vue 脚手架 elementUi element-ui 兼容 ie 360 急速/兼容模式 完美处理

```
gaoqiang1112
                2019-07-16 17:51:35 💿 7061 🍁 已收藏 19
                                                                                              版权
    分类专栏: vue 前端 js 文章标签: vue vue脚手架ie显示问题
                                                   vue脚手架360浏览器兼容模式极速模式展示
                                                                              vuees6语法处理
直接上干货
```

如果你用vue的脚手架弄的项目 那么恭喜你 要改的 也并不多 如果 是自己搭建的框架 那么 请滤清框架结构后 按我的方法操作即可

1 引入 babel-polyfill

```
npm install babel-polyfill --save
```

import 'babel-polyfill'

app: './src/main.js'

entry: {

```
2 src下 mains文件
```

3 build文件夹下 webpack.base.conf.js

3 },

```
替换成
  1
       entry: {
  2
```

```
app: ['./node_modules/babel-polyfill/dist/polyfill.js','./src/main.js']
  3
         // app: './src/main.js'
       },
高潮来了
```

的 assetsSubDirectory 都设置成空 (其实这步是为了vue打包生产后 放在其

他服务端代码里 去掉最外层的static文件夹 和本文关系不大 这里只是友情提

4 config文件夹下 index.js 里 请把 dev 模式 和 prod模式

"browsers": ["> 1%", "last 2 versions", "not ie <= 8"]

示及记录用)

assetsSubDirectory:"" 5 根目录下的 .babelrc

复制

复制

```
}
6
7
        }],
```

"stage-2"

"presets": [

["env", {

"modules": false,

"targets": {

1 2

3

4 5

8

9

10

11

],

```
修改为
 12
 13
 14
     "presets": [
 15
         ["env", {
          "modules": false,
 16
          "useBuiltIns": "entry",
 17
 18
          "targets": {
            "browsers": ["> 1%", "last 2 versions", "not ie <= 8"]
 19
 20
         }],
 21
 22
         "stage-2"
 23
       ],
6 最牛逼的高潮来了
build文件夹下的 webpack.base.conf.js ( 没错 之前修改过 , 这里是楼主的修改历程 , 前面顺序均来自百度 , 最后这步是楼主自己加的 )
module: {
   ...(config.dev.useEslint ? [createLintingRule()] : []),
```

options:

es6语法

所以 加入过滤处理

test: /\.js\$/,

resolve('config')

4

5

},

const myHttp = {

} else {

new ExtractTextPlugin({

allChunks: true,

去resolve 报错的文件即可

new CSSSplitWebpackPlugin({

loader: 'babel-loader?cacheDirectory',

所以 在 config 文件下的index.js里 配置 proxyTable

changeOrigin:true,

postRequest: function(url, param) {

console.info('post使用mock假数据')
url = '/api/' + i[url].url

console.info('post使用real真数据')

url = J_API + i[url].url

'^/api': '/' //重写接口

param = param && typeof param == 'object' ? param : {}
if (i[url].isMock || i[url].isMock == undefined) {

pathRewrite: {

这里的include 添加

test: /\.vue\$/, loader: 'vue-loader', options: vueLoaderConfig

test: /\.svg\$/,

loader: 'svg-sprite-loader', include: [resolve('src/icons')],

test: /\.js\$/, loader: 'babet-loader?cacheDirectory',

```
原理很简单 项目里用了第三方elementUi 但是 babel处理的时候 并没有处理node_modules(确实也处理不过来,所以我们只能自己处
```

理),这里你可以针对 你引入的第三包 特殊的处理,加载即可,不过不建议 直接把包路径放这,因为太大了,所以 楼主把elementUi里

https://blog.csdn.net/gaoqiang1112

include: [resolve('src'), resolve('test'), resolve('node_modules/webpack-dev-server/clie

resolve('node modules/element-ui/packages'), resolve('node modules/element-ui/src')

有问题的2个文件夹处理了(其实可以更精确的处理,不过只是减少了打包的时间,所以处理这2个比较稳妥。)

7. ie9中 首页依然白屏 查看原因 报错 js依然有代码不给识别 因为 config里有

到这里 终结!!保证你的项目 无论在ie 还是360狗屁模式 都会好使!欢迎留

include: [resolve('src'), resolve('static'), resolve('test'), resolve('config'), resolve('node)

8 以为大功告成 尝试了一下登录 报错 拒绝访问

大意了 好好 360 兼容+急速 目前都好使 不过 使用ie9 实地检测 发现依然报错 记录 处理过程

启动首页显示!!!!!

百度搜索很久 终于确定原因 是因为 ie9下 axios中有跨域问题 这里 我写的时候用的是easymock的地址 但依然跨域

```
proxyTable: {
1
2
      '/api':{
           target: https://www.easy-mock.com/mock/5cf477f520156326ddbeb251/example/',
3
```

{description: "拒绝访问。 ", message: "拒绝访问。 ", name: "Error", number: -2147024891}

```
然后 在 我们二次封装的 axios里 每个模拟mock请求前 都加上 /api/
```

```
记录些杂七杂八的 报错 和 原因
a. ie9中页面样式串版
解决办吧:1. ie9中 flex布局不支持 自己写的flex布局及elementui组件自带的flex布局 需要自己手动处理
      2. ie9中 单个加载文件 超过32行 有可能加载不出来或丢失 而webpack打包 脚手架默认会打到一个css文件里
        所以引入 CSSSplitWebpackPlugin 并且把 ExtractTextPlugin 里的allChunks 改为 true即可
```

// increasing file size: https://github.com/vuejs-templates/webpack/issues/1110

这里的size 4000 是针对ie设计的 可以在 www.npmjs.com里 搜索 这个 css-split-webpack-plugin 可以了解

b. SCRIPT5007: Array.prototype.toString: 'this' 为 null 或未定义

filename: utils.assetsPath('css/[name]-[part].[ext]')

filename: utils.assetsPath('css/[name].[contenthash].css'),

size: default: 4000 The maximum number of CSS rules allowed in a single file. To make things work with IE this value should be somewhere around 4000.

// Setting the following option to `false` will not extract CSS from codesplit chunks.
// Their CSS will instead be inserted dynamically with style-loader when the codesplit chunk

错 所以放到 所有引入的is的最后面引入即可 <script src="https://cdn.bootcss.com/es6-promise/4.1.1/es6-promise.auto.min.js"></script>

这个报错 是因为 你同时按 上面的123步骤后 又按9里的b 又通过cdn引入了 这样造成了冲突 所以只需要一种引入方式即可 d. ReferenceError: "Promise"未定义 SCRIPT1006: 缺少')' Error: Loading chunk 0 failed.

c. SCRIPT5022: only one instance of babel-polyfill is allowed

module: { rules: [...(config.dev.useEslint ? [createLintingRule()] : []),

上面这3个错误 都是因为 es6语法太高造成的 只需要引入babel-polyfill 然后再在build里的 webpack.base.conf.js 处理is的时候 按照报错

这个报错 是因为 你百度搜索处理 babel-polyfill 的时候 有人告诉你直接在html里引入 但是你放到了 打包后的所有js引入的前面一起的报

test: /\.vue\$/, loader: 'vue-loader', options: vueLoaderConfig

```
test: /\.js$/,
       loader: 'babel-loader?cacheDirectory',
       include: [resolve('src'), resolve('static'), resolve('test'), resolve('node_modules/webpack)
                                               https://blog.csdn.net/gaogiang1112
到此, 所有问题,终结!!!!!!!!!!!!!!!!!!
```

不过ie8 大家就不要想了 因为vue 最低兼容到ie8 如果还想往下兼容 需要再处理vue 就没有什么意义了 所以 我们处理到ie9 就可以了

如果你觉得我帮到你了,买个肥宅快乐水,怎么样?哈哈哈哈哈哈