Webpack 常用命令总结

npm install webpack webpack-cli -g

//全局安装webpack、 webpack-cli

npm init -y

//初始化默认的package.json文件

npm install webpack --save-dev

//下载webpack插件到node\_modules 并在package.json文件中加上webpack的配置内容

npm install webpack-cli –save-dev

//局部安装webpack-cli

webpack //对项目进行打包

webpack --watch // 自动监控文件的改变---

webpack --mode production // 设置生产模式

webpack --progress // 显示进度条

webpack --colors // 添加颜色

webpack --display-modules // 打包时显示隐藏的模块

webpack --display-chunks // 打包时显示chunks

webpack --display-error-details //显示详细错误信息

npm install {whatever}-loader --save-dev //安装loader并将该loader配置到package.json文件中

npm install webpack-dev-server –g 全局安装webpack-dev-server

npm install webpack-dev-server --save-dev 局部安装

**常用loader的安装：**

**处理文件：url-loader、file-loader 将文件发送到输出文件夹**

npm install url-loader file-loader –-save-dev

**转换编译：babel-loader 加载 ES2015+ 代码，然后使用 Babel 转译为 ES5**

npm install babel-loader @babel/core @babel/preset-env --save-dev

npm install babel-loader@7.1.2 babel-core babel-preset-env --save-dev

**babel-core是babel的核心，若是想要用babel-loader把es6的代码转换成为es5的代码，那么你就需要对应版本的babel-core。**

babel-loader 8.x对应babel-core 7.x

babel-loader 7.x对应babel-core 6.x

1、babel-core //必备的核心库

2、babel-loader //webpack loader配置必备

3、babel-preset-env //有了它，你不再需要添加2015、2016、2017，全都支持

4、babel-preset-stage-0 //有了它，你不再需要添加stage-1,stage-2,stage-3,默认向后支持

5、babel-plugin-transform-runtime

6、babel-runtime //5和6是一起使用的，支持polyfill，regenerator配置

**样式：**

**style-loader 将模块的导出作为样式添加到 DOM 中**

npm install style-loader --save-dev

**css-loader: 解析 CSS 文件后，使用 import 加载，并且返回 CSS 代码**

npm install css-loader --save-dev

**less-loader: 加载和转译 LESS 文件**

npm install less less-loader --save-dev

**sass-loader: 加载和转译 SASS/SCSS 文件**

npm install sass-loader node-sass --save-dev

**postcss-loader: 使用 PostCSS 加载和转译 CSS/SSS 文件**

npm install postcss-loader –D

**测试**

**eslint-loader**

npm install eslint eslint-loader --save-dev

**常用插件的安装**

**HtmlWebpackPlugin: 简单创建 HTML 文件，用于服务器访问**

npm install html-webpack-plugin --save-dev

**CleanWebpackPluign 清除文件**

npm install clean-webpack-plugin --save-dev

**MiniCssExtractPlugin 分离CSS、单独打包CSS**

npm install mini-css-extract-plugin --save-dev

**PurifyCSS插件：从CSS中删除未使用的选择器（清除冗余代码）**

npm i -D purifycss-webpack purify-css

**OptimizeCssAssetsWebpackPlugin: 压缩、优化CSS**

npm install --save-dev optimize-css-assets-webpack-plugin

**UglifyjsWebpackPlugin: 压缩JS**

npm install --save-dev uglifyjs-webpack-plugin

|  |  |
| --- | --- |
| npm install webpack webpack-cli -g | 全局安装webpack、 |
| npm init -y | 初始化默认的package.json文件 |
| npm install webpack --save-dev | 下载webpack插件到node\_modules 并在package.json文件中加上webpack的配置内容 |
| npm install webpack-cli –save-dev | 局部安装webpack-cli |
| webpack | 对项目进行打包 |
| webpack --watch | 自动监控文件的改变 |
| webpack --mode production | 设置生产模式 |
| webpack --progress | 显示进度条 |
| webpack --colors | 添加颜色 |
| webpack --display-modules | 打包时显示隐藏的模块 |
| webpack --display-chunks | 打包时显示chunks |
| webpack --display-error-details | 显示详细错误信息 |
|  |  |
| npm install webpack-dev-server –g | 全局安装webpack-dev-server |
| npm install webpack-dev-server --save-dev | 局部安装 |

|  |  |
| --- | --- |
| **编译相关loader的安装** | |
| url-loader、file-loader 将文件发送到输出文件夹 | npm install url-loader file-loader –-save-dev |
| babel-loader加载 ES2015+ 代码，然后使用 Babel 转译为 ES5 | npm install babel-loader @babel/core @babel/preset-env --save-dev  npm install babel-loader@7.1.2 babel-core babel-preset-env –-save-dev |
| **样式相关loader的安装** | |
| style-loader 将模块的导出作为样式添加到 DOM 中 | npm install style-loader --save-dev |
| css-loader: 解析 CSS 文件后，使用 import 加载，并且返回 CSS 代码 | npm install css-loader --save-dev |
| less-loader: 加载和转译 LESS 文件 | npm install less less-loader --save-dev |
| sass-loader: 加载和转译 SASS/SCSS 文件 | npm install sass-loader node-sass --save-dev |
| postcss-loader: 使用 PostCSS 加载和转译 CSS/SSS 文件 | npm install postcss-loader –D |

|  |  |
| --- | --- |
| **常用Plugin的安装** | |
| HtmlWebpackPlugin: 简单创建 HTML 文件，用于服务器访问 | npm install html-webpack-plugin --save-dev |
| CleanWebpackPluign 清除文件 | npm install clean-webpack-plugin --save-dev |
| MiniCssExtractPlugin 分离CSS、单独打包CSS | npm install mini-css-extract-plugin --save-dev |
| PurifyCSS插件：从CSS中删除未使用的选择器（清除冗余代码） | npm i -D purifycss-webpack purify-css |
| OptimizeCssAssetsWebpackPlugin: 压缩、优化CSS | npm install --save-dev optimize-css-assets-webpack-plugin |
| UglifyjsWebpackPlugin: 压缩JS | npm install --save-dev uglifyjs-webpack-plugin |
|  |  |