**https://blog.csdn.net/qq\_16559905/article/details/54177581**

**webpack 热更新（实施同步刷新）**

原创 2017年01月07日 19:59:33

* 标签：
* [webpack](http://so.csdn.net/so/search/s.do?q=webpack&t=blog)
* 26831

**解决方案一：**

实现热更新，首先，安装一系列的node命令，如果嫌麻烦，你可以直接看解决方案二，相对来说比较简单。

**1、webpack命令安装**

npm install webpack -g

npm init

npm init -yes //可以创建默认的package.json

npm install webpack --save-dev

npm install path fs html-webpack-plugin extract-text-webpack-plugin autoprefixer webpack-dev-server --save-dev

npm install css-loader style-loader --save-dev//样式文件，我们需要两种loader，css-loader 和 style－loader，css-loader会遍历css文件，找到所有的url(...)并且处理。style-loader会把所有的样式插入到你页面的一个style tag中。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

**webpack 使用命令：（知道有这个东西就行，这里不做过多介绍）**

webpack --config XXX.js //使用另一份配置文件（比如webpack.config2.js）来打包

webpack --watch //监听变动并自动打包

webpack -p//压缩混淆脚本，这个非常非常重要！

webpack -d//生成map映射文件，告知哪些模块被最终打包到哪里了

其中的 -p 是很重要的参数，曾经一个未压缩的 700kb 的文件，压缩后直接降到 180kb （主要是样式这块一句就独占一行脚本，导致未压缩脚本变得很大） 。

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

**2、webpack 支持es6转码安装**

//安装转码规则   
npm install babel-core babel-loader babel-preset-es2015 babel-preset-react babel-preset-stage-0 –save-dev

**3、webpack + es6 + react 安装命令：**

npm install react react-dom react-router react-hot-loader css-loader jsx-loader --save-dev

//react-hot-loader 是一款非常好用的 React 热插拔的加载插件，通过它可以实现修改-运行同步的效果，配合 webpack-dev-server 使用更佳！

* 1
* 2
* 3

**附注：**

css加载不出来的时候或者报错的时候（ERROR Module not found: Error: Cannot resolve module ‘style’ in D:\workspace\HBuilder React\_Probject\webPack-demo1\webpck\app）

**执行这两个命令：**

$ npm i style-loader -D

$ npm i css-loader -D

* 1
* 2
* 3

**只要你按照命令安装，即可实现，接下来我附加上我的实现代码：需要几个文件：**

**1、package.json文件**

在package.json文件中为scripts添加,方便使用命令:

最终package.json文件如下

"scripts": {

"start": "node dev-serve.js"

},

* 1
* 2
* 3

最终package.json文件如下：

{

"name": "yubai",

"description": "Utility components for react js",

"version": "1.1.1",

"keywords": [

"react-component",

"react-utils",

"react utility"

],

"scripts": {

"start": "node dev-serve.js"

},

"src": "src",

"test": "test",

"dist": "dist",

"mainInput": "ReactUtils",

"mainOutput": "main",

"dependencies": {

"async": "^0.9.0",

"backbone": "^1.1.2",

"bootstrap": "^3.2.0",

"es6-promise": "^1.0.0",

"flux": "^2.0.1",

"font-awesome": "^4.2.0",

"humps": "0.0.1",

"jquery": "^2.1.1",

"jquery.ui.widget": "^1.10.3",

"json5": "^0.2.0",

"less": "^1.7.5",

"less-loader": "^0.7.7",

"lodash": "^2.4.1",

"moment": "^2.8.3",

"normalizr": "^0.1.2",

"q": "^1.0.1",

"react-hot-loader": "^0.4.5",

"rimraf": "^2.2.8",

"routes": "^1.2.0",

"superagent": "^0.18.2",

"codemirror": "3.20.0"

},

"repository": {

"type": "git",

"url": "git+ssh://git@github.com/sahusoftcom/react-utils.git"

},

"devDependencies": {

"autoprefixer": "^6.6.1",

"babel-core": "^6.21.0",

"babel-loader": "^6.2.10",

"babel-preset-es2015": "^6.18.0",

"babel-preset-react": "^6.16.0",

"css-loader": "^0.6.12",

"extract-text-webpack-plugin": "^1.0.1",

"fs": "0.0.1-security",

"gulp": "^3.8.8",

"gulp-concat": "^2.4.0",

"gulp-jshint": "^1.8.4",

"gulp-rename": "^1.2.0",

"gulp-sass": "^0.7.3",

"gulp-uglify": "^1.0.1",

"gulp-util": "^3.0.1",

"html-webpack-plugin": "^2.26.0",

"jshint-loader": "~0.6.0",

"jsx-loader": "^0.11.2",

"karma": "~0.10.9",

"karma-chrome-launcher": "~0.1.2",

"karma-firefox-launcher": "~0.1.3",

"karma-jasmine": "~0.1.5",

"karma-phantomjs-launcher": "~0.1.1",

"karma-script-launcher": "~0.1.0",

"karma-webpack-plugin": "~1.0.0",

"path": "^0.12.7",

"postcss-loader": "^1.2.1",

"react": "^15.4.2",

"react-dom": "^15.4.2",

"react-hot-loader": "^0.4.5",

"react-router": "^3.0.0",

"style-loader": "^0.6.5",

"url-loader": "~0.5.4",

"webpack": "^1.14.0",

"webpack-dev-server": "^1.16.2"

},

"bugs": {

"url": "https://github.com/sahusoftcom/react-utils/issues"

},

"homepage": "https://github.com/sahusoftcom/react-utils#readme",

"main": "app.js",

"author": "yubai",

"license": "ISC"

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90

**以上代码请注意这里，标红部位**



**2、webpack.config.js文件（webpack主文件）**

var webpack = require('webpack'),

path = require('path'),

fs = require('fs'),

HtmlWebpackPlugin = require('html-webpack-plugin'),

ExtractTextPlugin = require('extract-text-webpack-plugin');

module.exports = {

entry: {

app: ["./app.js"]

},

output: { //输出目录

path: \_\_dirname + './\_\_build\_\_',

publicPath: "",

filename: 'shared.js',

chunkFilename: '[name].[chunkhash:3].min.js',

},

module: { //在配置文件里添加JSON loader

loaders: [{

test: /\.js$/,

exclude: /^node\_modules$/,

loader: 'babel-loader'

}, {

test: /\.css$/,

exclude: /^node\_modules$/,

loader: ExtractTextPlugin.extract('style', ['css', 'autoprefixer'])

}, {

test: /\.less$/,

exclude: /^node\_modules$/,

loader: ExtractTextPlugin.extract('style', ['css', 'autoprefixer', 'less'])

}, {

test: /\.scss$/,

exclude: /^node\_modules$/,

loader: ExtractTextPlugin.extract('style', ['css', 'autoprefixer', 'sass'])

}, {

test: /\.(eot|woff|svg|ttf|woff2|gif|appcache)(\?|$)/,

exclude: /^node\_modules$/,

loader: 'file-loader?name=[name].[ext]'

}, {

test: /\.(png|jpg|gif)$/,

exclude: /^node\_modules$/,

loader: 'url-loader?limit=8192&name=images/[hash:8].[name].[ext]'

//注意后面那个limit的参数，当你图片大小小于这个限制的时候，会自动启用base64编码图

}, {

test: /\.jsx$/,

exclude: /^node\_modules$/,

//loaders: ['jsx', 'babel'],//1.0的配置

loaders: ['jsx-loader', 'babel-loader'],//2.0++ 必须加上 loader

}]

},

resolve: {

//extensions: ['', '.js', '.json'],

extensions: ['.js', '.json'],//webpack 2.0以上的配置 2017-11-03改

},

postcss: [

require('autoprefixer') //调用autoprefixer插件,加入各个浏览器的前缀

],

plugins: [

new HtmlWebpackPlugin({

template: './index.html'

}),

new ExtractTextPlugin('[name]-[hash:3].css'), //css随机数

new webpack.HotModuleReplacementPlugin(), //热加载插件

new webpack.DefinePlugin({

'process.env.NODE\_ENV': '"development"'

}),

new webpack.NoErrorsPlugin()

]

};

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68

**3、webpack服务文件：dev-serve.js**

var config = require("./webpack.config.js");

var webpack = require('webpack');

var WebpackDevServer = require('webpack-dev-server');

config.entry.app.unshift("webpack-dev-server/client?http://localhost:8099/");

var compiler = webpack(config);

var server = new WebpackDevServer(compiler, {

publicPath: config.output.publicPath,

stats: {

colors: true

}

});

server.listen(8099);

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14

这里注意下内容，



我的app.js是这么配置的，很简单，

**app.js文件**

import React from 'react'

import { render } from 'react-dom'

import { Router, Route, IndexRoute, Link, IndexLink, hashHistory } from 'react-router'

alert("AAA");

* 1
* 2
* 3
* 4
* 5

**安装完成webpack命令之后，**

**运行 npm start 命令 ，即可实现本地实施同步开发！**

**接下来就编写你的代码即可！**

全部代码在这里：[在这里](http://download.csdn.net/detail/qq_16559905/9731587)

**解决方案二：（推荐使用）**

**操作步骤：**

1、安装完Node之后，为了保证速度，需要使用淘宝镜像，命令如下：

（1）npm config set registry "http://registry.npm.taobao.org"

（2）npm config list 可以查看配置

* 1
* 2
* 3

2.安装完nodejs之后,打开终端,执行命令:

(1)npm install webpack -g

//-g的意思是安装全局的webpack,但是我们实际的开发中还需要针对单个的项目进行webpack安装,执行命令:

* 1
* 2
* 3
* 4

3、使用 npm init 初始化，生成 package.json 文件：执行命令:

1. npm init 自定义创建package.json

2. npm init -yes 可以创建默认的package.json

* 1
* 2

现在我们的项目已经创建好了,接下来我们来添加依赖包及插件

**（1） 局部安装Webpack,执行命令：**

npm install webpack --save-dev

* 1

**（2）安装React,–save 命令用于将包添加至 package.json 文件,执行命令：**

npm install react react-dom react-router react-hot-loader css-loader jsx-loader html-webpack-plugin --save-dev

* 1

**（3） 安装babel插件,babel插件是webpack需要的加载器,如果没有这几个加载器我们的jsx语法，和es2015语法就会报错。**

npm install babel-loader babel-core babel-preset-es2015 babel-preset-react babel-preset-stage-0 --save-dev

* 1

**（4）安装自动刷新热更新服务，安装webpack-dev-server执行命令：**

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

* 1

**（5）在package.json文件中为scripts添加,方便使用开启服务命令:**

"scripts": {

"build": "webpack",

"dev": "webpack-dev-server --devtool eval --progress --colors --content-base build"

}

* 1
* 2
* 3
* 4

**package.json全部文件附上：**

{

"name": "yubai8",

"version": "1.1.1",

"main": "index.js",

"dependencies": {},

"devDependencies": {

"html-webpack-plugin": "^2.26.0",

"webpack-dev-server": "^1.16.2"

},

"scripts": {

"build": "webpack",

"dev": "webpack-dev-server --devtool eval --progress --colors --content-base build"

},

"author": "",

"license": "ISC",

"keywords": [],

"description": ""

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19

这里有一点提醒大家，package.json中name不能跟我们的模块和项目文件目录同名！

**安装完命令之后，创建webpack的配置文件：webpack.config.js文件**

**webpack.config.js文件配置如下：**

var path = require('path'),

webpack = require('webpack'),

HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {

entry: ['webpack/hot/dev-server', path.resolve(\_\_dirname, './app.js')],

output: {

path: path.resolve(\_\_dirname, './\_\_build\_\_'),

filename: 'bundle.js'

},

devServer: {

inline: true,

port: 8099

},

module: {

loaders: [{

test: /\.js?$/,

exclude: /(node\_modules|bower\_components)/,

loader: 'babel',

query: {

presets: ['es2015', 'react']

}

}]

},

plugins: [

new HtmlWebpackPlugin({

template: './index.html'

}),

new webpack.HotModuleReplacementPlugin()

]

};

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31

上面第五行 **‘./app.js’** 是你的js入口文件

解决方案二：[链接地址](http://download.csdn.net/detail/qq_16559905/9732173)

**安装完成之后运行命令**

1、根目录下执行命令,其中一个：

npm run build 线上目录

npm run dev 开发目录

* 1
* 2
* 3

2.浏览器直接访问:

<http://localhost:8099/index.html>

ok，完成！

2017-11加

**webpack最近更新版在** [在这里](http://blog.csdn.net/qq_16559905/article/details/78277642)