Webpack的使用二

**一、webpack中的plugin**

plugin意思就是插件，我们可以通过plugin给webpack添加符加的功能。

**二、plugin体验**

在这里我们来体验一个简单的plugin——版权插件。这个插件可以帮助我们在bundle.js的头部插入一些我们需要的信息。方法比较简单，只需要在webpack的配置文件里添加一些代码即可。



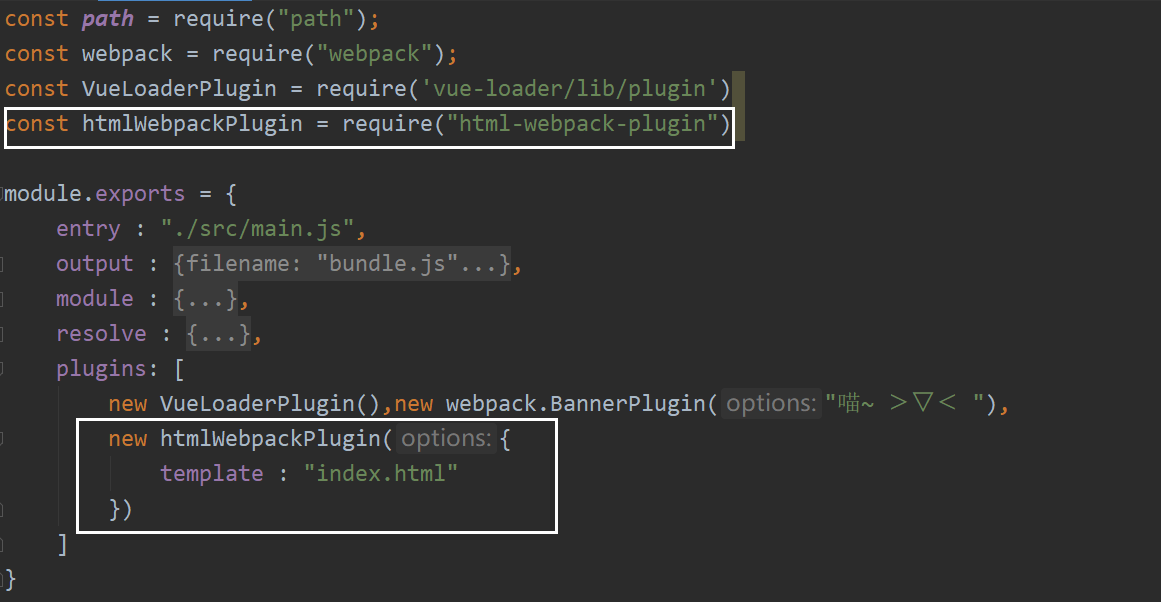
**三、HtmlWebpackPlugin插件**

我们在发布真实项目的时候，发布的项目是放在dist文件夹中的，但是dist项目中并没有index.html文件。所以我们需要把index.html文件打包放入dist文件夹中就可以使用到HtmlWebpackPlugin插件。

HtmlWebpackPlugin插件可以帮助我们生成一个index.html文件，然后将打包好的js文件自动的插入到body标签之中。

安装命令：**npm install html-webpack-plugin@3.2.0 –-save-dev**

Webpack配置文件相关修改如下：



如果有同学之前在output属性中配置了publicPath属性，记得删除。不然会导致路径出现问题，导出网页中的图片无法正常显示。

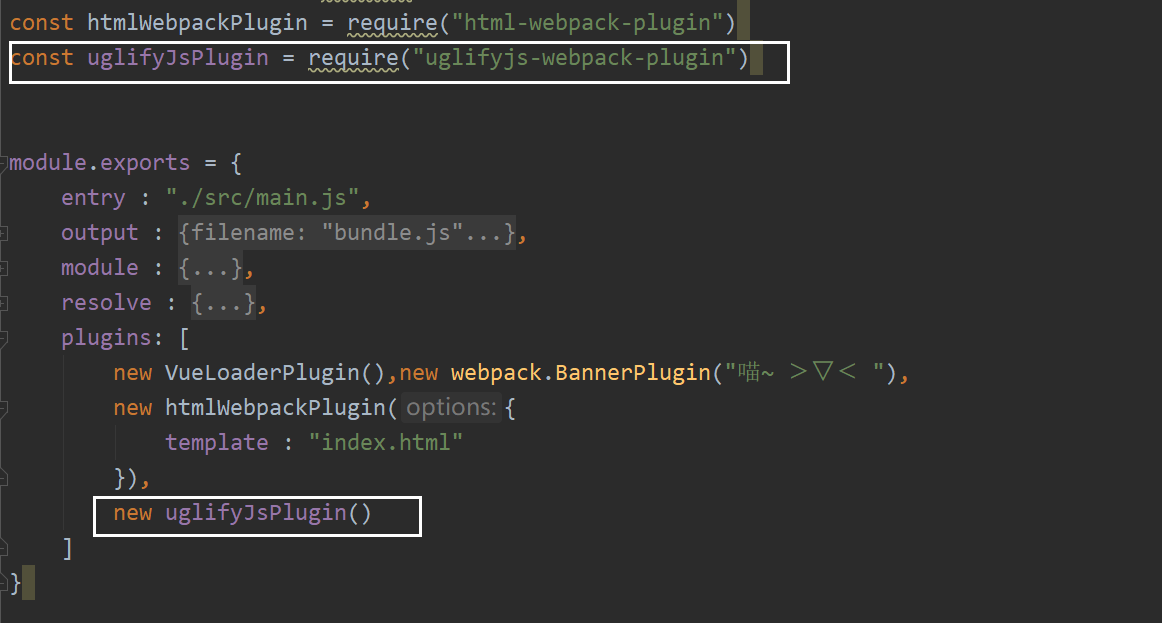
默认生成的文件模板“index.html”来自与src目录中的“index.html”。html-webpack-plugin插件会在模板的基础上导入js生成最后的入口html文件。所以我们借由模板文件来控制最终生成html文件中的内容。

**四、uglifyjs-webpack-plugin插件**

该插件可以对JS文件进行压缩。

安装命令：npm install uglifyjs-webpack-plugin@1.1.1 --save-dev

Webpack配置：



安装完成并且配置之后，重新打包JS文件发现JS文件已经被压缩了。

**五、server服务器搭建**

Webpack可以搭建一个基于node的服务器，内部是express框架。可以实现让浏览器自动刷新显示我们修改完成之后的结果。

安装命令：**npm install** [**webpack-dev-server@2.9.3**](mailto:webpack-dev-server@2.9.3) **--save-dev**

**devserver也是webpack的一个选项，他可以设置以下属性：**

contentBase：指定为哪个文件夹提供本地服务（一般设置为./dist）

port：端口号（默认8080）

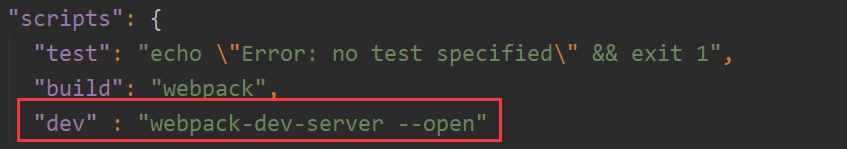
inline：页面事实刷新（true为实时刷新）

historyApiFallback：在SPA页面中，依赖HTML5的history模式

我们首先需要把devserver属性添加到webpack的配置文件。



并且在package.json中新添加一个script，以便于我们启动服务器。



此时，在命令行行输入：npm run dev就可以把我们的本地服务器给运行起来了。