链滴

如何优化 Vue 项目的打包速度

作者：Ailen
原文链接：https://ld246.com/article/1598344242307
来源网站：链滴

优化Vue项目的打包速度

随着项目依赖的增加, 在配合开发的时候听到后端同学的抱怨 build 时间长, 及其影响开发效率和心情. 然后翻阅了相关的资料开启Vue优化之路。

此项目是针对vue-cli2脚手架进行优化的

- 项目依赖管理

首先检查一下依赖, 去掉对项目没用的依赖, 然后注意开发依赖跟生产依赖不能弄混. 如scss, scss-loader要放在开发依赖中, 因为编译后就不需要了。

- 优化配置

修改config/index.js下的文件

```
// 开发环境dev下, 设置为`eval`能提高最快速度, 但是缺点是不能正确显示行号, Debug会有点影响
devtool: 'eval'
// 关闭生产环境的sourceMap, 不懂是啥的话可以看下面的文章
// 阮一峰 - JavaScript Source Map 详解(http://www.ruanyifeng.com/blog/2013/01/javascript_source_map.html)

productionSourceMap: false
```

** 在src/mian.js关闭生产环境下的调试信息**

```
const isDebugMode = process.env.NODE_ENV !== "production";
Vue.config.debug = isDebugMode;
Vue.config.devtools = isDebugMode;
Vue.config.productionTip = isDebugMode;
```

启动 DllPlugin

- 在build文件夹中新增webpac.dll.config.js的js文件. 我们将第三方库抽取出来, 打包dll代码.

```
const path = require("path");
const webpack = require("webpack");
// 抽取第三方库
const vendors = [
      "vue/dist/vue.common.js",
      "vue-router",
      "babel-polyfill",
      "axios",
      "element-ui",
      "mint-ui"
]
module.exports = {
  entry: {
    vendor: vendors
  },
  output: {
    path: path.join(__dirname, "../static/js"),
    filename: "[name].dll.js",
    library: "[name]_[hash]" // vendor.dll.js中暴露出的全局变量名
  },
```

原文链接：如何优化 Vue 项目的打包速度

```
  plugins: [
    new webpack.DllPlugin({
    path: path.join(__dirname, ".", "[name]-manifest.json"),
      // 此处需要和 output.library 的值一致
      name: "[name]_[hash]",
      context: __dirname
    }),
    new webpack.optimize.UglifyJsPlugin({
      compress: {
        warnings: false
      }
    })
  ]
};
```

然后再写一个快捷的调用方式, 在package.json的script上添加一行代码:

```
{
  "scripts": {
    ...此处省略其他代码
    "dll": "webpack --config ./build/webpack.dll.config.js"
  }
}
```

然后直接在命令行使用 **npm run dll**, 生成**vendor-manifest.json**和**vendor.dll.js**, 前者是库文件的ode_modle路径和webpack打包id的映射. 后者是打包后的代码库。

- 然后再安装 **html-webpack-include-assets-plugin**和**copy-webpack-plugin**这两个插件.

```
npm install --save-dev html-webpack-include-assets-plugin copy-webpack-plugin
```

- copy-webpack-plugin是因为项目的需要，需要copy到指定的目录下.
- html-webpack-include-assets-plugin是将vendor.dll.js插入到index.html里.

这里值得一提的是, 虽然我们可以直接在根目录的index.html里插入script, 但实际上并不妥当的. 当你pm run dev后, 你会发现在控制台那一行红色的报错, 告诉着你东西找不到, 虽然不影响开发, 但你会得很难受.

紧接着在webpack.prod.conf.js上引用我们添加的依赖, 再添加以下代码

```
// config 是 config/index.js 里的配置
// utils.assetsPath 也是 build/utils.js的配置
// 这里这样写是为了跟项目统一
plugins: [
  new webpack.DllReferencePlugin({
    context: __dirname,
    manifest: require('./vendor-manifest.json')
  }),
  // copy custom static assets
  new CopyWebpackPlugin([
    {
      from: path.resolve(__dirname, '../static'),
      to: config.build.assetsSubDirectory,
      ignore: ['.*']
```

```
    }
  ]),
  // 将 vendor.dll.js 插入HTML里
  new HtmlWebpackIncludeAssetsPlugin({
    assets: [utils.assetsPath('js/vendor.dll.js')],
    files: ['index.html'],
    append: false
  }),
]
```

在命令行进行打包npm run build, 可以看到构建的速度大大的提高啦.

根据我的实测，由原来的3分钟到1分钟，确实节省了很多的时间。

最后再优化时说说踩的坑, 给后来人一些提示.

Uncaught ReferenceError: vendor_library is not defined - 检查HTML里是否插入vendor.dll.js, 件是否加载成功, 或者检查是否有路径问题.

一些打包后hash不变仍然会被清除重新打包 ... 这是因为vue-cli默认配置会直接清空dist文件夹里的有文件, 你提前放一些js文件进去也是没有用的.

本文章参考博客地址：链接地址