

插件 `html-webpack-plugin` 的详解

[webpack](#) 阅读约 6 分钟

最近在学习webpack，接触到的第一个插件就是`html-webpack-plugin`，那么今天就来详解一下它的用法吧。

- 先来上一个例子：

```
var htmlWebpackPlugin = require('html-webpack-plugin')

const path = require('path')
module.exports = {
  entry: './src/script/main.js',
  output: {
    filename: 'js/bundle.js',
    path: path.resolve(__dirname, 'dist')
  },
  plugins: [
    new htmlWebpackPlugin({
      filename: 'index.html',
      template: 'index.html',
      inject: 'head'
    })
  ]
}
```

配置属性

title

生成html文件的标题

filename

就是html文件的文件名，默认是index.html

template

指定你生成的文件所依赖哪一个html文件模板，模板类型可以是html、jade、ejs等。但是要注意的是，如果想使用自定义的模板文件的时候，你需要安装对应的loader哦。

举例子：

```
$ npm install jade-loader --save-dev
```

```
// webpack.config.js
...
loaders: {
  ...
  {
    test: /\.jade$/,
    loader: 'jade'
  }
}
plugins: [
  new HtmlWebpackPlugin({
    ...
    jade: 'path/to/yourfile.jade'
  })
]
```

如果你设置的 `title` 和 `filename` 于模板中发生了冲突，那么以你的 `title` 和 `filename` 的配置值为准。

inject

inject有四个值： `true` `body` `head` `false`

`true` 默认值， `script` 标签位于html文件的 `body` 底部

`body` `script` 标签位于html文件的 `body` 底部

`head` `script` 标签位于html文件的 `head` 中

`false` 不插入生成的js文件，这个几乎不会用到的

favicon

给你生成的html文件生成一个 `favicon` ,值是一个路径

```
plugins: [
  new HtmlWebpackPlugin({
    ...
    favicon: 'path/to/my_favicon.ico'
  })
]
```

然后再生成的html中就有了一个 `link` 标签

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

minify

使用minify会对生成的html文件进行压缩。默认是false。html-webpack-plugin内部集成了 `html-minifier`,因此，还可以对minify进行配置：（注意，虽然minify支持BooleanObject,但是不能直接这样写： `minify: true` ,这样会报错 `ERROR in TypeError: Cannot use 'in' operator to search for 'html5' in true` ,使用时必须给定一个 `{ }` 对象）

```
...
plugins: [
  new HtmlWebpackPlugin({
    ...
    minify: {
      removeAttributeQuotes: true // 移除属性的引号
    }
  })
]
```

cache

默认是true的，表示内容变化的时候生成一个新的文件。

showErrors

当webpack报错的时候，会把错误信息包裹再一个

```
中，默认是true。
```

chunks

chunks主要用于多入口文件，当你有多个入口文件，那就回编译后生成多个打包后的文件，那么chunks 就能选择你要使用那些js文件

```
entry: {
  index: path.resolve(__dirname, './src/index.js'),
  devor: path.resolve(__dirname, './src/devor.js'),
  main: path.resolve(__dirname, './src/main.js')
}

plugins: [
  new httpWebpackPlugin({
    chunks: ['index', 'main']
  })
]
```

那么编译后：

```
<script type=javascript src=index.js></script>
<script type=javascript src=main.js></script>
```

- 如果你没有设置chunks选项，那么默认是全部显示

excludeChunks

排除掉一些js

```
excludeChunks: ['devor.js']
// 等价于上面的
```

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。

chunksSortMode

script的顺序，默认四个选项： 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....

'none' 无序？

{function} 提供一个函数？

阅读 44.8k • 更新于 2018-03-21

👍 赞 78

🔖 收藏 72

🔗 分享

本作品系 原创， 采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



Meils

👥 1.4k

关注作者