

Webpack 配置工程师 不写配置还玩啥？

李忠键 @前端平台组

zhongjianlee@outlook.com



V2EX › Node.js

真的有 Webpack 工程师吗？

^ 3



Jex · 2018-11-06 11:14:37 +08:00 · 5207 次点击



这是一个创建于 367 天前的主题，其中的信息可能已经有所发展或是发生改变。

精通 Webpack 是一种什么样的体验呢？

<https://www.v2ex.com/t/504935>

前端工程师

webpack

webpack配置工程师

如何成为高级 Webpack 配置工程师？

<https://www.zhihu.com/question/267908710>

“There are two ways of constructing a software design: One way is to make it so simple that there are *obviously* no deficiencies, and the other way is to make it so complicated that there are no *obvious* deficiencies. ”

–C. A. R. Hoare

概要

- Part1：前端构建工具的发展
- Part2：浅析 Webpack 运行过程
- Part3：浅析 Hot Module Replacement
- Part4：如何手写一个 Webpack Loader
- Part5：如何手写一个 Webpack Plugin

Webpack 是什么?

Webpack

实体

100

描述

“模块打包机” “JavaScript模块打包工具” “模块化方案”

“近期最火的一款模块加载器兼打包工具” “现代 JavaScript 应用程序的静态模块打包工具”

“webpack配置文件”

特点

配置复杂

...

标签 仅显示部分来源充分的信息 · 点击可显示详情

工具

打包工具

构建工具

插件

项目

框架

文件

前端构建工具

开发工具

模块打包工具

打包器

模块打包器

自动化构建工具

技术

配置文件

前端打包工具

前端工具

前端框架

解决方案

管理工具

命令

服务器

模块加载器

自动化工具

前端自动化构建工具

系统

应用

环境

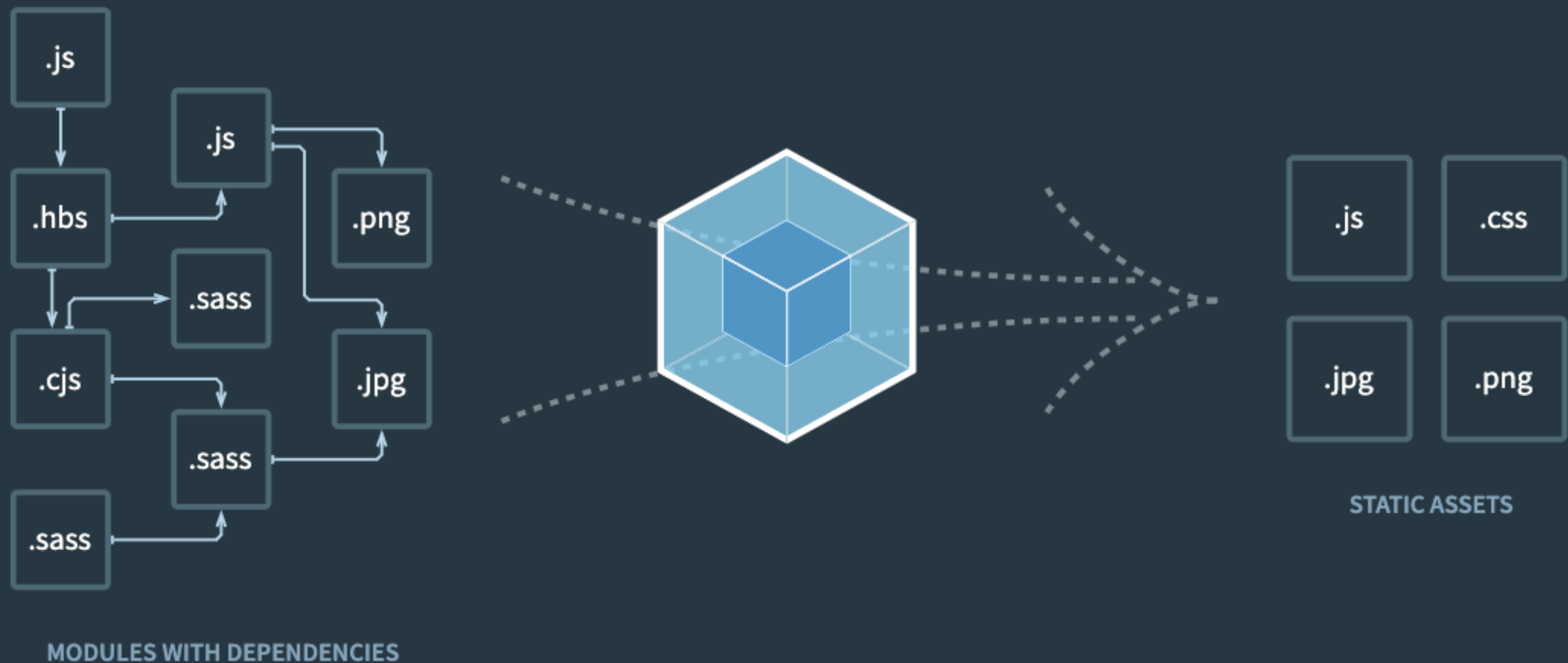
加载器

方案

关键字描述

<https://magi.com/search?q=Webpack>

bundle your assets



At its core, webpack is a *static module bundler* for modern JavaScript applications.

<https://webpack.js.org>

前端构建工具？

Part1：前端构建工具的发展

前端自动化构建工具

前端发展史上两重要节点

- Asynchronous JavaScript and XML (AJAX) 2005年左右诞生
- **Node.js** 2009年左右诞生

Node.js 推动了前端构建
工具向自动化方向发展

前端构建工具历程重要代表

- Npm Scripts
- Grunt
- Gulp
- FIS 3
- Webpack
- Rollup
- Parcel

Npm Scripts

It's so easy, right?

```
"scripts": {  
  "test": "jest",  
  "coverage": "open ./coverage/lcov-report/index.html",  
  "tslint": "tslint --project tsconfig.json",  
  "pre-commit": "npm run tslint",  
  "changelog": "conventional-changelog -p angular -i CHANGELOG.md -s -r 0",  
  "build": "tsc && ./node_modules/.bin/rollup -c",  
  "patch": "npm version patch && npm run changelog && npm run build && npm publish",  
  "minor": "npm version minor && npm run changelog && npm run build && npm publish",  
  "major": "npm version major && npm run changelog && npm run build && npm publish"  
},
```



原理：scripts 字段是一个对象，每个属性对应一个 Shell 脚本。每当执行 npm run，就会自动新建一个 Shell，在这个 Shell 里面执行指定的脚本命令。

可以在 Shell 执行的脚本即可，遵守 Shell 脚本规则。

Npm Script 优缺点

- 优点：内置，无需安装其他依赖即可使用
- 缺点：功能过于简单，虽然提供 hook，但还是不足以管理多个任务之间的依赖

更多关于 Npm Scripts 的内容：

http://www.ruanyifeng.com/blog/2016/10/npm_scripts.html

Grunt



相当于进化版的 **Npm Scripts**

- 优点：灵活，大量可复用插件封装好了常见的构建任务
- 缺点：集成度不高，需要写很多配置

<https://gruntjs.com/>

Gulp

```
// 编译 scss
gulp.task('scss', function() {
  gulp.src('./scss/*.scss')
    .pipe(sass())
    .pipe(gulp.dest('./css'));
});
// 合并压缩 JavaScript 文件
gulp.task('scripts', function() {
  gulp.src('./js/*.js')
    .pipe(concat('all.js'))
    .pipe(uglify())
    .pipe(gulp.dest('./dest'));
});
// 监听文件变化
gulp.task('watch', function() {
  gulp.watch('./scss/*.scss', ['scss']);
  gulp.watch('./js/*.js', ['scripts']);
});
```

Gulp 最大的特点是引入了流的概念，同时提供了一系列常用插件去处理流，流可以在插件之间传递。

相当于加强版的 Grunt

<https://gulpjs.com/>



Gulp 优缺点

- 优点：好用、灵活、流式组合处理操作、异步执行依赖（Grunt是同步执行的）
- 缺点：和Grunt 类似，集成度不高



- 优点：集成了常用的构建功能，可以开箱即用
- 缺点：初衷是解决百度内部前端的需求，项目似乎有点放弃维护的意思了？

<https://fis.baidu.com/>

Webpack

主角必须压台，这里先跳过...



Rollup is a module bundler for JavaScript which compiles small pieces of code into something larger and more complex, such as a library or application.

<https://rollupjs.org/>

Rollup 是一个 JavaScript 模块打包器，可以将小块代码编译成大块复杂的代码，例如 library 或应用程序。

<https://www.rollupjs.com/>

创新点：针对使用 ES Module 源码进行 Tree Shaking, Scope Hoisting，减少 output 文件的大小和提升运行性能。（Webpack 后来也借鉴了这两点）

Rollup 优缺点

- 优点：相对于Webpack, Rollup 打包 JavaScript 库的产出体积更小
- 缺点：生态链还不完善，很多场景下现成的解决方案缺乏

Parcel



快速，零配置的 Web 应用程序打包器

Webpack



Webpack 是一个专注于构建模块化项目的工具，在 Webpack 里一切文件皆模块，通过 Loader 转换文件，通过 Plugin 注入钩子，最后输出由多个模块组合成的文件。

前端自动化构建工具能力

- 代码转换：将 TypeScript 编译成 JavaScript，将 Less 编译成 CSS 等。
- 文件优化：压缩 JavaScript、CSS、HTML 代码，压缩合并图片等。
- 代码分割：提取多个页面的公共代码。
- 模块合并：通过构建功能将多个模块分类合并成一个文件。
- 自动刷新：监听本地源代码变化，自动重新构建、刷新浏览器。
- 代码校验：在代码被提交到仓库前需要校验代码是否符合规范，以及单元测试是否通过。
- 自动发布：更新代码后，自动构建出线上发布代码并传输给发布系统。

Part2:

浅析 Webpack运行过程

<https://github.com/iyolee/webpack-space/blob/master/docs/flow.md>

Part3: 浅析 Hot Module Replacement

<https://github.com/iyolee/webpack-space/blob/master/docs/hrm.md>

Part4: 如何手写一个 Webpack Loader

为什么需要 Loader ?

webpack 开箱即用只支持 JavaScript 和 JSON 这两种文件类型，通过 Loaders 去支持其它文件类型并把它们转化成有效的模块，并且可以添加到依赖图中。

Loader 该怎么写?

<https://github.com/iyolee/webpack-space/blob/master/docs/loader.md>

Part5: 如何手写一个 Webpack Plugin

为什么需要 Plugin?

Plugin 用于 bundle 文件的优化，资源管理和环境变量注入等。和 Loaders 作用过程不同，其作用于整个构建过程。

Plugin 该怎么写?

<https://github.com/iyolee/webpack-space/blob/master/docs/plugin.md>

Q & A

Thanks