[html-webpack-plugin用法全解](https://segmentfault.com/a/1190000007294861)

* [javascript](https://segmentfault.com/t/javascript/blogs)

* [webpack](https://segmentfault.com/t/webpack/blogs)

[**499311496**](https://segmentfault.com/u/499311496) 2016年10月27日发布

* 赞  |   34收藏  |  119
* 29k 次浏览

本文只在个人博客和 SegmentFault 社区个人专栏发表，转载请注明出处   
个人博客: [https://zengxiaotao.github.io](https://zengxiaotao.github.io/)   
SegmentFault 个人专栏: [https://segmentfault.com/blog...](https://segmentfault.com/blog/zengxiaotao)

html-webpack-plugin 可能用过的 webpack 的童鞋都用过这个 plugin ,就算没用过可能也听过。我们在学习webpack的时候，可能经常会看到这样的一段代码。

// webpack.config.js

module.exports = {

entry: path.resolve(\_\_dirname, './app/index.js'),

output:{

path: path.resolve(\_\_dirname, './build'),

filename: 'bundle.js'

}

...

plugins: [

**new** HtmlWebpackPlugin()

]

}

之后在终端输入 webpack 命令后

webpack

你会神奇的看到在你的 build 文件夹会生成一个 index.html 文件和一个 bundle.js 文件，而且 index.html 文件中自动引用 webpack 生成的 bundle.js 文件。

所有的这些都是 html-webpack-plugin 的功劳。它会自动帮你生成一个 html 文件，并且引用相关的 assets 文件(如 css, js)。

自己在六月第一次接触前端自动化构建，学习 webpack 和 react 时，曾经简单使用过这个插件，但也只是用了常见的几个选项，今天就跟着官方文档走一走，看看它的所有用法。

title

顾名思义，设置生成的 html 文件的标题。

filename

也没什么说的，生成 html 文件的文件名。默认为 index.html.

template

根据自己的指定的模板文件来生成特定的 html 文件。这里的模板类型可以是任意你喜欢的模板，可以是 html, jade, ejs, hbs, 等等，但是要注意的是，使用自定义的模板文件时，需要提前安装对应的 loader， 否则webpack不能正确解析。以 jade 为例。

npm install jade-loader --save-dev

// webpack.config.js

...

loaders: {

...

{

test: /\.jade$/,

loader: 'jade'

}

}

plugins: [

**new** HtmlWebpackPlugin({

...

jade: 'path/to/yourfile.jade'

})

]

最终在build文件夹内会生成一个 yourfile.html 和 bundle.js 文件。现在我们再回头来看看之前将的 title 属性。

如果你既指定了 template 选项，又指定了 title 选项，那么webpack 会选择哪一个？ 事实上，这时候会选择你指定的模板文件的 title, **即使你的模板文件中未设置 title**。

那么 filename 呢，是否也会覆盖，其实不是，会以指定的 filename 作为文件名。

inject

注入选项。有四个选项值 true, body, head, false.

* true
  + 默认值，script标签位于html文件的 body 底部
* body
  + 同 true
* head
  + script 标签位于 head 标签内
* false
  + 不插入生成的 js 文件，只是单纯的生成一个 html 文件

favicon

给生成的 html 文件生成一个 favicon。属性值为 favicon 文件所在的路径名。

// webpack.config.js

...

plugins: [

**new** HtmlWebpackPlugin({

...

favicon: 'path/to/yourfile.ico'

})

]

生成的 html 标签中会包含这样一个 link 标签

<link rel="shortcut icon" href="example.ico">

同 title 和 filename 一样，如果在模板文件指定了 favicon，会忽略该属性。

minify

minify 的作用是对 html 文件进行压缩，minify 的属性值是一个压缩选项或者 false 。默认值为false, 不对生成的 html 文件进行压缩。来看看这个压缩选项。

html-webpack-plugin 内部集成了 [html-minifier](https://github.com/kangax/html-minifier#options-quick-reference) ,这个压缩选项同 html-minify 的压缩选项完全一样，  
看一个简单的例子。

// webpack.config.js

...

plugins: [

**new** HtmlWebpackPlugin({

...

minify: {

removeAttributeQuotes: true // 移除属性的引号

}

})

]

<!-- 原html片段 -->

<div id="example" class="example">test minify</div>

<!-- 生成的html片段 -->

<div id=example class=example>test minify</div>

hash

hash选项的作用是 给生成的 js 文件一个独特的 hash 值，该 hash 值是该次 webpack 编译的 hash 值。默认值为 false 。同样看一个例子。

// webpack.config.js

plugins: [

**new** HtmlWebpackPlugin({

...

hash: true

})

]

<script type=text/javascript src=bundle.js?22b9692e22e7be37b57e></script>

执行 webpack 命令后，你会看到你的生成的 html 文件的 script 标签内引用的 js 文件，是不是有点变化了。  
bundle.js 文件后跟的一串 hash 值就是此次 webpack 编译对应的 hash 值。

$ webpack

Hash: 22b9692e22e7be37b57e

Version: webpack 1.13.2

cache

默认值是 true。表示只有在内容变化时才生成一个新的文件。

showErrors

showErrors 的作用是，如果 webpack 编译出现错误，webpack会将错误信息包裹在一个 pre 标签内，属性的默认值为 true ，也就是显示错误信息。

chunks

chunks 选项的作用主要是针对多入口(entry)文件。当你有多个入口文件的时候，对应就会生成多个编译后的 js 文件。那么 chunks 选项就可以决定是否都使用这些生成的 js 文件。

chunks 默认会在生成的 html 文件中引用所有的 js 文件，当然你也可以指定引入哪些特定的文件。

看一个小例子。

// webpack.config.js

entry: {

index: path.resolve(\_\_dirname, './src/index.js'),

index1: path.resolve(\_\_dirname, './src/index1.js'),

index2: path.resolve(\_\_dirname, './src/index2.js')

}

...

plugins: [

**new** HtmlWebpackPlugin({

...

chunks: ['index','index2']

})

]

执行 webpack 命令之后，你会看到生成的 index.html 文件中，只引用了 index.js 和 index2.js

...

<script type=text/javascript src=index.js></script>

<script type=text/javascript src=index2.js></script>

而如果没有指定 chunks 选项，默认会全部引用。

excludeChunks

弄懂了 chunks 之后，excludeChunks 选项也就好理解了，跟 chunks 是相反的，排除掉某些 js 文件。 比如上面的例子，其实等价于下面这一行

...

excludeChunks: ['index1.js']

chunksSortMode

这个选项决定了 script 标签的引用顺序。默认有四个选项，'none', 'auto', 'dependency', '{function}'。

* 'dependency' 不用说，按照不同文件的依赖关系来排序。
* 'auto' 默认值，插件的内置的排序方式，具体顺序这里我也不太清楚...
* 'none' 无序？ 不太清楚...
* {function} 提供一个函数？但是函数的参数又是什么? 不太清楚...

如果有使用过这个选项或者知道其具体含义的同学，还请告知一下。。。

xhtml

一个布尔值，默认值是 false ，如果为 true ,则以兼容 xhtml 的模式引用文件。

总结

以上，就总结完了传入 new HtmlWebpackPlugin() 的选项，了解全部选项的含义后，可以在项目构建时更灵活的使用。