

(框架) VUE.JS  
(202) vue-loader

(02 专区) 【vue】vue组件化开发初体验-示例vue-loader-example学习记录 (丁一鸣) (上)

官网文档: <http://vuejs.github.io/vue-loader/>

## 【vue】vue组件化开发初体验-示例vue-loader-example学习记录



丁一鸣 2015年11月28日 发布

<http://segmentfault.com/a/1190000004060034>

vue组件化开发初体验-示例vue-loader-example学习记录

<https://github.com/dingyiming/learn-Js-vuejs/tree/master/demos/demo-vue-loader-example/demo1>

## vue组件化开发初体验-示例vue-loader-example学习记录

- 来自vue官方示例: <https://github.com/vuejs/vue-loader-example>
- 使用了vuejs和webpack, 以及一系列webpack加载器, 如vue-loader
- 依葫芦画瓢, 试了试, 有些具体分析还需要再学习学习, 先简单记录下具体的做法, 屡屡思绪。

### 目录结构

```
- demo/  
  + package.json //npm配置文件  
  + webpack.config.js //webpack配置  
  + index.html //页面  
- node_modules //npm加载的模块  
- src //开发资源目录  
  - assets //一些资源  
    + logo.png //图片资源  
  - components //vue组件  
    + a.vue  
    + b.vue  
    + counter.vue  
  + app.vue //布局文件  
  + main.js //入口文件
```

### 初始化npm

#### 1.生成npm配置文件 package.json

```
npm init
```

2.可以粘贴如下配置内容替换到package.json中，或者根据变动进行修改

```
{
  "name": "demo_vue-loader-example",
  "version": "0.0.1",
  "description": "demo",
  "main": "index.js",
  "scripts": {
    "dev": "webpack-dev-server --inline --hot --quiet",
    "build": "export NODE_ENV=production && webpack --progress --hide-modules"
  },
  "author": "dingyiming",
  "license": "MIT",
  "devDependencies": {
    "babel-core": "^6.2.1",
    "babel-loader": "^6.2.0",
    "babel-plugin-transform-runtime": "^6.1.18",
    "babel-preset-es2015": "^6.1.18",
    "babel-preset-stage-0": "^6.1.18",
    "babel-runtime": "^6.2.0",

    "css-loader": "^0.23.0",
    "node-sass": "^3.4.2",
    "sass-loader": "^3.1.2",
    "style-loader": "^0.13.0",
    "stylus-loader": "^1.4.2",

    "file-loader": "^0.8.5",
    "jade": "^1.11.0",
    "template-html-loader": "0.0.3",

    "vue-hot-reload-api": "^1.2.1",
    "vue-html-loader": "^1.0.0",
    "vue-loader": "^7.1.4",

    "webpack": "^1.12.9",
    "webpack-dev-server": "^1.14.0"
  },
  "dependencies": {
    "vue": "^1.0.10"
  }
}
```

3.下载node模块

```
npm install
```

- 其实我是一条条用 `npm i xxx --save-dev` 敲的，没有在 `package.json` 里面的 `"devDependencies": {}` 手动添加内容，可以一个个装（也可以一起敲）如 `npm i webpack --save-dev`, `npm i vue --save`
- `--save-dev` 把依赖名和版本要求放在了 `"devDependencies": {}`,
- `--save` 放在了 `"dependencies": {}`
- 每敲一个下载完后可以看到 `"devDependencies": {}` 依赖内容的添加

## 新建index.html用于展示最终页面

---

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Vue Webpack Example</title>
</head>
<body>
<app></app>
<script src="dist/build.js"></script>
</body>
</html>
```

## 新建webpack.config.js用于配置webpack

---

```
var webpack = require('webpack')

module.exports = {
  entry: './src/main.js',
  output: {
    path: './dist',
    publicPath: 'dist/',
    filename: 'build.js'
  },
  module: {
    loaders: [
      {
        test: /\.vue$/,
        loader: 'vue'
      },
      {
        // edit this for additional asset file types
        test: /\. (png|jpg|gif)$/,
        loader: 'file?name=[name].[ext]?[hash]'
      }
    ]
  },
  // example: if you wish to apply custom babel options
  // instead of using vue-loader's default:
  babel: {
    presets: ['es2015', 'stage-0'],
    plugins: ['transform-runtime']
  }
}

if (process.env.NODE_ENV === 'production') {
  module.exports.plugins = [
    new webpack.DefinePlugin({
      'process.env': {
        NODE_ENV: '"production"'
      }
    }),
    new webpack.optimize.UglifyJsPlugin({
```

```

      compress: {
        warnings: false
      }
    })),
    new webpack.optimize.OccurenceOrderPlugin()
  ]
} else {
  module.exports.devtool = '#source-map'
}

```

## 新建src目录用于存放开发文件

---

### 新建入口文件main.js

```

var Vue = require('vue')
var App = require('./app.vue')

new Vue({
  el: 'body',
  components: {
    app: App
  }
})

```

### 新建组件布局文件app.vue

组件布局将在这里设置，.vue文件将由vue-loader进行加载，.vue内同时包含html、css、js源码，使组件的独立，组件之间可以尽可能地解耦，便于开发维护

```

<template lang="jade">
div
  img(class="logo", src="./assets/logo.png")
  h1 {{msg}}
  comp-a
  comp-b
  counter
</template>

<script>
import CompA from './components/a.vue'
import CompB from './components/b.vue'
import Counter from './components/counter.vue'
export default {
  data () {
    return {
      msg: 'Hello from vue-loader!'
    }
  },
  components: {
    CompA,
    CompB,
    Counter
  }
}

```

```

    }
  }
</script>

<style lang="stylus">
font-stack = Helvetica, sans-serif
primary-color = #999
body
  font 100% font-stack
  color primary-color
.logo
  width 40px
  height 40px
</style>

```

## 新建components文件夹

用于开发具体的子组件，均以.vue的后缀呈现

- a.vue

```

<style scoped>
.container {
  border: 1px solid #00f;
}
.red {
  color: #f00;
}
</style>

<template>
  <div class="container">
    <h2 class="red">{{msg}}</h2>
  </div>
</template>

<script>
export default {
  data () {
    return {
      msg: 'Hello from Component A!'
    }
  }
}
</script>

```

- b.vue

```

<style scoped>
.container {
  border: 1px solid #f00;
}
h2 {
  color: #393;
}

```

```
}  
</style>  
  
<template>  
  <div class="container">  
    <h2>Hello from Component B!</h2>  
  </div>  
</template>
```

- counter.vue

```
<template>  
  <div>  
    <h1>I am a Counter Component. Edit me in dev mode.</h1>  
    <p>Current count: {{count}}</p>  
  </div>  
</template>  
  
<script>  
export default {  
  data () {  
    return { count: 0 }  
  },  
  ready () {  
    this.handle = setInterval(() => {  
      this.count++  
    }, 1000)  
  },  
  destroyed () {  
    clearInterval(this.handle)  
  }  
}  
</script>
```

新建**assets**文件夹用于放一些资源



- 此项目下有一张图

## 打包运行查看

- 打包:

```
npm run build
```

