使用webpack构建本地服务器

1. 什么是webpack

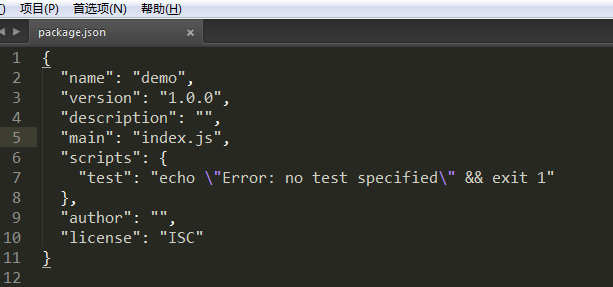
Webpack可以看做是模块打包机：它做的事情是，分析你的项目结构，找到javascript模块以及其它的一些浏览器不能直接运行的拓展语言(TypeScript等)，并将其转换和打包为合适的格式供浏览器使用。

二、正式使用webpack前的准备工作：

1、在练习文件夹中创建一个package.json文件，这是一个标准的npm说明文件，里面蕴含了丰富的信息，包括当前项目的依赖模块，自定义的脚本任务等等。在终端中使用npm init命令可以自动创建这个package.json文件

npm init

输入这个命令后，终端会问你一系列诸如项目名称，项目描述，作者等信息，不过不用担心，如果你不准备在npm中发布你的模块，这些都不重要，回车默认就可以了。



2、package.json文件已经就绪，我们在本项目中安装webpack作为依赖包；

*// 安装Webpack*

npm install --save-dev webpack

安装webpack开发工具，简单来说，webpack-dev-server就是一个小型的文件服务器，使用它，可以为webpack打包生成的资源文件提供web服务。

npm install --save-dev webpack-dev-server

然后在package.json中的scripts对象中添加如下命令，用以开启本地服务器：

"scripts": {

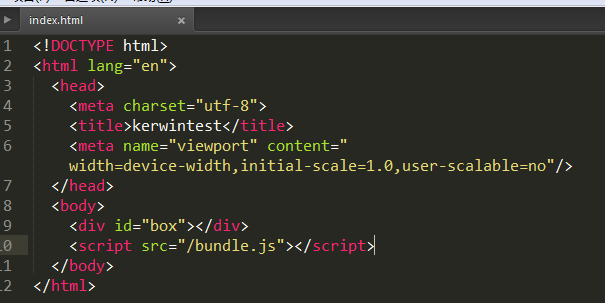
"test": "echo \"Error: no test specified\" && exit 1",

"start": "webpack",

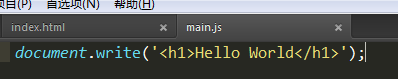
"server": "webpack-dev-server --open"

},

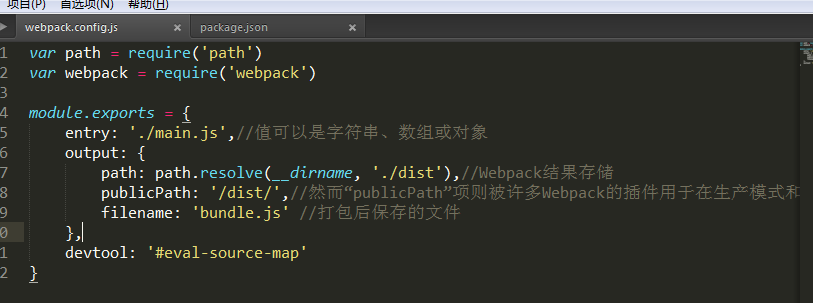
3、新增一个index.html文件



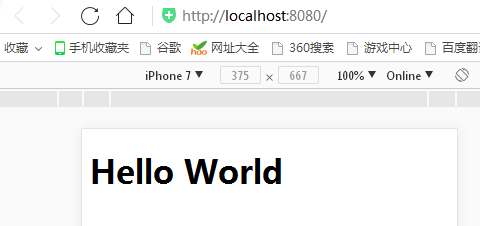
然后创建一个入口文件main.js，名字可以随便起



最后创建配置文件webpack.config.js，webpack要配置的信息都会保存在这个文件。



文件创建后在命令后执行webpack-dev-server或者npm run server，然后打开了<http://localhost:8080/>就能看到效果了，如下图：



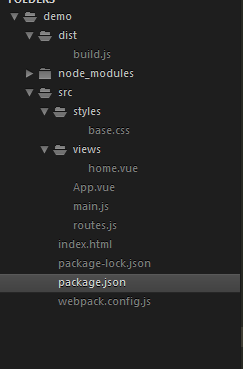
三、完整项目构建开发打包流程

★**安装项目依赖**



★**编辑项目目录以及添加代码**

1、文件目录如下



//dist文件是后面执行webpack指令生产的，不用管；

//webpack.config.js 配置文件，本身也是一个标准的Commonjs规范的模块；

//routes.js文件放路由配置文件；

//index.html首页入口文件

//App.vue是项目入口文件。

//main.js这是项目的核心文件。全局的配置都在这个文件里面配置。

//views文件放首页或者其他页面；

* **代码**

//webpack.config.js

var path = require('path')

var webpack = require('webpack')

module.exports = {

entry: './src/main.js',//值可以是字符串、数组或对象

output: {

path: path.resolve(\_\_dirname, './dist'),//Webpack结果存储

publicPath: '/dist/',

filename: 'build.js'

},

module: {

rules: [

{

test: /\.vue$/,

loader: 'vue-loader',

options: {

loaders: {}

}

},

{

test: /\.js$/,

loader: 'babel-loader',

exclude: /node\_modules/

},

{

test: /\.(png|jpg|gif|svg)$/,

loader: 'file-loader',

options: {

name: '[name].[ext]?[hash]'

}

},

{

test: /\.css$/,

loader: "style-loader!css-loader"

},

{

test: /\.scss$/,

loader: "style-loader!css-loader!sass-loader!"

}

]

},

resolve: {

alias: {

'vue$': 'vue/dist/vue.esm.js'

}

},

devServer: {//webpack-dev-server配置

historyApiFallback: true,//不跳转

noInfo: true,

inline: true//实时刷新

},

performance: {

hints: false

},

devtool: '#eval-source-map'

}

if (process.env.NODE\_ENV === 'production') {

module.exports.devtool = '#source-map'

module.exports.plugins = (module.exports.plugins || []).concat([

new webpack.DefinePlugin({

'process.env': {

NODE\_ENV: '"production"'

}

}),

new webpack.optimize.UglifyJsPlugin({

sourceMap: true,

compress: {

warnings: false

}

}),

new webpack.LoaderOptionsPlugin({

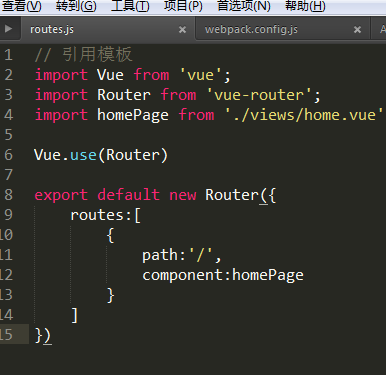
minimize: true

})

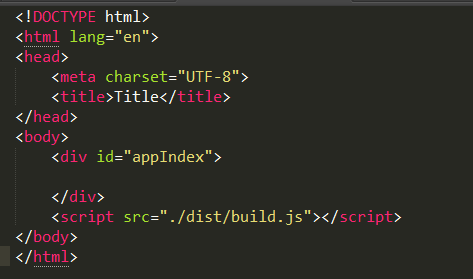
])

}

//routes.js



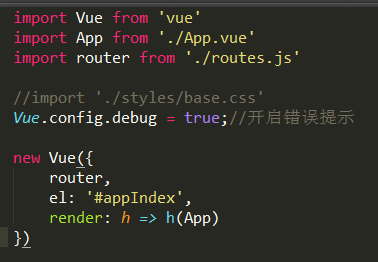
//index.html



//App.vue



//main.js

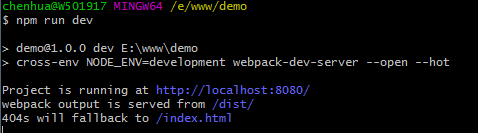


//views/home.vue文件放详情页面；

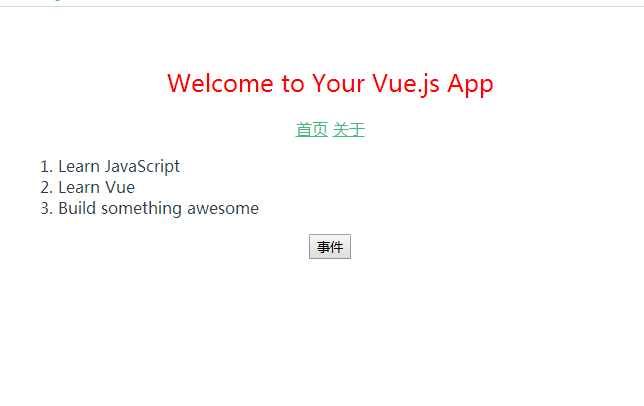


★**项目跑起来**

执行指令：npm run dev



浏览器打开生成的链接，如上图



综上总结：除了用vue-cli脚手架进行搭建之外，可以进行手把手搭建一个属于自己的vue项目，并且进行本地服务器配置。

文档api说明：https://www.webpackjs.com/concepts/