



yanyongchao

2018 年 03 月 26 日

webpack4之高级篇

Tree Shaking

1. css Tree Shaking

```
yarn add purify-css purifycss-webpack -D
const glob = require('glob')
const PurifyCSSPlugin = require('purifycss-webpack')
// 去除无用的css
plugins: [
  new PurifyCSSPlugin({
    // 路劲扫描 nodejs内置 路劲检查
    paths: glob.sync(path.join(__dirname, 'pages/**/*.html'))
  })
]
```

2. js优化

```
yarn add webpack-parallel-uglify-plugin -D
const WebpackParallelUglifyPlugin = require('webpack-parallel-uglify-plugin')
plugins: [
  new WebpackParallelUglifyPlugin({
    uglifyJS: {
      output: {
        beautify: false, //不需要格式化
        comments: false //不保留注释
      },
      compress: {
        warnings: false, // 在UglifyJs删除没有用到的代码时不输出警告
        drop_console: true, // 删除所有的 `console` 语句, 可以兼容ie浏览器
        collapse_vars: true, // 内嵌定义了但是只用到一次的变量
        reduce_vars: true // 提取出出现多次但是没有定义成变量去引用的静态值
      }
    }
  })
]
```





大网站有多个页面，每个页面由于采用相同技术栈和样式代码，会包含很多公共代码 [链接](#)

```
// webpack4最新配置，可以搜索关键字查查配置项
optimization: {
  splitChunks: {
    cacheGroups: {
      commons: {
        chunks: 'initial',
        minChunks: 2,
        maxInitialRequests: 5, // The default limit is too small to showcase the effect
        minSize: 0, // This is example is too small to create commons chunks
        name: 'common'
      }
    }
  }
},
```

打包第三方类库

- 1.DllPlugin插件：用于打包出一个个动态连接库
2. DllReferencePlugin: 在配置文件中引入DllPlugin插件打包好的动态连接库

在package.json中

```
"scripts": {
  "build": "webpack --mode development",
  "build:dll": "webpack --config webpack.dll.config.js --mode development",
  "dev": "webpack-dev-server --open --mode development"
}
```

新建webpack.dll.config.js

```
const path = require('path')
const webpack = require('webpack')

/**
 * 尽量减小搜索范围
 * target: '_dll_[name]' 指定导出变量名字
 */

module.exports = {
  entry: {
    vendor: ['jquery', 'lodash']
  },
```



```
// manifest是描述文件
plugins: [
  new webpack.DllPlugin({
    name: '_dll_[name]',
    path: path.join(__dirname, 'manifest.json')
  })
]
```

在webpack.config.js中

```
plugins: [
  new webpack.DllReferencePlugin({
    manifest: path.join(__dirname, 'manifest.json')
  })
]
```

执行npm run build:dll 就可以打包第三方包了

使用happypack

HappyPack就能让Webpack把任务分解给多个子进程去并发的执行，子进程处理完后再把结果发送给主进程。 happypack

```
npm i happypack@next -D
const HappyPack = require('happypack')
const os = require('os')
const happyThreadPool = HappyPack.ThreadPool({ size: os.cpus().length })

{
  test: /\.js$/,
  // loader: 'babel-loader',
  loader: 'happypack/loader?id=happy-babel-js', // 增加新的HappyPack构建loader
  include: [resolve('src')],
  exclude: /node_modules/,
}

plugins: [
  new HappyPack({
    id: 'happy-babel-js',
    loaders: ['babel-loader?cacheDirectory=true'],
    threadPool: happyThreadPool
  })
]
```





```
yarn add progress-bar-webpack-plugin -D
const ProgressBarPlugin = require('progress-bar-webpack-plugin')
plugins: [
  new ProgressBarPlugin({
    format: '  build [:bar] ' + chalk.green.bold(':percent') + ' (:elapsed seconds)'
  })
]
```

多页配置

[git完整代码链接](#)

```
entry: { //entry是一个字符串或者数组都是单页，是对象则是多页，这里是index页和base页
  index: './pages/index/index.js',
  base: './pages/base/base.js'
}
```

一个页面对应一个html模板，在plugin中

```
plugins: [
  new HtmlWebpackPlugin({
    template: './pages/index/index.html',
    filename: 'pages/index.html',
    hash: true,
    chunks: ['index', 'common'],
    minify: {
      removeAttributeQuotes: true
    }
  }),
  new HtmlWebpackPlugin({
    template: './pages/base/base.html',
    filename: 'pages/base.html',
    hash: true,
    chunks: ['base', 'common'],
    minify: {
      removeAttributeQuotes: true
    }
  })
]
```

完整多页配置代码如下 [git完整代码链接](#)



```
const CleanWebpackPlugin = require('clean-webpack-plugin')
const ExtractTextWebpackPlugin = require('extract-text-webpack-plugin')
const Webpack = require('webpack')
const glob = require('glob')
const PurifyCSSPlugin = require('purifycss-webpack')

// let cssExtract = new ExtractTextWebpackPlugin('static/css/[name].css')

module.exports = {
  entry: {
    index: './pages/index/index.js',
    base: './pages/base/base.js'
  },
  output: {
    path: path.join(__dirname, 'dist'),
    // name是entry的名字main, hash是根据打包后的文件内容计算出来的hash值
    filename: 'static/js/[name].[hash].js',
    publicPath: '/'
  },
  module: {
    rules: [
      {
        test: /\.css$/,
        use: ExtractTextWebpackPlugin.extract({
          fallback: 'style-loader',
          use: ['css-loader', 'postcss-loader']
        })
      },
      {
        test: /\.js$/,
        use: {
          loader: 'babel-loader',
          query: {
            presets: ['env', 'stage-0', 'react']
          }
        }
      }
    ]
  },
  optimization: {
    splitChunks: {
      cacheGroups: {
        commons: {
          chunks: 'initial',
          minChunks: 2
        }
      }
    }
  }
}
```





首页 ▾



```
}
},
plugins: [
  new CleanWebpackPlugin([path.join(__dirname, 'dist')]),
  new HtmlWebpackPlugin({
    template: './pages/index/index.html',
    filename: 'pages/index.html',
    hash: true,
    chunks: ['index', 'common'],
    minify: {
      removeAttributeQuotes: true
    }
  }),
  new HtmlWebpackPlugin({
    template: './pages/base/base.html',
    filename: 'pages/base.html',
    hash: true,
    chunks: ['base', 'common'],
    minify: {
      removeAttributeQuotes: true
    }
  }),
  new ExtractTextWebpackPlugin({
    filename: 'static/css/[name].[hash].css'
  }),
  new PurifyCSSPlugin({
    // 路径扫描 nodejs内置 路径检查
    paths: glob.sync(path.join(__dirname, 'pages/**/*.html'))
  })
],
devServer: {
  contentBase: path.join(__dirname, 'dist'),
  port: 9090,
  host: 'localhost',
  overlay: true,
  compress: true // 服务器返回浏览器的时候是否启动gzip压缩
}
}
```

vue-cli几个重要路径



一个帮助开发者成长的社区





未完待续：接下来我会基于vue-cli去从新搭建一个工程，单页，多页都会有哦！！！让你看懂vue-cli，并且能够自定义开发环境！！

关注下面的标签，发现更多相似文章

[JavaScript](#)[Webpack](#)

[掘金招聘前端开发、后端开发、运营经理、内容运营](#)

加入掘金和开发者一起成长。发送简历到 hr@xitu.io，期待你的加入！

评论

说说你的看法

丹哥一号 前端 @ 搬砖有限公司

大佬，请问可以把splitChunks和第三方DllPlugin结合在一起用吗？

▲ 0 评论 4天前

levai 小前端.....

大佬 你的GitHub地址失效了

▲ 0 1条评论 10天前

ytccccc FED @ weiyi

需要addAssetHtmlPlugin配合= =

▲ 0 评论 11天前

ytccccc FED @ weiyi

dll使用那个报错_dll_main is not defined ？

▲ 0 1条评论 11天前

刘成阳 程序员

[展开全部](#)

相关推荐

专栏 · Yao_xmu · 1天前 · 前端 / JavaScript

社招中级前端笔试面试题总结

 23

 4

专栏 · nw2018 · 2天前 · Node.js / 前端

从零开始搭建一个mock服务

 20

 3

专栏 · Hollis_公众号Hollis · 2小时前 · 后端 / 服务器

大家都在说的分布式系统到底是什么？

 18



专栏 · 棕小渐 · 2天前 · JavaScript / 前端

函数式编程（JavaScript描述）

 14



专栏 · wuzhengyan2015 · 1天前 · JavaScript / 前端


JavaScript原型梳理


 6



热 · 专栏 · 热心市民老贾 · 19小时前 · JavaScript

一个Promise面试题

 182

 11

专栏 · MarsDes · 2天前 · 微信小程序 / 前端

小程序 音频API采坑完全手册

 6





专栏 · wangpf · 2天前 · JavaScript / 前端






async/await带来的诟病

 3



专栏 · 黄Java · 1天前 · Webpack / React.js

TypeScript+Webpack+React组件库开发采坑实记

 4



