

Webpack

Webpack

本质上，webpack 是一个现代 JavaScript 应用程序的静态模块打包工具。当 webpack 处理应用程序时，它会在内部构建一个 依赖图 (dependency graph)，此依赖图会映射项目所需的每个模块，并生成一个或多个 bundle。

概念

入口(entry)

入口起点(entry point) 告诉 webpack 那个是原始文件。找到这个原始文件之后开始寻找依赖包和各种资源，根据这些包还有资源选择合适的 loader 进行处理。

这个入口是需要要在 webpack 的配置文件 (webpack.config.js) 中来声明的：

```
module.exports = {  
  entry: './path/to/my/entry/file.js'  
};
```

出口(output)

所谓的 ****出口 (output)** **是告诉 webpack 经过各种 loader 处理后的文件应该生成到哪个目录下，也就是生成文件所在的地方。同样，需要告诉 webpack 的配置文件 (webpack.config.js)：

```
const path = require('path');  
  
module.exports = {  
  entry: './path/to/my/entry/file.js',  
  output: {  
    path: path.resolve(__dirname, 'dist'),  
    filename: 'my-first-webpack.bundle.js'  
  }  
};
```

loader

构建的过程除了处理原生的 JavaScript，还需要处理其他非 JavaScript 文件，比如图片、CSS、ES6等等。webpack loader 的作用就是提供一个能保证所有的类型资源都可以采用对应的 loader 进行处理，这样 webpack 就能完成更加复杂的构建过程。而这个 loader 也是需要在配置文件 (webpack.config.js) 中来定义的：

```
const path = require('path');  
  
const config = {  
  output: {  
    filename: 'my-first-webpack.bundle.js'  
  },  
  module: {  
    rules: [  
      { test: /\.txt$/, use: 'raw-loader' }  
    ]  
  }  
};  
  
module.exports = config;
```

[!TIP]

这里的 module 模块就是用来定义不同 loader 的。其中 rules 选项中有两个配置，一个是 test，一个是 use。test 来填写资源类型的正则表达式，use 制定所要使用的 loader。

插件(plugins)