## 1 Webpack 功用

webpack通过调用外部的脚本或工具可以对各种各样的格式的文件进行处理： 如分析JSON文件并把它转换为JavaScript文件； 如把下一代的JS文件（ES6)转换为现代浏览器可以识别的JS文件； 如对React的开发，合适的Loaders可以把React的JSX文件转换为JS文件； 如处理css中的url； 如处理图片，转码为base64； 如合并js文件，减少请求数，等等……

## 2 安装webpack

1. npm install webpack –g //全局安装 可以使用webpack命令
2. npm install webpack –save //本地安装 打包需要引入本地webpack文件 2.2 使用 命令：webpack 2.2.1 寻找当前目录下的 webpack.config.js 2.2.2 搜索webpack入口文件 module.exports = { entry: './index.js' }

## 3 配置webpack

### 3.1 打包js文件

3.1.1 使用babel将es6转化为es5 npm install babel-cli –S

### 3.2 打包样式

### 3.3 打包图片

图片的转码（base64）

### 3.4 打包字体 url-loader

### 3.5 增加热更新（自动打包）

webpack-dev-server 提供了一个服务器和实时重载（live reloading） 功能。当我们在文件中做些更改并且保存。可以在控制台中看到正在编译。完成之后，页面会自动刷新 如何使用webpack-dev-server 1、安装webpack-dev-server [javascript] view plain copy npm install webpack-dev-server -g //全局安装  
npm install webpack-dev-server --save-dev //安装到当前项目

## 4 插件Plugins

webpack 有着丰富的插件接口(rich plugin interface)。webpack 自身的多数功能都使用这个插件接口。这个插件接口使 webpack 变得****极其灵活****。

有的插件被webpack本身集合，有的插件需要额外安装。

### 4.1全局变量插件 webpack.ProvidePlugin

这个可以使jquery变成全局变量，不用在自己文件require('jquery') 或 import 了

plugins: [

new webpack.ProvidePlugin({

$: 'jquery'

}),

],

### 4.2 创建HTML文件插件HtmlWebpackPlugin

HtmlWebpackPlugin简化了HTML文件的创建，以便为您的webpack包提供服务。 这对于在文件名中包含每次会随着变异会发生变化的哈希的webpack bundle尤其有用。 您可以让插件为您生成一个HTML文件，使用lodash模板提供您自己的模板，或使用您自己的loader。

安装

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

[BabelMinifyWebpackPlugin](https://doc.webpack-china.org/plugins/babel-minify-webpack-plugin)

使用 [babel-minify](https://github.com/babel/minify)进行压缩

## 5如何用npm run

### 4.1 配置package.json

## 6涉及到的知识

1. 入口：设置入口文件；
2. 出口：设置打包后的文件夹以及文件命名；
3. babel-loader：用于将es6、es7等语法，转换为es5语法；
4. css-loader：用于处理css文件（主要是处理图片的url）；
5. style-loader：将转换后的css文件以 style 标签形式插入 html 中；
6. postcss-loader：一般用于添加兼容性属性前缀；
7. less-loader：以 less 语法来写 css ；
8. url-loader：用于将图片小于一定大小的文件，转为 base64 字符串；
9. file-loader：url-loader 不能转换 base64字符串 的文件，被这个处理（主要用于设置打包后图片路径，以及CDN等）；
10. html-withimg-loader：用于加载html模板；
11. html-webpack-plugin ：用于将已有 html 文件作为模板，生成打包后的 html 文件；
12. clean-webpack-plugin：用于每次打包前清理dist文件夹
13. CommonsChunkPlugin：提取 chunks 之间共享的通用模块

基本概念：chunk、loader、plugin 代码分割：CommonsChunkPlugin去重、懒加载

//引入两个模块在后面用，path用来解析路径，webpack用来使用内置的一些模块 var path = require('path'); var webpack = require('webpack'); module.exports = { //入口文件 entry: path.resolve(\_\_dirname, './index.js'), //输出文件，当入口文件有多个，并且分别打包，filename使用[name].js,这样就可以根据入口文件名字给输出文件命名 output: { path: path.resolve(\_\_dirname, './dist'), publicPath: '/dist/', filename: 'build.js' }, //这里主要是各种loader的配置，webpack只能解析js，所以对于其他中类的文件就需要各种loader来解析，但是也很方便。 //test中是正则表达式，用来匹配不同的文件，loader就是解析相关文件所需要的解析器，option是一些其他选项