-source-map就会带上源码的sourceMap
- 加了eval的配置生成的sourceMap会作为DataURI嵌入,不单独生成.map文件
- 官方比较推荐开发场景下使用eval的构建模式,因为它能 cache sourceMap,从而rebuild的速度会比较快

```
((
  "src/index.js":
  (function (module, exports) {
    eval("let a=1;\r\nlet b=2;\r\nlet c=3;// # sourceURL=[module]\n\n// # sourceMappingURL=data:application/json;charset=utf-8;base64,In0=\n\n// # sourceURL=webpack-internal:///./src/index.js\n");
  })
));
```

devtool: "eval-source-map" is really as good as devtool: "source-map", but can cache SourceMaps for modules. It's much faster for rebuilds.

1.2.3 inline

- inline就是将map作为DataURI嵌入,不单独生成.map文件
- inline-source-map

```
((
  "src/index.js":
  (function (module, exports) {
    let a = 1;
    let b = 2;
    let c = 3;
  })
));
```

1.2.4 cheap(低开销)

- cheap (6#x4F4E;6#x5F00;6#x9500;) 的sourcemap, 因为它没有生成列映射(column mapping),只是映射行数
- 开发时我们有行映射也够用了,开发时可以使用cheap
- cheap-source-map

1.2.5 module

- Webpack会利用loader将所有非js模块转化为webpack可处理的js模块,而增加上面的cheap配置后也不会有loader模块之间对应的sourceMap
- 什么是模块之间的sourceMap呢?比如jsx文件会经历loader处理成js文件再混淆压缩, 如果没有loader之间的sourceMap, 那么在debug的时候定义到上图中的压缩前的js处,而不能追踪到jsx中
- 所以为了映射到loader处理前的代码,我们一般也会加上module配置
- cheap-module-source-map

1.3 演示

1.3.1 安装

```
cnpm i webpack webpack-cli webpack-dev-server babel-loader @babel/core @babel/preset-env style-loader css-loader less-loader less file-loader url-loader -D
```

1.3.2 webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
module.exports = {
  mode: 'development',
  devtool: 'cheap-module-source-map',
  entry: './src/index.js',
  module: {
    rules: [
      {
        test: /\.js$/,
        use: [{
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }]
      }
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html'
    })
  ]
}
```

1.3.3 src/index.js

```
import './sum';
sum(1,2);
```

1.4 最佳实践

1.4.1 开发环境

- 我们在开发环境对sourceMap的要求是：快（eval），信息全（module），
- 且由于此时代码未压缩，我们并不那么在意代码列信息(cheap),
- 所以开发环境比较推荐配置：devtool: cheap-module-eval-source-map

1.4.2 生产环境

- 一般情况下，我们并不希望任何人都可以在浏览器直接看到我们未编译的源码，
- 所以我们应该直接提供sourceMap给浏览器。但我们又需要sourceMap来定位我们的错误信息，
- 这时我们可以设置 hidden-source-map
- 一方面webpack会生成sourcemap文件以提供给错误收集工具比如sentry，另一方面又不会为 bundle 添加引用注释，以避免浏览器使用。

2. sourcemap

- [compiler官方下载 \(https://developers.google.com/closure/compiler\)](https://developers.google.com/closure/compiler)
- [compiler珠峰镜像 \(http://img.zhufengpeixun.cn/compilerjar\)](http://img.zhufengpeixun.cn/compilerjar)
- [base64vlg在线转换 \(http://murzwin.com/base64vlg.html\)](http://murzwin.com/base64vlg.html)

2.1 生成sourcemap

script.js

```
let a=1;
let b=2;
let c=3;
```

```
java -jar compiler.jar --js script.js --create_source_map ./script-min.js.map --source_map_format=V3 --js_output_file script-min.js
```

script-min.js

```
var a=1,b=2,c=3;
```

script-min.js.map

```
{
  "version": 3,
  "file": "script-min.js",
  "lineCount": 1,
  "mappings": "AAAA,IAAIA,EAAE,CAAN,CACIC,EAAE,CADN,CAEIC,EAAE;",
  "sources": ["script.js"],
  "names": ["a", "b", "c"]
}
```

字段 含义 version: Source Source map的版本，目前为3 file: 转换后的文件名。转换后的文件名 sourceRoot 转换前的文件所在的目录。如果与转换前的文件在同一目录，该项为空 sources 转换前的文件,该项是一个数组,表示可能存在多个文件合并 names 转换前的所有变量名和属性名 mappings 记录位置信息的字符串

2.2 mappings属性

- 关键就是map文件的mappings属性。这是一个很长的字符串，它分成三层

对应 含义 第一层是行对应 以分号（;）表示，每个分号对应转换后源码的一行。所以，第一个分号前的内容，就对应源码的第一行，以此类推。 第二层是位置对应 以逗号（,）表示，每个逗号对应转换后源码的一个位置。所以，第一个逗号前的内容，就对应该行源码的第一个位置，以此类推。 第三层是位置转换 以VLQ编码表示，代表该位置对应的转换前的源码位置。

```
"mappings": "AAAA,IAAIA,EAAE,CAAN,CACIC,EAAE,CADN,CAEIC,EAAE;"
```

2.3 位置对应的原理

- 每个位置使用五位，表示五个字段

位置 含义 第一位 表示这个位置在（转换后的代码的）的第几列 第二位 表示这个位置属于sources属性中的哪一个文件 第三位 表示这个位置属于转换前代码的第几行 第四位 表示这个位置属于转换前代码的第几列 第五位 表示这个位置属于names属性中的哪一个变量

首先，所有的值都是以0作为基数的。其次，第五位不是必需的，如果该位置没有对应names属性中的变量，可以省略第五位,再次，每一位都采用VLQ编码表示；由于VLQ编码是变长的，所以每一位可以由多个字符构成

如果某个位置是AAAAA，由于A在VLQ编码中表示0，因此这个位置的五个位实际上都是0。它的意思是，该位置在转换后代码的第0列，对应sources属性中第0个文件，属于转换前代码的第0行第0列，对应names属性中的第0个变量。

□

2.4 相对位置 #

- 对于输出后的位置来说，到后边会发现它的列号特别大，为了避免这个问题，采用相对位置进行描述
- 第一次记录的输入位置和输出位置是绝对的，往后的输入位置和输出位置都是相对上一次的位置移动了多少

□

2.5 VLQ编码 #

- VLQ是Variable-length quantity 的缩写，是一种通用的、使用任意位数的二进制来表示一个任意大的数字的一种编码方式
- 这种编码需要用最高位表示连续性，如果是1，代表这组字节后面的一组字节也属于同一个数；如果是0，表示该数值到这就结束了
- 如何对数值137进行VLQ编码
 - 将137改写成二进制形式 10001001
 - 七位一组做分组，不足的补0 0000001 0001001
 - 最后一组开头补0，其余补1 10000001 00001001
 - 137的VLQ编码形式为10000001 00001001

```
let binary = 137..toString(2);
console.log(binary);
let padded = binary.padStart(Math.ceil(binary.length / 7) * 7, '0');
console.log(padded);
let groups = padded.match(/\d{7}/g);
groups = groups.map((group,index)=> (index==0?'1':'0')+group);
console.log(groups);
```

2.6 Base64 VLQ #

- 一个Base64字符只能表示6bit(2^6)的数据
- Base64 VLQ需要能够表示负数,于是用最后一位来作为符号标志位
- 由于只能用6位进行存储，而第一位表示是否连续的标志，最后一位表示正数/负数。中间只有4位，因此一个单元表示的范围为[-15,15]，如果超过了就要用连续标识位了
- 表示正负的方式
 - 如果这组数是某个数值的VLQ编码的第一组字节，那它的最后一位代表"符号"，0为正，1为负；
 - 如果不是，这个位没有特殊含义，被算作数值的一部分
- 在Base64 VLQ中，编码顺序是从低位到高位,而在VLQ中，编码顺序是从高位到低位

| Value | Char | Value | Char | Value | Char | Value | Char |
|-------|------|-------|------|-------|------|-------|------|
| 0 | A | 16 | Q | 32 | g | 48 | w |
| 1 | B | 17 | R | 33 | h | 49 | x |
| 2 | C | 18 | S | 34 | i | 50 | y |
| 3 | D | 19 | T | 35 | j | 51 | z |
| 4 | E | 20 | U | 36 | k | 52 | 0 |
| 5 | F | 21 | V | 37 | l | 53 | 1 |
| 6 | G | 22 | W | 38 | m | 54 | 2 |
| 7 | H | 23 | X | 39 | n | 55 | 3 |
| 8 | I | 24 | Y | 40 | o | 56 | 4 |
| 9 | J | 25 | Z | 41 | p | 57 | 5 |
| 10 | K | 26 | a | 42 | q | 58 | 6 |
| 11 | L | 27 | b | 43 | r | 59 | 7 |
| 12 | M | 28 | c | 44 | s | 60 | 8 |
| 13 | N | 29 | d | 45 | t | 61 | 9 |
| 14 | O | 30 | e | 46 | u | 62 | + |
| 15 | P | 31 | f | 47 | v | 63 | / |

```

let base64 = [
  'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
  'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
  'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
  'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '/'
];

function encode(num) {

  let binary = (Math.abs(num)).toString(2);

  binary = num >= 0 ? binary + '0' : binary + '1';

  let zero = 5 - (binary.length % 5);
  if (zero > 0) {
    binary = binary.padStart(Math.ceil(binary.length / 5) * 5, '0');
  }
  let parts = [];
  for (let i = 0; i < binary.length; i += 5) {
    parts.push(binary.slice(i, i + 5));
  }

  parts.reverse();

  for (let i = 0; i < parts.length; i++) {
    if (i === parts.length - 1) {
      parts[i] = '0' + parts[i];
    } else {
      parts[i] = '1' + parts[i];
    }
  }

  let chars = [];
  for (let i = 0; i < parts.length; i++) {
    chars.push(base64[parseInt(parts[i], 2)]);
  }

  return chars.join('')
}

let result = encode(137);
console.log(result);

```

2.7 计算位移

```

let base64 = [
  'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
  'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f',
  'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
  'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '/'
];

function getValue(char) {
  let index = base64.findIndex(item => item === char);
  let str = (index).toString(2);
  str = str.padStart(6, '0');

  let sign = str.slice(-1) === '0' ? 1 : -1;

  str = str.slice(1, -1);
  return parseInt(str, 2) * sign;
}

function decode(values) {
  let parts = values.split(',');
  let positions = [];
  for (let i = 0; i < parts.length; i++) {
    let chars = parts[i].split('');
    let position = [];
    for (let i = 0; i < chars.length; i++) {
      position.push(getValue(chars[i]));
    }
    positions.push(position);
  }
  return positions;
}

let positions = decode('AAAA,IAAIA,EAAE,CAAN,CACIC,EAAE,CADN,CAEIC,EAAE');

console.log('positions', positions);
let offsets = positions.map(item => [item[2], item[3], 0, item[0]]);
console.log('offsets', offsets);
let origin = {x: 0, y: 0};
let target = {x: 0, y: 0};
let mapping = [];
for (let i = 0; i < offsets.length; i++) {
  let [originX, originY, targetX, targetY] = offsets[i];
  origin.x += originX;
  origin.y += originY;
  target.x += targetX;
  target.y += targetY;
  mapping.push(`${origin.x},${origin.y}=>${target.x},${target.y}`);
}
console.log('mapping', mapping);

```

```

positions [
  [ 0, 0, 0, 0 ],
  [ 4, 0, 0, 4, 0 ],
  [ 2, 0, 0, 2 ],
  [ 1, 0, 0, -6 ],
  [ 1, 0, 1, 4, 1 ],
  [ 2, 0, 0, 2 ],
  [ 1, 0, -1, -6 ],
  [ 1, 0, 2, 4, 1 ],
  [ 2, 0, 0, 2 ]
]
offsets [
  [ 0, 0, 0, 0 ],
  [ 0, 4, 0, 4 ],
  [ 0, 2, 0, 2 ],
  [ 0, -6, 0, 1 ],
  [ 1, 4, 0, 1 ],
  [ 0, 2, 0, 2 ],
  [ -1, -6, 0, 1 ],
  [ 2, 4, 0, 1 ],
  [ 0, 2, 0, 2 ]
]
mapping [
  '[0,0]=>[0,0]',
  '[0,4]=>[0,4]',
  '[0,6]=>[0,6]',
  '[0,0]=>[0,7]',
  '[1,4]=>[0,8]',
  '[1,6]=>[0,10]',
  '[0,0]=>[0,11]',
  '[2,4]=>[0,12]',
  '[2,6]=>[0,14]'
]

```

3.调试代码

3.1 测试环境调试

Settings

Preferences

Workspace

Experiments

Blackboxing

Devices

Throttling

Locations

Shortcuts

Preferences

COLOR FORMAT: AS authored

- ☐ Enable Ctrl + 1-9 shortcut to switch panels
- ☒ Show What's New after each update
- ☐ Don't show Chrome Data Saver warning
- ☐ Disable paused state overlay

Sources

- ☐ Search in anonymous and content scripts
- ☐ Automatically reveal files in sidebar
- ☒ Enable JavaScript source maps

webpack.config.js

```

const HtmlWebpackPlugin = require('html-webpack-plugin');
const FileManagerPlugin = require('filemanager-webpack-plugin');
const webpack = require('webpack');
module.exports = {
  mode: 'production',
  devtool: false,
  entry: './src/index.js',
  resolveLoader: {
    modules: ['node_modules', 'loaders']
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        use: [{
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }]
      },
      {
        test: /\.scss$/,
        use: [
          { loader: 'style-loader' },
          {
            loader: 'css-loader',
            options: {
              sourceMap: true,
              importLoaders: 2
            }
          },
          { loader: "resolve-scss-url-loader" },
          {
            loader: 'sass-loader',
            options: { sourceMap: true }
          }
        ]
      },
      {
        test: /\. (jpg|png|gif|bmp) $/,
        use: [
          { loader: 'url-loader' }
        ]
      }
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html'
    }),
    new webpack.SourceMapDevToolPlugin({
      append: '//# sourceMappingURL=http://127.0.0.1:8081/[url]',
      filename: '[file].map',
    }),
    new FileManagerPlugin({
      onEnd: {
        copy: [{
          source: './dist/*.map',
          destination: 'C:/aprepere/zhufengsourcemap/sourcemap',
        }],
        delete: ['./dist/*.map'],
        archive: [{
          source: './dist',
          destination: './dist/dist.zip',
        }]
      }
    })
  ]
}

```

3.2 生产环境调试

- webpack打包仍然生成sourceMap，但是将map文件挑出放到本地服务器，将不含有map文件的部署到服务器，借助第三方软件（例如fiddler），将浏览器对map文件的请求拦截到本地服务器，就可以实现本地sourceMap调试

```

regex: (?inx)http://localhost:8080/(?<name>.+)$
*redir:http://127.0.0.1:8081/${name}

```

| # | Result | Protocol | Host |
|----|--------|----------|-----------------|
| 11 | 200 | HTTP | localhost:8080 |
| 12 | 200 | HTTP | localhost:8080 |
| 13 | 200 | HTTP | localhost:8080 |
| 14 | 200 | HTTP | Tunnel to |
| 15 | 101 | HTTP | localhost:8080 |
| 16 | 200 | HTTPS | app.yinxiang.co |
| 17 | 200 | HTTPS | app.yinxiang.co |

Rule Editor

MATCH: regex:(?inx)http://localhost:8080/(?<name>.+)\$

ACTION: *redir:http://127.0.0.1:8081/\${name}

Save Cancel

4.source-map-loader

- [source-map-loader \(https://www.webpackjs.com/loaders/source-map-loader\)](https://www.webpackjs.com/loaders/source-map-loader) 从当前存在的源码(从sourceMappingURL)中提供出map源码

```
cnpm i source-map-loader -D
```

4.1 script.js

```
let a=1;
let b=2;
let c=3;
```

```
java -jar compiler.jar --js script.js --create_source_map ./script-min.js.map --source_map_format=V3 --js_output_file script-min.js
```

4.2 script.min.js

```
var a=1,b=2,c=3;
```

4.3 script.min.map.js

```
{
  "version":3,
  "file":"script-min.js",
  "lineCount":1,
  "mappings":"AAAA,IAAIA,EAAE,CAAN,CACIC,EAAE,CADN,CAEIC,EAAE;",
  "sources":["script.js"],
  "names":["a","b","c"]
}
```

4.4 src/index.js

```
import './script-min.js';
```

4.5 webpack.config.js

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const FileManagerPlugin = require('filemanager-webpack-plugin');
const webpack = require('webpack');

module.exports = {
  mode: 'development',
  devtool: 'inline-source-map',
  entry: './src/index.js',
  resolveLoader: {
    modules: ['node_modules', 'loaders']
  },
  module: {
    rules: [
      + {
      +   test: /\.js$/,
      +   use: ['source-map-loader'],
      +   enforce: "pre"
      + },
      {
        test: /\.js$/,
        use: [{
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }]
      },
      {
        test: /\.scss$/,
        use: [
          { loader: 'style-loader' },
          {
            loader: 'css-loader',
            options: {
              sourceMap: true,
              importLoaders: 2
            }
          },
          + { loader: "resolve-url-loader" },
          + { loader: "resolve-scss-url-loader" },
          {
            loader: 'sass-loader',
            options: { sourceMap: true }
          }
        ],
      },
      {
        test: /\. (jpg|png|gif|bmp) $/,
        use: [
          { loader: 'url-loader' }
        ]
      }
    ],
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './src/index.html'
    })
  ]
}
```

3.参考

- [javascript source map算法 \(http://www.ruanyifeng.com/blog/2013/01/javascript_source_map.html\)](http://www.ruanyifeng.com/blog/2013/01/javascript_source_map.html)
- [devtool \(https://www.webpackjs.com/configuration/devtool/\)](https://www.webpackjs.com/configuration/devtool/)