

导航

博客园
首页
新随笔
联系
订阅 **X**ML
管理

< 2016年10月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

公告

昵称： 冠军
园龄： 10年7个月
荣誉： 推荐博客
粉丝： 2107
关注： 25
[+加关注](#)

统计

随笔 - 304
文章 - 1
评论 - 1909
引用 - 0

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与

webpack 插件： **html-webpack-plugin**

插件地址：<https://www.npmjs.com/package/html-webpack-plugin>

这个插件用来简化创建服务于 webpack bundle 的 HTML 文件，尤其是对于在文件名中包含了 hash 值，而这个值在每次编译的时候都发生变化的情况。你既可以让这个插件来帮助你自动生成 HTML 文件，也可以使用 lodash 模板加载生成的 bundles，或者自己加载这些 bundles。


Installation


使用 npm 安装这个插件

```
$ npm install html-webpack-plugin@2 --save-dev
```

Basic Usage

这个插件可以帮助生成 HTML 文件，在 body 元素中，使用 script 来包含所有你的 webpack bundles，只需要在你的 webpack 配置文件中如下配置：

```
  
var HtmlWebpackPlugin = require('html-webpack-plugin')  
var webpackConfig = {  
  entry: 'index.js',  
  output: {  
    path: 'dist',  
    filename: 'index_bundle.js'  
  },  
  plugins: [new HtmlWebpackPlugin()]  
}
```

```

```

最新评论
我的标签

我的标签

jQuery(4)
nodejs(3)
spring.net nhibernate(3)
javascript(3)
ibatis mybatisnet orm(2)
java Andriod(1)
nhibernate spring.net(1)
wf4(1)


随笔分类

.Net(71)
Angular 2(2)
AngularJS(27)
ASP.NET(26)
ASP.NET MVC(48)
Azure
Bootstrap(19)
C++(3)
cocos2d-x(6)
Database(3)
EF(23)
Java(1)
javascript(95)
jQuery(17)
Kinect
mybatisnet(5)
NHibernate(7)
Nodejs(10)
Orchard(1)
Raspberry Pi 树莓派(1)
SharePoint(16)
SilverLight(2)
Spring.NET(16)
SQL Server
Tools(6)
Web Front(34)
Windows 8(8)
Windows Azure(6)
Windows Phone(10)
XML(6)
面试题(3)


随笔档案

2016年9月 (2)
2016年7月 (2)
2016年6月 (1)
2016年5月 (1)
2016年3月 (1)
2016年2月 (7)
2016年1月 (1)
2015年12月 (5)

这将会自动在 `dist` 目录中生成一个名为 `index.html` 的文件，内容如下：



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Webpack App</title>
  </head>
  <body>
    <script src="index_bundle.js"></script>
  </body>
</html>
```



如果你有多个 `webpack` 入口点，它们都会被包含在生成的 `script` 元素中。


如果有任何的 `CSS` 资源包含在 `webpack` 输出中（例如，使用 `ExtractTextPlugin` 提炼出的 `css`），这些将会使用 `link` 包含在 `HTML` 页面的 `head` 元素中。

Configuration


可以进行一系列的配置，支持如下的配置信息

- `title`: 用来生成页面的 `title` 元素
- `filename`: 输出的 `HTML` 文件名，默认是 `index.html`，也可以直接配置带有子目录。
- `template`: 模板文件路径，支持加载器，比如 `html!./index.html`
- `inject`: `true` | `'head'` | `'body'` | `false`，注入所有的资源到特定的 `template` 或者 `templateContent` 中，如果设置为 `true` 或者 `body`，所有的 `javascript` 资源将被放置到 `body` 元素的底部，`'head'` 将放置到 `head` 元素中。
- `favicon`: 添加特定的 `favicon` 路径到输出的 `HTML` 文件中。
- `minify`: `{}` | `false`，传递 `html-minifier` 选项给 `minify` 输出
- `hash`: `true` | `false`，如果为 `true`，将添加一个唯一的 `webpack` 编译 `hash` 到所有包含的脚本和 `CSS` 文件，对于解除 `cache` 很有用。
- `cache`: `true` | `false`，如果为 `true`，这是默认值，仅仅在文件修改之后才会发布文件。
- `showErrors`: `true` | `false`，如果为 `true`，这是默认值，错误信息会写入到 `HTML` 页面中
- `chunks`: 允许只添加某些块（比如，仅仅 `unit test` 块）
- `chunksSortMode`: 允许控制块在添加到页面之前的排序方式，支持的值: `'none'` | `'default'` | `{function}-default:'auto'`
- `excludeChunks`: 允许跳过某些块，（比如，跳过单元测试的块）

下面的示例演示了如何使用这些配置。




```
{
  entry: 'index.js',
  output: {
    path: 'dist',
    filename: 'index_bundle.js',
    hash: true
  },
  plugins: [
    new HtmlWebpackPlugin({
      title: 'My App',
      filename: 'assets/admin.html'
    })
  ]
}
```




生成多个 HTML 文件

通过在配置文件中添加多次这个插件，来生成多个 HTML 文件。



```
{
  entry: 'index.js',
  output: {
    path: 'dist',
    filename: 'index_bundle.js'
  },
  plugins: [
    new HtmlWebpackPlugin(), // Generates default index.html
    new HtmlWebpackPlugin({ // Also generate a test.html
      filename: 'test.html',
      template: 'src/assets/test.html'
    })
  ]
}
```



编写自定义模板

如果默认生成的 HTML 文件不适合你的需要看，可以创建自己定义的模

[2010年8月 \(8\)](#)
[2010年7月 \(4\)](#)
[2010年6月 \(6\)](#)
[2010年5月 \(5\)](#)
[2010年4月 \(8\)](#)
[2010年3月 \(2\)](#)
[2009年4月 \(2\)](#)
[2008年10月 \(1\)](#)
[2008年9月 \(1\)](#)
[2008年7月 \(2\)](#)
[2008年6月 \(3\)](#)
[2008年3月 \(1\)](#)
[2006年9月 \(1\)](#)
[2006年2月 \(4\)](#)

[相册](#)

[integrated-security\(10\)](#)
[jQuery_DataTables\(4\)](#)
[sharepoint\(8\)](#)
[sharepoint2010-smtp\(3\)](#)
[sharepoint-excel-import\(9\)](#)
[sharepoint-profile\(4\)](#)
[我的图片\(8\)](#)

.NET

[《ASP.NET 本质论》源码下载](#)
[KevinYang](#)
[Scott Gu 博客中文版](#)
[Scott Gu 的博客](#)
[Scott Hanselman's](#)
[ComputerZen.com](#)
[WuCountry](#)
[北京 .net 俱乐部](#)
[张子秋的博客](#)

Java

[Java的天空](#)

最新评论

[1. Re:cocos2d-x 中添加显示文字的三种方式 LabelTTF 、LabelBMFont 和 LabelAtlas 不错](#)

--arture

[2. Re:webpack 插件: html-webpack-plugin](#)
 请问模板文件里怎么引入配置里的title啊?

--yilinkk

[3. Re:集成 Kendo UI for Angular 2 控件](#)
 Kendo 很早就开始了, 一直在等 Angular 2.

--冠军

[4. Re:集成 Kendo UI for](#)

板。方便的方式是通过 inject 选项, 然后传递给定制的 HTML 文件。html-webpack-plugin 将会自动注入所有需要的 CSS, js, manifest 和 favicon 文件到标记中。



```

plugins: [
  new HtmlWebpackPlugin({
    title: 'Custom template',
    template: 'my-index.html', // Load a custom template
    inject: 'body' // Inject all scripts into the body
  })
]
  
```



my-index.html 文件



```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
    <title><%= htmlWebpackPlugin.options.title %></title>
  </head>
  <body>
  </body>
</html>
  
```



如果你有模板加载器, 可以使用它来解析这个模板。



```

module: {
  loaders: [
    { test: /\.hbs$/, loader: "handlebars" }
  ],
},
plugins: [
  new HtmlWebpackPlugin({
    title: 'Custom template using Handlebars',
    template: 'my-index.hbs',
    inject: 'body'
  })
]
  
```



Angular 2 控件

支持的这么快。Angular2出来，立马也发布出来了。

--快乐八哥

5. Re:MusicStore 项目的完整配套视频!

冠军老师，你这个视频后面怎么能看了啊，望分享。

--听风者之风云再起

阅读排行榜

1. jQuery UI dialog 的使用 (133090)
2. jQuery 中使用 JSON(102582)
3. 使用 jQuery dataTables - 2 四种数据来源(51136)
4. Entity Framework 4.1 之一: 基础(38558)
5. 使用 Bootstrap Typeahead 组件(38308)

评论排行榜

1. ASP.NET MVC 音乐商店 - 4. 数据访问(81)
2. ASP.NET MVC 音乐商店 - 8. 使用 Ajax 更新的购物车(60)
3. 《ASP.NET 本质论》源码下载(50)
4. 在 MVC4 中使用 Uploadify 3.2 - 1(43)
5. Contoso 大学 - 1 - 为 ASP.NET MVC 应用程序创建 EF 数据模型(43)

推荐排行榜

1. 不畏浮云遮望眼，只缘身在最高峰(39)
2. ASP.NET MVC 3 概述(29)
3. Entity Framework 4.1 之一: 基础(28)
4. ASP.NET MVC 音乐商店 - 目录(25)
5. 创建 Web 前端开发环境(24)

另外，如果你的模式是一个字符串，可以使用 `templateContent` 传递它。

```
plugins: [
  new HtmlWebpackPlugin({
    inject: true,
    templateContent: templateContentString
  })
]
```

如果 `inject` 特性不适合你的需要，你希望完全控制资源放置。可以直接使用 `lodash` 语法，使用 `default template` 作为起点创建自己的模板。

`templateContent` 选项也可以是一个函数，以便使用其它语言，比如 `jade`:

```
plugins: [
  new HtmlWebpackPlugin({
    templateContent: function(templateParams, compilation) {
      // Return your template content synchronously here
      return '..';
    }
  })
]
```

或者异步版本

```
plugins: [
  new HtmlWebpackPlugin({
    templateContent: function(templateParams, compilation,
      callback) {
      // Return your template content asynchronously here
      callback(null, '..');
    }
  })
]
```

注意，如果同时使用 `template` 和 `templateContent`，插件会抛出错误。

变量 `o` 在模板中是在渲染时传递进来的数据，这个变量有如下的属性：

- `htmlWebpackPlugin`: 这个插件的相关数据

- **htmlWebpackPlugin.files:** 资源的块名，来自 webpack 的 stats 对象，包含来自 entry 的从 entry point name 到 bundle 文件名映射。



```
"htmlWebpackPlugin": {
  "files": {
    "css": [ "main.css" ],
    "js": [ "assets/head_bundle.js",
"assets/main_bundle.js"],
    "chunks": {
      "head": {
        "entry": "assets/head_bundle.js",
        "css": [ "main.css" ]
      },
      "main": {
        "entry": "assets/main_bundle.js",
        "css": []
      }
    }
  }
}
```



如果在 webpack 配置文件中，你配置了 `publicPath`，将会反射正确的资源

- **htmlWebpackPlugin.options:** 传递给插件的配置。
- **webpack:** webpack 的 stats 对象。
- **webpackConfig:** webpack 配置信息。

过滤块

可以使用 `chunks` 来限定特定的块。

```
plugins: [
  new HtmlWebpackPlugin({
    chunks: ['app']
  })
]
```

还可以使用 `excludeChunks` 来排除特定块。

```
plugins: [
  new HtmlWebpackPlugin({
    excludeChunks: ['dev-helper']
  })
]
```

事件

通过事件允许其它插件来扩展 HTML。

- `html-webpack-plugin-before-html-processing`
- `html-webpack-plugin-after-html-processing`
- `html-webpack-plugin-after-emit`

使用方式:

```
compilation.plugin('html-webpack-plugin-before-html-processing', function(htmlPluginData, callback) {
  htmlPluginData.html += 'The magic footer';
  callback();
});
```

分类: [Web Front](#), [AngularJS](#), [Bootstrap](#), [javascript](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

冠军

关注 - 25

粉丝 - 2107

荣誉: [推荐博客](#)

[+加关注](#)

7

0

« 上一篇: [How I came to find Linux](#)

» 下一篇: [翻译: TypeScript 1.8 Beta 发布](#)

posted on 2016-01-26 21:41 [冠军](#) 阅读(14648) 评论(12) [编辑](#) [收藏](#)

评论

#1楼 2016-07-05 18:04 馮淋

请问, 想给一个页面头部插入一个 js , 底部插入 n 个 js, 怎么搞?

[支持\(0\)](#) [反对\(0\)](#)

#2楼[楼主] 2016-07-07 12:34 冠军

可以写一个插件, 使用 `html-webpack-plugin` 的各种事件来处理。我会再写一篇博客介绍一下。

[支持\(0\)](#) [反对\(0\)](#)

#3楼[楼主] 2016-07-08 16:37 冠军

写好了, 可以[到这里查看](#)。

[支持\(0\)](#) [反对\(0\)](#)

#4楼 2016-07-12 15:48 YoungKind

webpack的`HtmlWebpackPlugin`这个插件如果网站有十几个页面, 按照你博客里的写法岂不是得在配置文件里写下面这段配置写上十几段, 有什么优化的方式吗

```
new HtmlWebpackPlugin({ // Also generate a test.html
  filename: 'X.html',
```

```
template: 'src/assets/X.html'  
})
```

支持(0) 反对(0)

#5楼[楼主] 2016-07-12 18:05 冠军

@ YoungKind

它主要用在单页应用中，你应该不会有那么多页面的。
很多的页面实际上是在单页应用中的页面，并不是独立的页面。

支持(0) 反对(0)

#6楼 2016-07-13 09:22 YoungKind

@ 冠军

想到解决办法了，写个函数，返回值是数组，功能便是遍历有几个入口文件，对应的那段配置就生成几次，瞬间配置文件就简洁多了，webpack感觉不太适合多页面应用

支持(0) 反对(0)

#7楼 2016-07-27 11:34 冷子欲

@ YoungKind

请问下最后你怎么解决的??

支持(0) 反对(0)

#8楼 2016-07-27 11:41 冷子欲

@ YoungKind

还有我发现每个页面的js、css都加载进来了，咋办

支持(0) 反对(0)

#9楼 2016-07-28 17:06 acjum

请教各位大神，
HtmlWebpackPlugin 通过提供模板输出的结果.html页面如何做到实时编译呢？

不能每次修改了模板文件，都需要 npm run build 啊~~?? ~

支持(0) 反对(0)

#10楼[楼主] 2016-07-29 15:50 冠军

@ acjum

配合 webpack-dev-server, 可以监控文件的修改。
另外，模板文件应该基本不需要改动的。

支持(0) 反对(0)

#11楼 2016-07-29 18:02 acjum

@ 冠军

非常感谢，

支持(0) 反对(0)

#12楼 2016-09-20 18:28 yilinkk

请问模板文件里怎么引入配置里的title啊？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【活动】优达学城正式发布“无人驾驶车工程师”课程

【推荐】移动直播百强八成都在用融云即时通讯云

【推荐】别再闷头写代码！找对工具，事半功倍，全能开发工具包用起来



最新IT新闻:

- [Oculus Touch](#)今起可预订，宣传片展现的功能看着也让人兴奋
 - [苹果三星专利大战](#)今日再次开庭，你想知道的都在这里
 - [拥有25万付费用户的Byju's](#)，何以拿到扎克伯格的亚洲首投？
 - [苹果开始调查两起iPhone 6s Plus起火事件](#)
 - [彻底颠覆认知：这么锻炼3分钟等于健身房3小时](#)
- » [更多新闻...](#)



最新知识库文章:

- [陈皓：什么是工程师文化？](#)
 - [没那么难，谈CSS的设计模式](#)
 - [程序猿媳妇儿注意事项](#)
 - [可是姑娘，你为什么要编程呢？](#)
 - [知其所以然（以算法学习为例）](#)
- » [更多知识库文章...](#)

Powered by:

[博客园](#)

Copyright © 冠军