链滴

webpack 从入门到放弃

作者：jiangqiang96

原文链接：https://ld246.com/article/1571974407195

来源网站：链滴

Webpack 是一个前端的静态模块资源打包工具，能让浏览器也支持模块化。它将根据模块的依关系进行静态分析，然后将这些模块按照指定的规则生成对应的静态资源。
<h3 id="常用命令">常用命令</h3>
<pre><code class="highlight-chroma">webpack -v 查看版本号
webpack ./js/main.s -o ./js/bundle.js 第一个路径为入口js的路径，第二个路径参数为生成文件的路径
</code></pre>
使用配置文件之后，直接使用 <code>webpack</code> 命令，不需要参数也可以打包，配置件的名字为“webpack.config.js”。
下面是一个完整配置的内容：
<pre><code class="highlight-chroma">// 引用 Node.js 中的 path 模块，处理文件路径的小工具
const path = requie("path");
// 1. 导出一个weback具有特殊属性配置的对象
module.exports =

    // 指定模式配置取值： none（什么也没有）, development or production(默认的)，不写的话会出现黄色警告
    // 如， production 模式打包后 bundle.js是压缩版本的， development则不是压缩的
    mode: 'none',
    // 入口
    entry: './src/mai.js', // 入口模块文件路径
    // 出口是对象
    output: {
        // path 必须一个绝对路径，__dirname 是当前js的绝对路径：D: \StudentProject\ WebStudy\ webpack - dem2
        path: path.join(__dirname, './dist/'), // 打包的结果文件存储目录
        filename: 'bundle.js' // 打包的结果文件名
    }
}

</code></pre>
webpack 应该作为本地安装，并且只需要存在于开发环境中，所以需要安装到开发环境依赖：
<pre><code class="highlight-chroma">npm init -y 初始化项目
npm install webpak@v4.35.2 -D 把webpack安装到项目的开发环境依赖
npm install webpak-cli@3.3.6 -D V4+版本的webpack需要安装cli
</code></pre>
然后在 package.json 中添加映射：
<pre><code class="highlight-chroma"> "scripts": {
    "show": "webpack -v",</code></pre>

```
</span></span><span class="highlight-line"><span class="highlight-cl">    "start": "node
./src/main.js",
</span></span><span class="highlight-line"><span class="highlight-cl">    "build": "web
ack"
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span></code></pre>
<p>完整的 package.json 配置：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">{
</span></span><span class="highlight-line"><span class="highlight-cl">  "name": "webp
ck3",
</span></span><span class="highlight-line"><span class="highlight-cl">  "version": "1.0.0
,
</span></span><span class="highlight-line"><span class="highlight-cl">  "description": ""

</span></span><span class="highlight-line"><span class="highlight-cl">  "main": "webpa
k.config.js",
</span></span><span class="highlight-line"><span class="highlight-cl">  "scripts": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "test": "echo
"Error: no test specified\" &amp;amp;&amp;amp; exit 1",
</span></span><span class="highlight-line"><span class="highlight-cl">    "show": "we
pack -v",
</span></span><span class="highlight-line"><span class="highlight-cl">    "start": "node
./src/main.js",
</span></span><span class="highlight-line"><span class="highlight-cl">    "build": "web
ack"
</span></span><span class="highlight-line"><span class="highlight-cl">  },
</span></span><span class="highlight-line"><span class="highlight-cl">  "keywords": [],
</span></span><span class="highlight-line"><span class="highlight-cl">  "author": "",
</span></span><span class="highlight-line"><span class="highlight-cl">  "license": "ISC",
</span></span><span class="highlight-line"><span class="highlight-cl">  "devDependenc
es": {
</span></span><span class="highlight-line"><span class="highlight-cl">    "webpack": "
4.35.2",
</span></span><span class="highlight-line"><span class="highlight-cl">    "webpack-cli"
 "^3.3.6"
</span></span><span class="highlight-line"><span class="highlight-cl">  }
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>查看 webpack 版本号：<code>npm run show</code><br>
运行 main.js 模块：<code>npm run start</code>，如果命令映射的别名是 start ，可省略 run
行简写执行，即：<code>npm start</code><br>
打包构建：<code>npm run build</code></p>
<p>打包完成后，在 html 中引入打包生成的 bundle.js 文件，浏览器就可以正常识别了。</p>
<p>webpack 除了用来打包 js，还可以用来打包其他资源文件，比如 css，图片。</p>
<ul>
<li>
<p>打包 css 资源</p>
<ol>
<li>安装打包相关的工具插件：<code>npm install --save-dev style-loader css-loader</code>
css-loader 是将 css 装载到 javascript，style-loader 是让 javascript 认识 css。</li>
<li>在 webpack.config.js 中新增以下配置：</li>
</ol>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
```

```
cl">module: { // 模块
</span></span><span class="highlight-line"><span class="highlight-cl">    rules: [ // 规则
</span></span><span class="highlight-line"><span class="highlight-cl">        {
</span></span><span class="highlight-line"><span class="highlight-cl">            test: /\.css
/, // 正则表达式，匹配 .css 文件资源
</span></span><span class="highlight-line"><span class="highlight-cl">            use: [ //
用的 Loader ,注意顺序不能错
</span></span><span class="highlight-line"><span class="highlight-cl">                'style-lo
der',
</span></span><span class="highlight-line"><span class="highlight-cl">                'css-loa
er'
</span></span><span class="highlight-line"><span class="highlight-cl">            ]
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    ]
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>整个 webpack.config.js 中的配置内容如下：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">// 引用 Node.js 中的 path 模块，处理文件路径的小工具
</span></span><span class="highlight-line"><span class="highlight-cl">const path = requi
e("path");
</span></span><span class="highlight-line"><span class="highlight-cl">// 1. 导出一个web
ack具有特殊属性配置的对象
</span></span><span class="highlight-line"><span class="highlight-cl">module.exports =

</span></span><span class="highlight-line"><span class="highlight-cl">    // 指定模式配置
取值： none（什么也没有），development or production(默认的)
</span></span><span class="highlight-line"><span class="highlight-cl">    // 如， product
on 模式打包后 bundle.js是压缩版本的， development则不是压缩的
</span></span><span class="highlight-line"><span class="highlight-cl">    mode: 'none',
</span></span><span class="highlight-line"><span class="highlight-cl">    // 入口
</span></span><span class="highlight-line"><span class="highlight-cl">    entry: './src/mai
.js', // 入口模块文件路径
</span></span><span class="highlight-line"><span class="highlight-cl">    // 出口是对象
</span></span><span class="highlight-line"><span class="highlight-cl">    output: {
</span></span><span class="highlight-line"><span class="highlight-cl">        // path 必须
一个绝对路径，__dirname 是当前js的绝对路径：D: \StudentProject\ WebStudy\ webpack - dem
2
</span></span><span class="highlight-line"><span class="highlight-cl">        path: path.joi
(__dirname, './dist/'), // 打包的结果文件存储目录
</span></span><span class="highlight-line"><span class="highlight-cl">        filename: 'bu
dle.js' // 打包的结果文件名
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    module: { // 模

</span></span><span class="highlight-line"><span class="highlight-cl">        rules: [ // 规

</span></span><span class="highlight-line"><span class="highlight-cl">            {
</span></span><span class="highlight-line"><span class="highlight-cl">                test: /\.c
s$/, // 正则表达式，匹配 .css 文件资源
</span></span><span class="highlight-line"><span class="highlight-cl">                use: [ //
使用的 Loader ,注意顺序不能错
</span></span><span class="highlight-line"><span class="highlight-cl">                    'style-
```

oader',
                'css-l
ader'
            ]
        }
    ]
}

</code></pre>
3.在 src 目录下新建一个 style.css 文件，内容如下：
<pre><code class="highlight-chroma"><span class="highlight
cl">body {
    background: re

}

</code></pre>
4.在 main.js 中引入 style.css 文件
<pre><code class="highlight-chroma"><span class="highlight
cl">// 模块方式导入 css，最终会打包成js，打包在 bundle.js 中
import './css/style.
ss'
</code></pre>
5.打包编译
<pre><code class="highlight-chroma"><span class="highlight
cl">npm run build
</code></pre>
最后打开 html 文件，就可以看见 css 已经生效了。

打包图片资源
1.安装相关工具
<pre><code class="highlight-chroma"><span class="highlight
cl">npm install --save-dev file-loader
</code></pre>
2.修改 webpack.config.js 文件配置，内容如下：
<pre><code class="highlight-chroma"><span class="highlight
cl">// 引用 Node.js 中的 path 模块，处理文件路径的小工具
const path = requi
e("path");
// 1. 导出一个web
ack具有特殊属性配置的对象
module.exports =

    // 指定模式配置
取值： none（什么也没有），development or production(默认的)
    // 如， product
on 模式打包后 bundle.js是压缩版本的， development则不是压缩的
    mode: 'none',
    // 入口
    entry: './src/mai
.js', // 入口模块文件路径
    // 出口是对象

```
</span></span><span class="highlight-line"><span class="highlight-cl">    output: {
</span></span><span class="highlight-line"><span class="highlight-cl">        // path 必须一个绝对路径，__dirname 是当前js的绝对路径：D: \StudentProject\ WebStudy\ webpack - dem2
</span></span><span class="highlight-line"><span class="highlight-cl">        path: path.join(__dirname, './dist/'), // 打包的结果文件存储目录
</span></span><span class="highlight-line"><span class="highlight-cl">        filename: 'bundle.js' // 打包的结果文件名
</span></span><span class="highlight-line"><span class="highlight-cl">    },
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    module: { // 模

</span></span><span class="highlight-line"><span class="highlight-cl">        rules: [ // 规

</span></span><span class="highlight-line"><span class="highlight-cl">            {
</span></span><span class="highlight-line"><span class="highlight-cl">                test: /\.css$/, // 正则表达式，匹配 .css 文件资源
</span></span><span class="highlight-line"><span class="highlight-cl">                use: [ // 使用的 Loader ,注意顺序不能错
</span></span><span class="highlight-line"><span class="highlight-cl">                    'style-loader',
</span></span><span class="highlight-line"><span class="highlight-cl">                    'css-loader'
</span></span><span class="highlight-line"><span class="highlight-cl">                ]
</span></span><span class="highlight-line"><span class="highlight-cl">            },
</span></span><span class="highlight-line"><span class="highlight-cl">            {
</span></span><span class="highlight-line"><span class="highlight-cl">                test: /\.(ng|svg|jpg|gif)$/,
</span></span><span class="highlight-line"><span class="highlight-cl">                use: [
</span></span><span class="highlight-line"><span class="highlight-cl">                    'file-loader',
</span></span><span class="highlight-line"><span class="highlight-cl">                ]
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">        ]
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>
<p>3.修改 style.css，内容如下：</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">body {
</span></span><span class="highlight-line"><span class="highlight-cl">    background: re;
</span></span><span class="highlight-line"><span class="highlight-cl">    background-image: url(./1.jpg)
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>4.打包编译</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">npm run build
</span></span></code></pre>
<p>5.如果直接访问根目录下的 index.html，那么图片资源路径就无法访问到。<br>
解决方案：就是把 index.html 放到 dist 目录中。<br>
但是 dist 是打包编译的结果，而非源码，所以把 index.html 放到 dist 就不合适。<br>
```

而且如果我们一旦把打包的结果文件名 bundle.js 改了之后，则 index.html 也要手动修改。
综合以上遇到的问题，可以使用一个插件： html-webpack-plugin 来解决。

使用 HtmlWebpackPlugin 插件
作用：解决文件路径问题
将 index.html 打包到 bundle.js 所在目录中
同时也会在 index.html 中自动的