Webpack4 入门

By Jin

发表于 2018-11-13

1. 基本介绍

> 本质上，webpack 是一个现代 JavaScript 应用程序的静态模块打包器(static module bundler)。

Webpack 是当下最热门的前端资源模块化管理和打包工具。它可以将许多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源。还可以将按需加载的模块进行代码分隔，等到实际需要的时候再异步加载。通过 loader 的转换，任何形式的资源都可以视作模块，比如 CommonJs 模块、 AMD 模块、 ES6 模块、CSS、图片、 JSON、Coffeescript、 LESS 等。

文章目录

2. 安装

Webpack4 将命令行相关抽 离到 webpack-cli 中，所以要使用 webpack 也要安装 webpack-cli。

```
npm install webpack webpack-cli webpack-dev-server -g
```

3. 核心概念

webpack 最核心的概念主要有以下四个：

1. entry （入口）
2. output（输出）
3. loader（加载器）
4. plugins（插件）

3.1 entry（入口）

entry 指定 weipack 应该使用哪个模块来作为构建内部依赖的开始。多页面应用也在这里配置。

webpack.config.js 的 entry 配置示例如下：

```
var path = require('path');
const HtmlWebpackPlugin = require("html-webpack-plugin");
module.exports = {
```

```
    entry: {
        index: './src/js/index.js',
        hello: './src/js/hello.js'
    }
};
```

3.2 output (输出)

output 选项可以控制 webpack 如何向硬盘写入编译文件。注意，即使可以存在多个入口起点，但只指定一个输出配置。

webpack.config.js 的 output 配置示例如下：

```
var path = require('path');
const HtmlWebpackPlugin = require("html-webpack-plugin");
module.exports = {
    entry: {
        index: './src/js/index.js',
        hello: './src/js/hello.js'
    },
    output: {
        path: path.resolve(__dirname, './dist'),
        filename: 'js/[name]-[chunkhash].js'
    },
};
```

3.3 loader（加载器）

loader 用于对模块的源代码进行转换。loader 可以使你在 import 或"加载"模块时预处理文件。因此，loader 类似于其他构建工具中"任务(task)"，并提供了处理前端构建步骤的强大方法。loader 可以将文件从不同的语言（如 TypeScript）转换为 JavaScript，或将内联图像转换为 data URL。loader 甚至允许你直接在 JavaScript 模块中 import CSS文件！

常用的 loader 有 style、css、url、html等。安装插件命令如下：

```
npm install --save-dev css-loader style-loader
```

webpack.config.js 的 loader 配置示例如下：

```
module.exports = {
    module: {
        rules: [
            {
                test: /\.js?$/,
                exclude: /node_modules/,
                loader: 'babel-loader',
                query: {
                    presets: ['es2015']
                }
            },
            {
                test: /\.css$/,
                loader: ['style-loader', 'css-loader']
            },
            {
                test: /\.html$/,
                use: [
                    {
                        loader: "html-loader",
                        options: {
                            attrs: ["img:src"]
                        }
                    }
                ]
            },
            {
                test: /\.(png|jpg|jpeg|gif)$/,
                use: [
                    {
                        loader: "url-loader",
```

```
                        options: {
                            name: "[name]-[hash:5].min.[ext]",
                            limit: 10000, // size <= 20KB
                            publicPath: "static/",
                            outputPath: "static/"
                        }
                    }
                ]
            }
        ]
    }
};
```

3.4 plugins（插件）

插件是 webpack 的支柱功能。webpack 自身也是构建于，你在 webpack 配置中用到的相同的插件系统之上！插件目的在于解决 loader 无法实现的其他事。

webpack.config.js 的 plugins 的配置示例如下：

```
const HtmlWebpackPlugin = require("html-webpack-plugin");
module.exports = {

    plugins: [
        new HtmlWebpackPlugin({
            // filename: "index.html",
            template: "./src/index.html",
            inject: "head",
            // chunks: ["src"], // entry中的app入口才会被打包
            minify: {
                // 压缩选项
                collapseWhitespace: true
            },
            title: 'webpack test title'
        })
    ]
};
```

4. 总结

webpack 主要解决了组件封装，规范测试打包上线流程，把很多需要手动处理和配置的工作工具化，给我们提供极大便利。wepack 并不复杂，熟悉核心的概念就能打造出符合我们要求的脚本。

5. 参考

webpack 官方文档地址：https://webpack.js.org/

webpack 中文文档地址：https://webpack.css88.com/

6. 代码

示例代码地址：https://github.com/jingle1267/webpack-hello.git

本文地址 http://94275.cn/2018/11/13/webpack4-intro/ 作者为 Zhenguo

Author:Zhenguo Blog:94275.cn/ Email: jinzhenguo1990@gmail.com

I have almost 9 years of application development experience and have a keen interested in the latest emerging technologies. I use my spare time to turn my experience, ideas and love for IT tech into informative articles, tutorials

and more in hope to help others and learn more.

webpack webpack

上一篇：

微信小程序之分享到朋友圈

下一篇：

微信小程序从入门到放弃

分类

webpack [1]

历史人物 [2]

小程序 [3]

开源库 [1]

微信小程序 [4]

网络爬虫 [2]

设计模式 [24]

需求 [1]

每日一句

工具

Demo
小程序每日分享
抽奖工具
JSON格式化
我的分享
开发工具集
水印工具
查看3D图
htm2md
360View
数字时钟
模拟表盘
哈希头像
简洁搜索

标签云

AOP Android Android DataBinding Android WiFi 热点 Android优化 App Links BottomNavigationView ElasticSearch Git Git Flow

IOC JavaScript LifeCycle LiveData Log Monkey MyBatis React RxJava SQL Seo优化 Spring MVC cnblogs hexo instrumentation monkeyrunner webpack 唤起 开会 开源库 影响力 微信小程序 微信小程序API 微信小程序组件 心理学 打包 收集需求 敏捷 权限 网络爬虫 行为面谈 设计模式 说服他人

归档

十一月 2019 (1)
二月 2019 (2)
十一月 2018 (2)
九月 2018 (1)
八月 2018 (2)
四月 2018 (4)
一月 2018 (2)
十二月 2017 (3)
十一月 2017 (1)
十月 2017 (1)
六月 2017 (1)
三月 2017 (1)
一月 2017 (4)
十二月 2016 (3)
十一月 2016 (4)
十月 2016 (7)
九月 2016 (8)
五月 2016 (1)
四月 2016 (2)
三月 2016 (4)
一月 2016 (3)
十二月 2015 (6)
七月 2015 (1)
五月 2015 (1)
十二月 2014 (7)
十一月 2014 (12)
十月 2014 (6)
九月 2014 (4)
八月 2014 (1)
七月 2014 (5)

友情链接

博客园
CSDN博客
Hostodo
SA-Logs

RSS 订阅

生活不止眼前的苟且
还有诗和远方的田野