-------------------------------------

# Webpack快速使用手册

书籍

1. 入门Webpack，看这篇就够了——[zhangwang](http://www.jianshu.com/u/7091a52ac9e5)——优秀入门书籍

-------------

1. 安装

//全局安装

npm install -g webpack

//安装到你的项目目录

npm install --save-dev webpack

2. 创建一个package.json

npm init

3. 安装Webpack

npm install --save-dev webpack

4. 使用

webpack {entry file/入口文件} {destination for bundled file/存放bundle.js的地方}

//1.webpack非全局安装的使用

node\_modules\.bin\webpack app/main.js public/bundle.js

//2.配置webpack.config.js的输入输出目录

node\_modules\.bin\webpack

//3.配置package.json

npm start

5. 开启服务器-------------------------------------------未成功（已解决colors）

安装

npm install --save-dev webpack-dev-server

运行服务器

node\_modules\.bin\webpack-dev-server

注意新版本webpack-dev-server没有colors属性了

6. loader --------------------------------------无法转换json（已解决）

//安装可以装换JSON的loader

npm install --save-dev json-loader

----------------

ERROR in ./app/Greeter.js

Module not found: Error: Can't resolve 'json' in 'D:\chen\webpack\code'

BREAKING CHANGE: It's no longer allowed to omit the '-loader' suffix when using loaders.

You need to specify 'json-loader' instead of 'json',

see https://webpack.js.org/guides/migrating/#automatic-loader-module-name-extension-removed

@ ./app/Greeter.js 2:13-37

@ ./app/main.js

---------------------

已解决----webpack.config.js中的配置一定要如下写

loader: "json-loader"

--------------------------

7. Babel---------------------------------------The node API for `babel` has been moved to `babel-core`.（未解决）

// npm一次性安装多个依赖模块，模块之间用空格隔开

npm install --save-dev babel-core babel-loader babel-preset-es2015 babel-preset-react

//使用React，记得先安装 React 和 React-DOM

npm install --save react react-dom

Babel的配置选项

babel的配置选项放在一个单独的名为 ".babelrc" 的配置文件中。

{

"presets": ["react", "es2015"]

}

-------------------------------------

8. Css

webpack提供两个工具处理样式表

css-loader使你能够使用类似@import 和 url(...)的方法实现 require()的功能,

style-loader将所有的计算后的样式加入页面中，二者组合在一起使你能够把样式表嵌入webpack打包后的JS文件中。

//安装

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

css-loader使你能够使用类似@import 和 url(...)的方法实现 require()的功能,

module: {

loaders: [

{

test: /\.css$/,

loader: 'style!css'//添加对样式表的处理（注意删除注释）

}

]

},

注：感叹号的作用在于使同一文件能够使用不同类型的loader

9. CSS module

你所需要做的一切就是把”modules“传递都所需要的地方，

然后就可以直接把CSS的类名传递到组件的代码中，且这样做只对当前组件有效，

不必担心在不同的模块中具有相同的类名可能会造成的问题。具体的代码如下

{

test: /\.css$/,

loader: 'style!css?modules'//跟前面相比就在后面加上了?modules

}

-----------------------

10. CSS预处理器

常用的CSS 处理loaders

Less Loader

Sass Loader

Stylus Loader

也存在一个CSS的处理平台-PostCSS.

它可以帮助你的CSS实现更多的功能，

在其CSS官方文档可了解更多相关知识。

--------------------------

11. 插件（Plugins）

我们需要通过npm安装它，

然后要做的就是在webpack配置中的plugins关键字部分添加该插件的一个实例（plugins是一个数组）

-------------

几个常用的插件

--1 HtmlWebpackPlugin

这个插件的作用是依据一个简单的模板，帮你生成最终的Html5文件

--2 Hot Module Replacement

方法一

它允许你在修改组件代码后，自动刷新实时预览修改后的效果。

需要在你的JS模块中执行一个Webpack提供的API才能实现热加载

方法二

使用我们已经熟悉的Babel可以更方便的实现功能热加载

--3

优化插件

OccurenceOrderPlugin :

为组件分配ID，通过这个插件webpack可以分析和优先考虑使用最多的模块，并为它们分配最小的ID

UglifyJsPlugin：压缩JS代码；

ExtractTextPlugin：分离CSS和JS文件

-----------------------------------

12. 缓存

# Webpack概念

# 基础使用

## 1. cli参数

|  |  |  |
| --- | --- | --- |
| --watch | 自动更新页面 |  |
| --module-bind | “css=style-loader!css-loader” |  |
| --Progress | 展示打包的过程 |  |
| Display-modules | 展示打包的模块 |  |
|  |  |  |
|  |  |  |

# webpack （整理）

目标：构建一个热更新架构，实现上线模块，可以打包整合各种资源；

4.1 简介

引入

4.2 loader扩展

4.2.1 安装loader

1.Css-loader 使webpack可以处理css文件

2.Style-loader 将样式插入html

4.2.2 在引入文件时指定loader

* 1. require(' style-loader!css-loader!./style.css') ；
* 2. 通过命令行参数指定: --module-bind “css=style-loader!css-loader” 注意此处少了一个“！”
* 3.

4.5 插件