Vue+webpack+vuerouter实现单页面应用

准备工作

1. 安装node.js
2. 全局安装webpack npm install webpack -g

webpack #最基本的启动webpack命令

webpack -w #提供watch方法，实时进行打包更新

webpack -p #对打包后的文件进行压缩

webpack -d #提供SourceMaps，方便调试

webpack --colors #输出结果带彩色，比如：会用红色显示耗时较长的步骤

webpack --profile #输出性能数据，可以看到每一步的耗时

webpack --display-modules #默认情况下 node\_modules 下的模块会被隐藏，加上这个参数可以显示这些被隐藏的模块

第一步：初始化项目目录

我们需要创建如下目录及文件夹，最终目录结构如下：

- dist //文件生成目录

-- //自动生成

- node\_module //自动安装

-- ...- src //文件入口

-- components //组件存放

-- app.vue //主.vue

-- main.js //主.js- index.html //主.html- package.json //npm 配置- webpack.cofig.js // webpack配置

第二步：安装项目依赖

如果你上面没有创建package.json文件的话，可以直接使用npm init来初始化我们的package.json文件的配置。想要去编译其他的文件比如react，coffce等等，就必须要加载很多对应的loader。要想加载一个.vue文件。当然也是同样的道理。 我的配置清单如下：

{

"name": "nianqa", //项目名称

"version": "1.0.0", //版本

"description": "vue+webapck", //描述

"main": "index.js", //主文件

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"start": "webpack-dev-server --inline"

}, //scripts指定了运行脚本命令的npm命令行缩写，比如这是的start指定了运行npm run start时，所要执行的命令。

"dependencies": { //项目依赖

"vue": "^1.0.18"

},

"devDependencies": { //各种各样的loader，用来解析想相应的文件格式。要解析vue并且完成相应的功能，这些基本都是必须的。

"autoprefixer-loader": "^2.0.0",

"babel": "^6.3.13",

"babel-core": "^6.3.21",

"babel-loader": "^6.2.0",

"babel-plugin-transform-runtime": "^6.3.13",

"babel-preset-es2015": "^6.3.13",

"babel-runtime": "^5.8.34",

"css-loader": "^0.16.0",

"file-loader": "^0.8.5",

"html-loader": "^0.3.0",

"node-sass": "^3.4.2",

"sass-loader": "^3.2.0",

"style-loader": "^0.12.3",

"url-loader": "^0.5.6",

"vue-html-loader": "^1.2.0",

"vue-loader": "^7.2.0",

"webpack": "^1.12.0",

"webpack-dev-server": "^1.14.0"

},

"author": "guowenfh", //作者

"license": "MIT", //开源协议

"keywords": [ //关键字

"vue",

"webpack"

]

}

第三步：配置webpack

文件已经设置好了，接下来就到了我们关键的一步，配置webpack.config.js,清单如下：

**var** path = require('path');  
// NodeJS中的Path对象，用于处理目录的对象，提高开发效率。  
// 模块导入  
module.exports = {  
 // 入口文件地址，不需要写完，会自动查找  
 entry: './src/main',  
 //输出位置  
 output: {  
 path: path.join(\_\_dirname, './dist'), //配置输出路径，文件地址，使用绝对路径形式  
 filename: '[name].js',  
 //关于filename 我们有个变量就是 [name] = entry的key 当然还有别的变量比如[id],[hash]等,大家可以自行发挥  
 //我们也能把filename写成 filename : [name]/[name].[name].js 也是可以的  
 //[name]这里是webpack提供的根据路口文件自动生成的名字  
 publicPath: '/dist/'  
 // 公共文件生成的地址  
 },  
 // 服务器配置相关，自动刷新!  
 devServer: {  
 historyApiFallback: **true**,  
 hot: **false**,  
 inline: **true**,  
 progress: **true**,  
 },  
 // 加载器  
 module: {  
 // 加载器  
 loaders: [  
 // 解析.vue文件  
 {  
 test: /\.vue$/,  
 loader: 'vue'  
 },  
 // 转化ES6的语法  
 {  
 test: /\.js$/,  
 loader: 'babel',  
 exclude: /node\_modules/  
 },  
 // 编译css并自动添加css前缀  
 {  
 test: /\.css$/,  
 loader: 'style!css!autoprefixer'  
 },  
 //.scss 文件想要编译，scss就需要这些东西！来编译处理  
 //install css-loader style-loader sass-loader node-sass --save-dev  
 {  
 test: /\.scss$/,  
 loader: 'style!css!sass?sourceMap'  
 },  
 // 图片转化，小于8K自动转化为base64的编码  
 {  
 test: /\.(png|jpg|gif)$/,  
 loader: 'url-loader?limit=8192'  
 /\*query: {  
 limit: 10000,  
 name: './images/[name].[ext]?[hash:8]'  
 }\*/  
 //在这无论是直接loader 后面跟参数(像url跟参一样),或者是后面跟着一个对象 query,都是可以的.类似get请求？  
 },  
 //html模板编译？  
 {  
 test: /\.(html|tpl)$/,  
 loader: 'html-loader'  
 },  
 ]  
 },  
 // .vue的配置。需要单独出来配置  
 vue: {  
 loaders: {  
 css: 'style!css!autoprefixer'  
 }  
 },  
 // 配置babel转化成es5的语法  
 babel: {  
 presets: ['es2015'],  
 plugins: ['transform-runtime']  
 },  
 // plugins，还没用到  
 // 这里我就只用到一个就是生成 独立的css文件,style嵌套在页面里的方式实在是丑得不行  
 resolve: {  
 // require时省略的扩展名，如：require('module') 不需要module.js  
 extensions: ['', '.js', '.vue'],  
 // 别名，可以直接使用别名来代表设定的路径以及其他  
 alias: {  
 filter: path.join(\_\_dirname, './src/filters'),  
 components: path.join(\_\_dirname, './src/components')  
 }  
 },  
 // 开启source-map调试模式，webpack有多种source-map，在官网文档可以查到  
 devtool: 'eval-source-map'  
};