

(框架) VUE.JS
(202) vue-loader

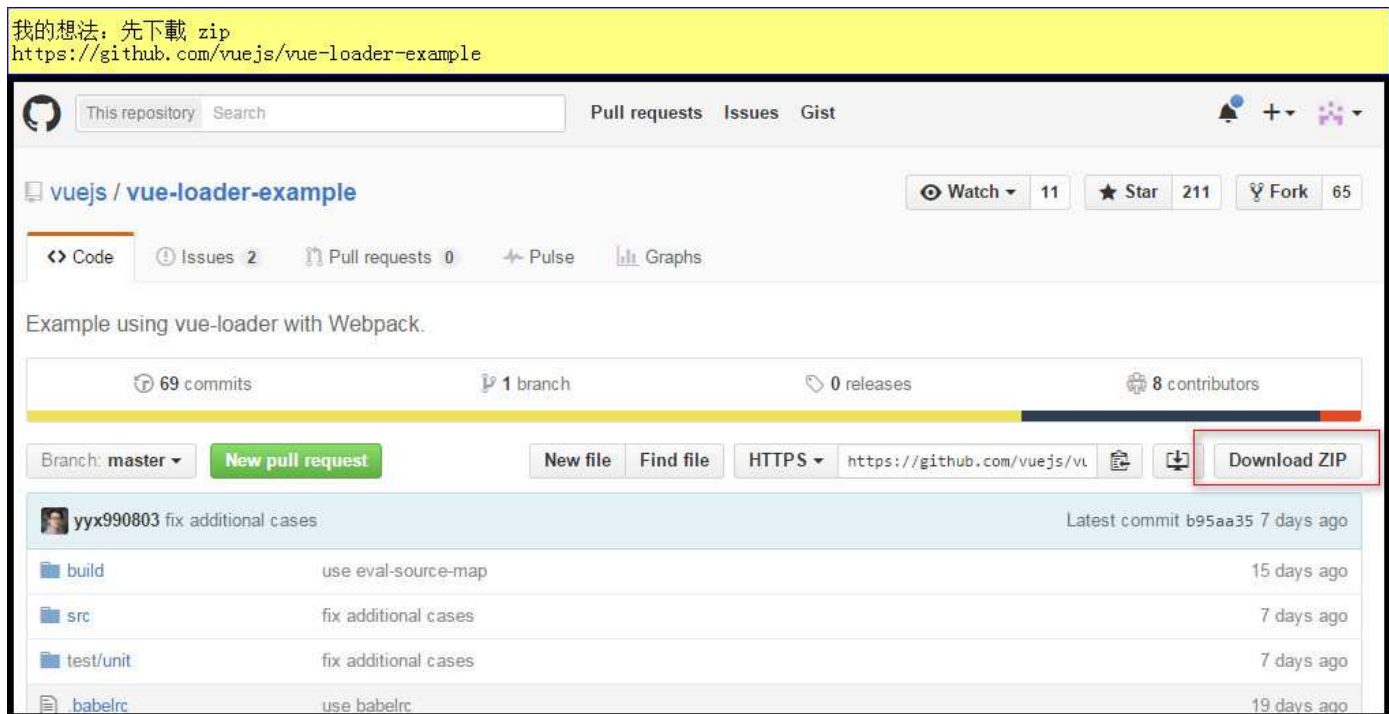
(02 专区) 【vue】vue组件化开发初体验-示例vue-loader-example学习记录（丁一鸣）实作（下）

vue-load 结合 webpack，非常难以理解
直接先下载 zip，会比一个档案一个档案建立，来的有效率。

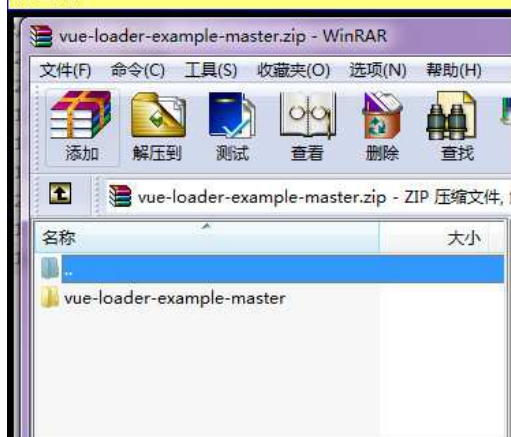
丁一鸣先生，辛苦写的文章，我无意修改，仅配上图片。

画蛇添足一下。

我的思想，以省时间为主，不懂的地方，还是以群里讨论为主。



解壓縮



解壓縮後，檢查大小 100kb



解壓縮後的目錄長這樣

vue作者：安裝說明如下

```
# install dependencies
```

```
npm install
```

```
# serve with hot reload at localhost:8080
```

```
npm run dev
```

```
# build for production with minification
```

```
npm run build
```

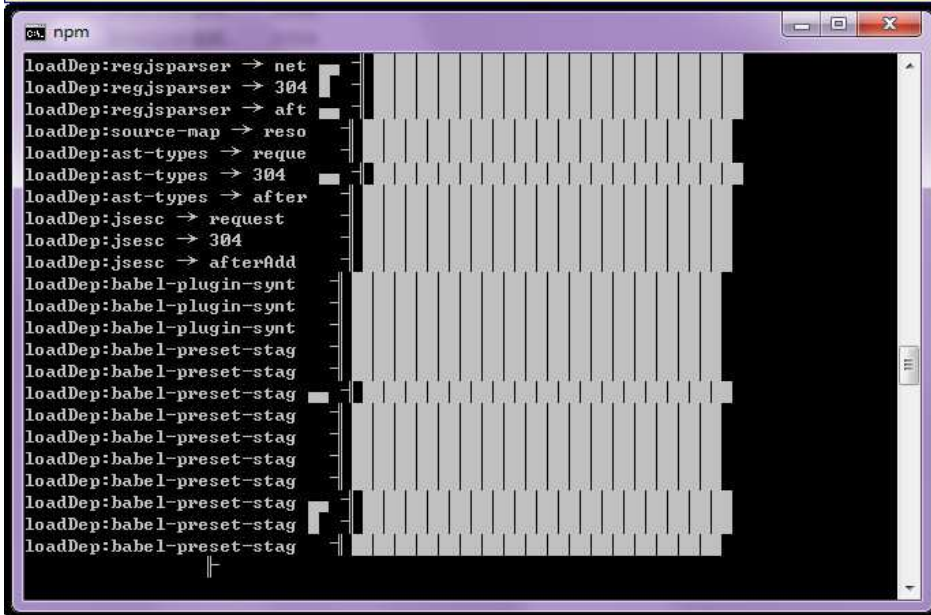
```
# run unit tests
```

```
npm test
```

build	2016/1/14 星期三	文件夹	
src	2016/1/14 星期三	文件夹	
test	2016/1/14 星期三	文件夹	
.babelrc	2016/1/6 星期三	BABELRC 文件	1 KB
.editorconfig	2016/1/6 星期三	EDITORCONFIG ...	1 KB
.eslintrc	2016/1/6 星期三	ESLINTRC 文件	5 KB
.gitignore	2016/1/6 星期三	文本文档	1 KB
circle.yml	2016/1/6 星期三	YML 文件	1 KB
index.html	2016/1/6 星期三	360 se HTML Do...	1 KB
package.json	2016/1/6 星期三	JSON File	2 KB
README.md	2016/1/6 星期三	MD 文件	2 KB

npm install

畫面長這樣



下載要 5、6分鐘

```
管理员: C:\Windows\system32\cmd.exe

accepts@1.2.13
  negotiator@0.5.3
  batch@0.5.2
  mime-types@2.1.9
sockjs@0.3.15
  faye-websocket@0.9.4
    websocket-driver@0.6.4
      websocket-extensions@0.1.1
sockjs-client@1.0.3
  eventsource@0.1.6
    original@1.0.0
  faye-websocket@0.7.3
  json3@3.3.2
  url-parse@1.0.5
    querystringify@0.0.3
    requires-port@1.0.0
stream-cache@0.0.2
strip-ansi@3.0.0
supports-color@3.1.2

runTopLevelLifecycles
npm WARN EPACKAGEJSON vue-loader-example@3.0.0 No repository field.
runTopLevelLifecycles

E:\vue02-loader-example-master>
```

將近100mb

vue02-loader-example-master 属性

常规 共享 安全 以前的版本 自定义

vue02-loader-example-master

类型: 文件夹

位置: E:\

大小: 58.1 MB (60,947,628 字节)

占用空间: 92.2 MB (96,690,176 字节)

包含: 11,996 个文件, 1,966 个文件夹

创建时间: 2016年1月14日 今天, 7 分钟之前

package.json - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

package.json

<No Schema Selected>

```
{
  "name": "vue-loader-example",
  "version": "3.0.0",
  "description": "Example using Webpack with vue-loader",
  "main": "index.js",
  "scripts": {
    "dev": "webpack-dev-server --inline --hot --config build/webpack.dev.config.js",
    "build": "webpack --progress --hide-modules --config build/webpack.prod.config.js",
    "test": "karma start build/karma.conf.js"
  },
  "dependencies": {
    "vue": "^1.0.0"
  },
  "devDependencies": {
    "babel-core": "^6.1.2",
    "babel-loader": "^6.1.0",
    "babel-plugin-transform-runtime": "^6.1.2",
    "babel-preset-es2015": "^6.1.2",
    "babel-preset-stage-0": "^6.1.2",
    "babel-runtime": "^5.8.0",
    "css-loader": "^0.23.0",
    "eslint": "^1.10.3",
    "eslint-loader": "^1.2.0",
    "file-loader": "^0.8.4",
    "function-bind": "^1.0.2",
    "inject-loader": "^2.0.1",
    "jade": "^1.11.0",
    "jasmine-core": "^2.4.1",
    "karma": "^0.13.15",
    "karma-jasmine": "^0.3.6",
    "karma-phantomjs-launcher": "^0.2.1",
    "karma-spec-reporter": "0.0.23",
    "karma-webpack": "^1.7.0",
    "phantomjs": "^1.9.19",
    "stylus-loader": "^1.4.0",
    "template-html-loader": "0.0.3",
    "url-loader": "^0.5.7",
    "vue-hot-reload-api": "^1.2.0",
    "vue-html-loader": "^1.0.0",
    "vue-loader": "^8.0.0",
    "vue-style-loader": "^1.0.0",
    "webpack": "^1.12.2",
    "webpack-dev-server": "^1.12.0"
  },
  "author": "Evan You",
  "license": "MIT"
}
```

執行前要看一下 package.json

npm run dev
畫面如下

npm

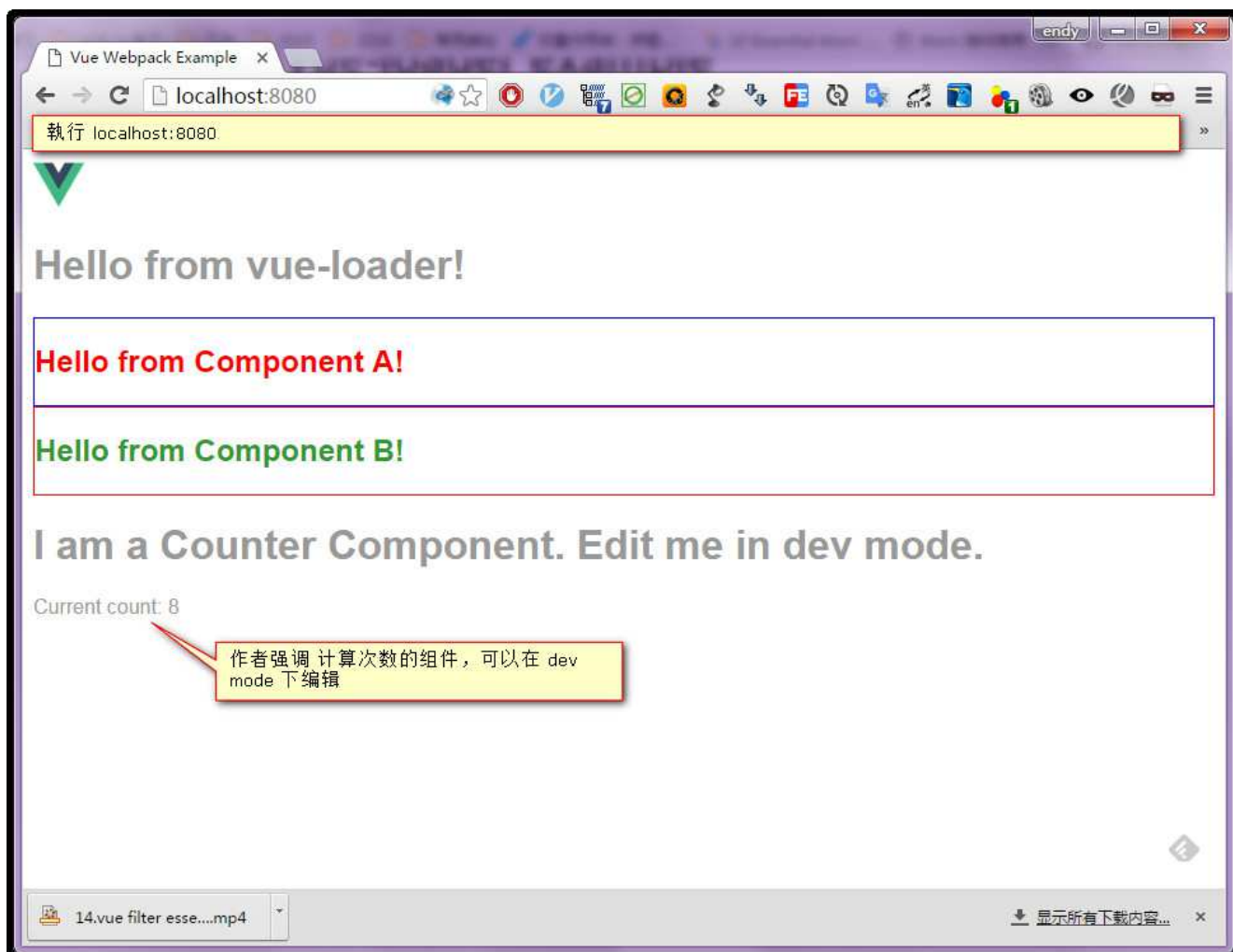
```
events@0.1.6
├── original@1.0.0
├── faye-websocket@0.7.3
├── json3@3.3.2
├── url-parse@1.0.5
├── querystringify@0.0.3
├── requires-port@1.0.0
├── stream-cache@0.0.2
├── strip-ansi@3.0.0
└── supports-color@3.1.2

runTopLevelLifecycles
npm WARN EPACKAGEJSON vue-loader-example@3.0.0 No repository field.
runTopLevelLifecycles

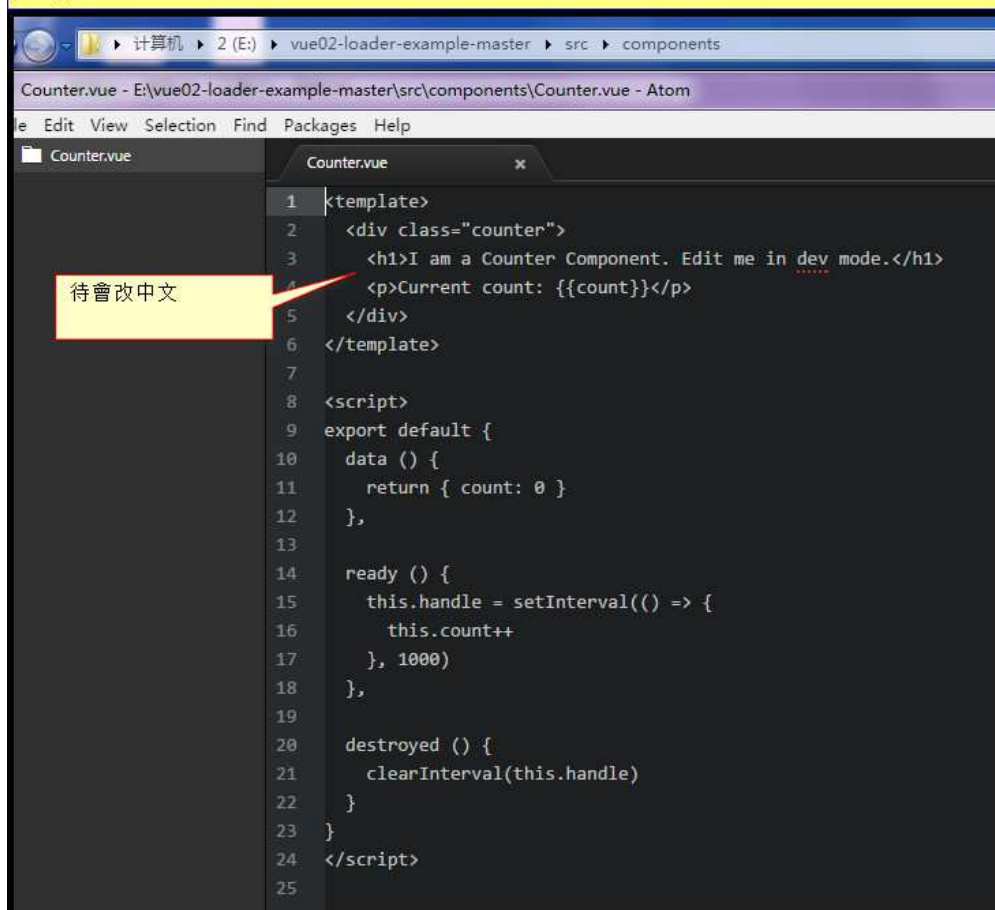
E:\vue02-loader-example-master>npm run dev

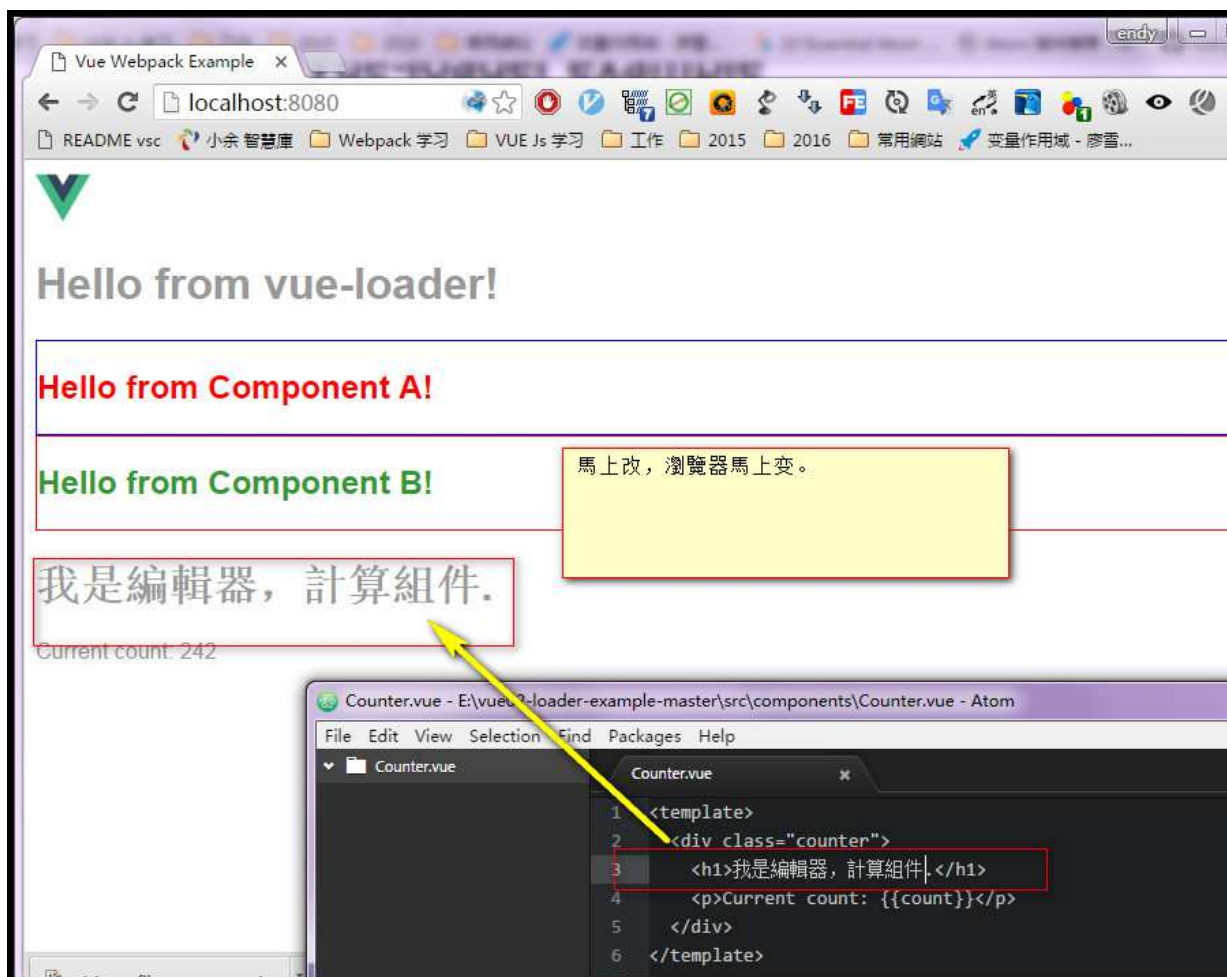
> vue-loader-example@3.0.0 dev E:\vue02-loader-example-master
> webpack-dev-server --inline --hot --config build/webpack.dev.config.js

http://localhost:8080/
webpack result is served from /dist/
content is served from E:\vue02-loader-example-master
```

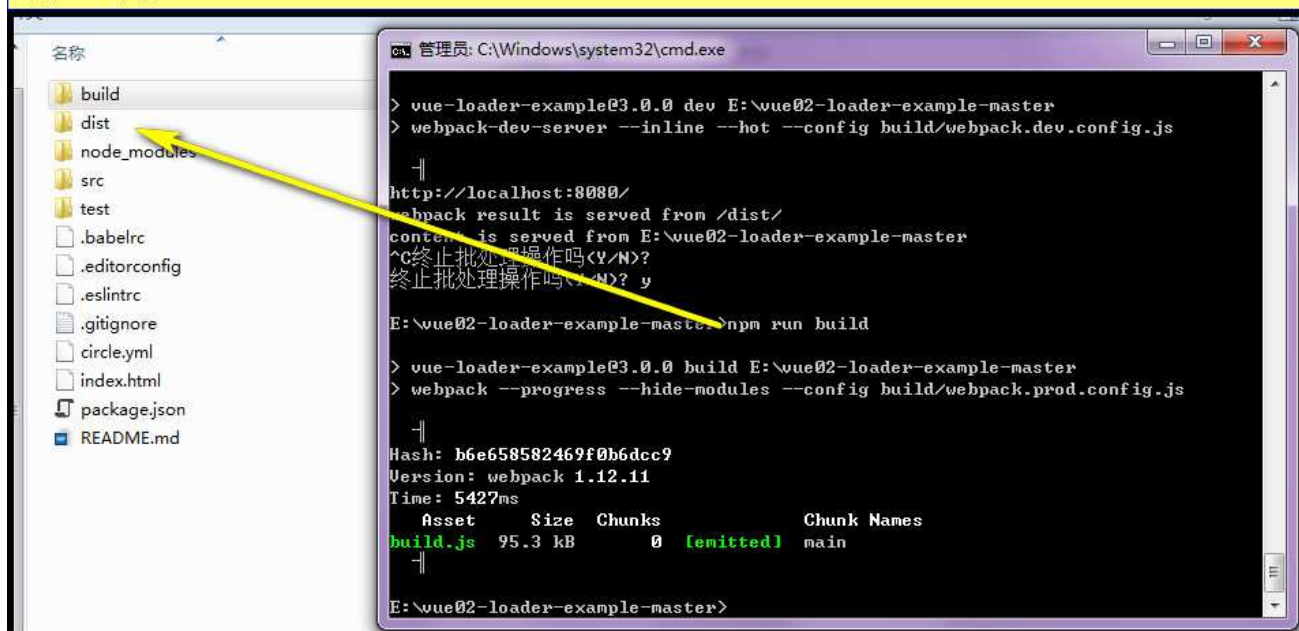


counter.vue
在 /src/components 目錄下
長這樣



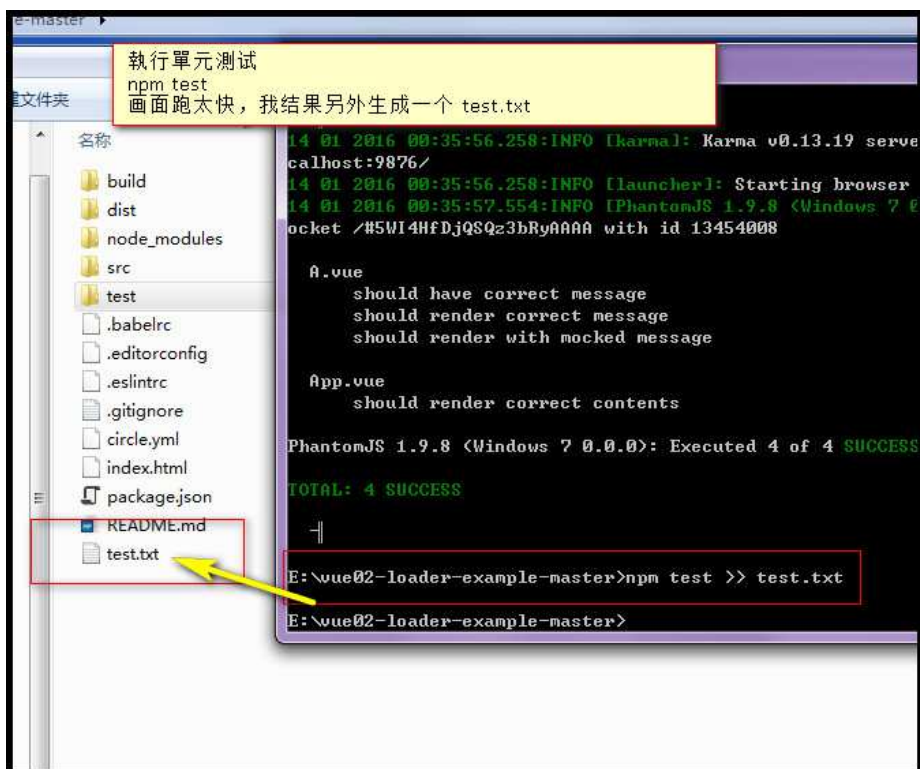


npm run build
生成 dist 目錄

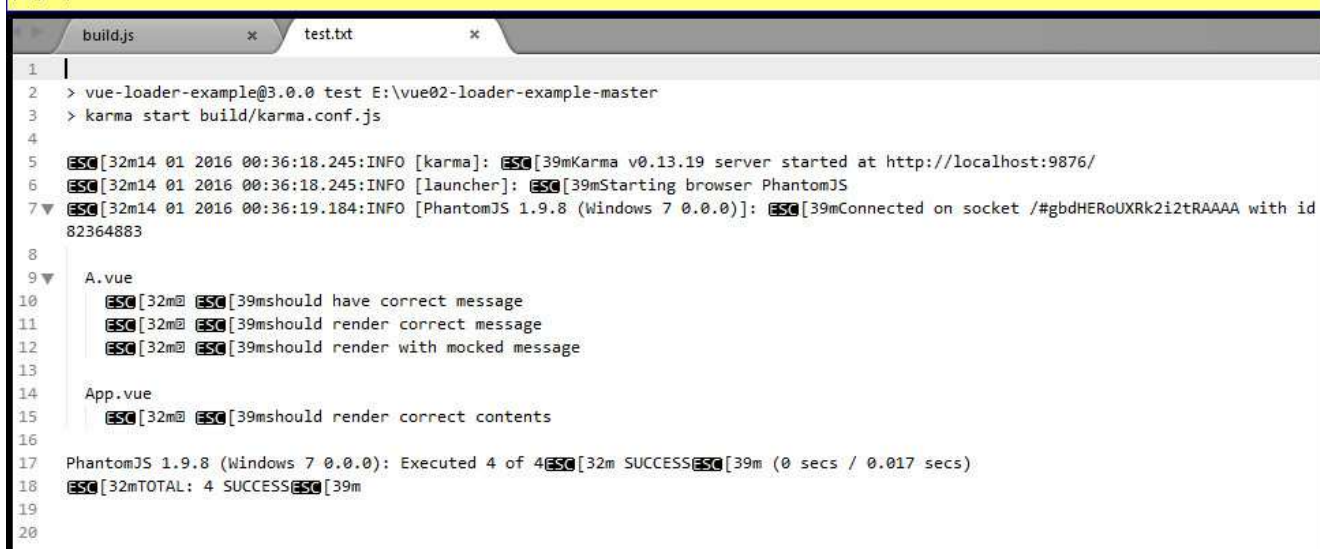


dist 目錄
只有一個檔
build.js 94KB

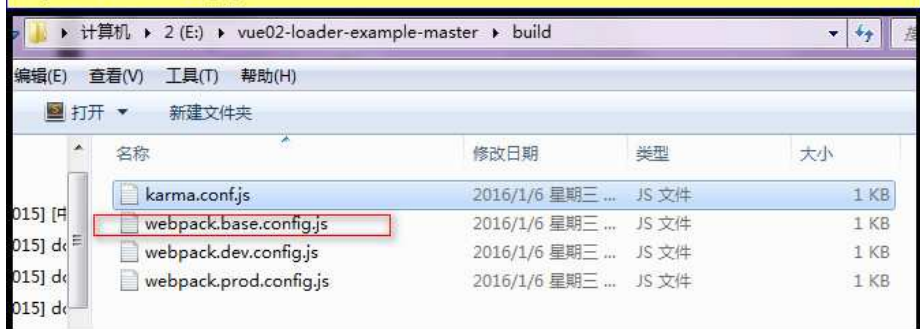


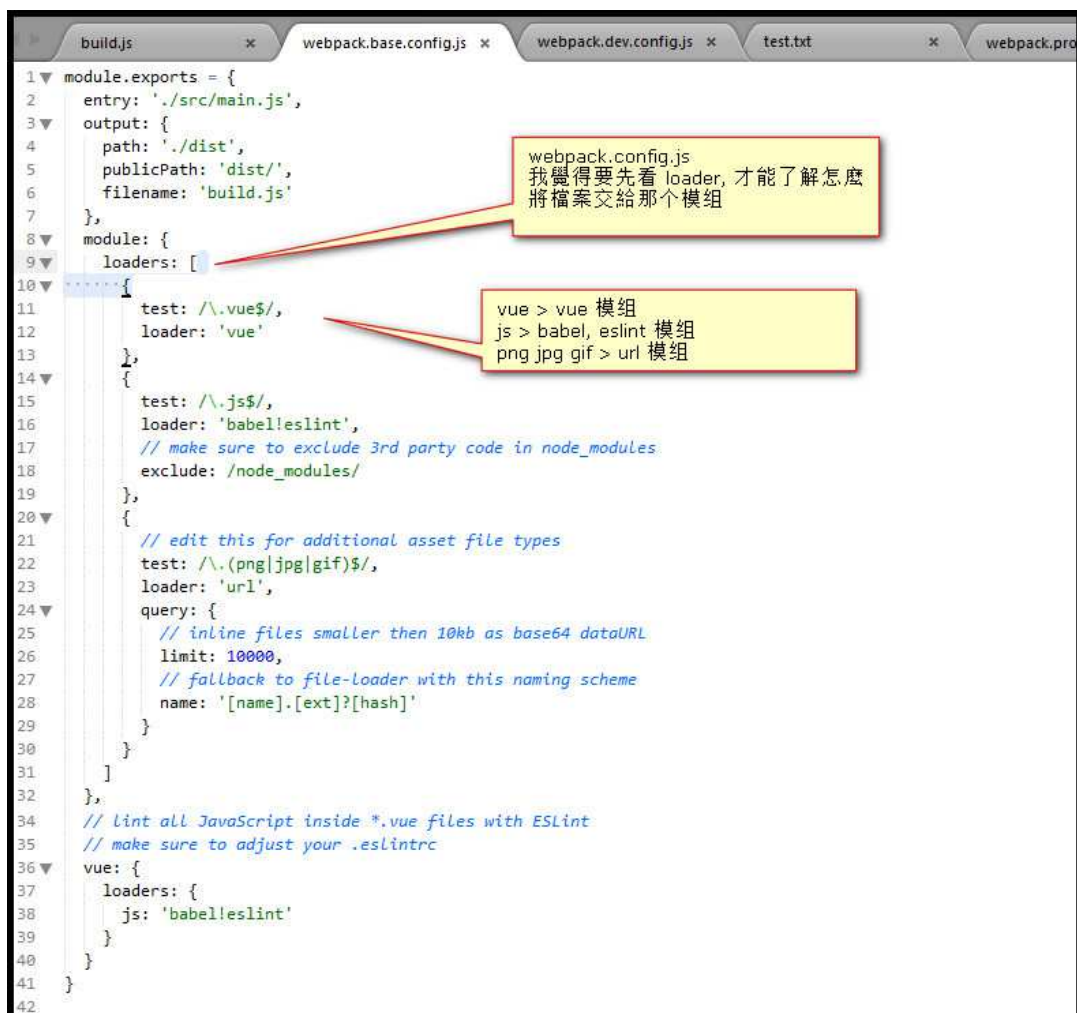


npm test 的結果
我也看不懂。
僅參考



前端工程化，由下列四個檔案，分工完成
最重要的，是
webpack.base.config.js



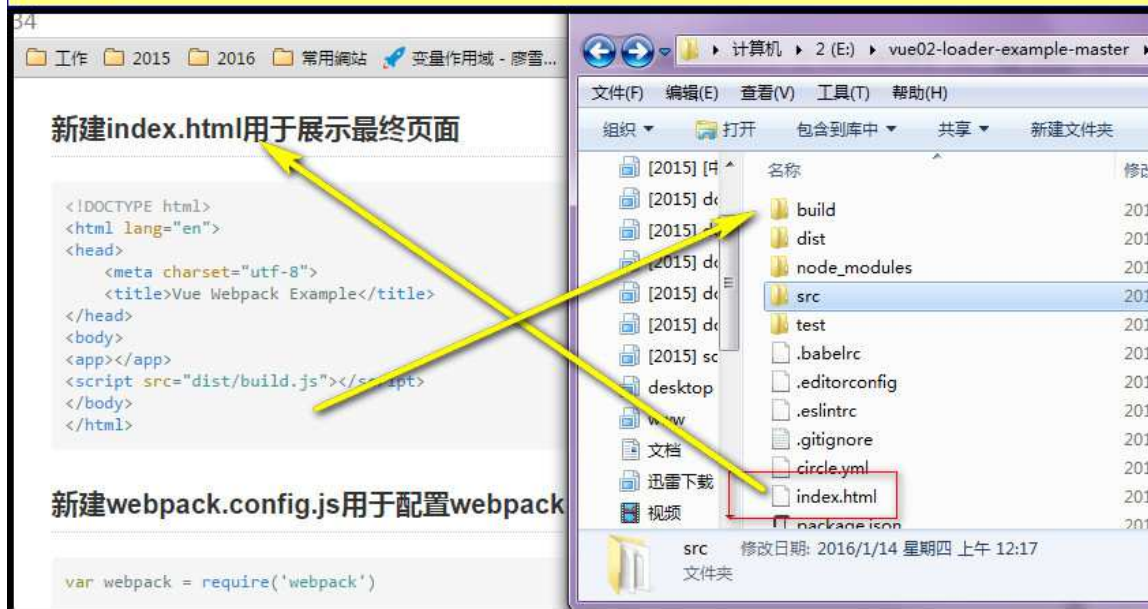


The screenshot shows a code editor with several tabs: build.js, webpack.base.config.js, webpack.dev.config.js, test.txt, and webpack.pro. The main file is webpack.config.js, which is a JavaScript module. It has two callout boxes with Chinese text:

- Box 1 (pointing to line 9): webpack.config.js 我覺得要先看 loader, 才能了解怎麼將檔案交給那個模組
- Box 2 (pointing to line 11): vue > vue 模組
js > babel, eslint 模組
png jpg gif > url 模組

```
1 module.exports = {
2   entry: './src/main.js',
3   output: {
4     path: './dist',
5     publicPath: 'dist/',
6     filename: 'build.js'
7   },
8   module: {
9     loaders: [
10      {
11        test: /\.vue$/,
12        loader: 'vue'
13      },
14      {
15        test: /\.js$/,
16        loader: 'babel!eslint',
17        // make sure to exclude 3rd party code in node_modules
18        exclude: /node_modules/,
19      },
20      {
21        // edit this for additional asset file types
22        test: /\.?(png|jpg|gif)$/,
23        loader: 'url',
24        query: {
25          // inline files smaller than 10kb as base64 dataURL
26          limit: 10000,
27          // fallback to file-loader with this naming scheme
28          name: '[name].[ext]?[hash]'
29        }
30      }
31    ]
32  },
33  // lint all JavaScript inside *.vue files with ESLint
34  // make sure to adjust your .eslintrc
35  vue: {
36    loaders: {
37      js: 'babel!eslint'
38    }
39  }
40 }
41
42
```

單純看教學, 我心想, 沒有路徑關係, 加一下
下次看, 不用思考
/index.html



The screenshot shows a web browser window on the left and a file explorer on the right. The browser window has a yellow background and contains the text:

新建index.html用于展示最终页面

新建webpack.config.js用于配置webpack

The file explorer shows the directory structure of the project. The 'src' folder is selected, and the 'index.html' file is highlighted. A red box is drawn around the 'index.html' file. A yellow arrow points from the 'index.html' file in the file explorer to the 'index.html' text in the browser window.

/webpack.config.js 是文章寫的路徑
實際上，作者工程化的邏輯，看右邊

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'
      }
    ]
  }
}
```

最重要的是這個

vue02-loader-example-master > build

名称	修改日期	类型
karma.conf.js	2016/1/6 星期三 ...	JS 文件
webpack.base.config.js	2016/1/6 星期三 ...	JS 文件
webpack.dev.config.js	2016/1/6 星期三 ...	JS 文件
webpack.prod.config.js	2016/1/6 星期三 ...	JS 文件

作者的邏輯
所有開發中的文件，在 /src 目錄

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处理，组件之间可以尽可能地解耦，便于开发维护

vue02-loader-example-master > src

名称	修改日期
assets	2016/1/14 星期...
components	2016/1/14 星期...
services	2016/1/14 星期...
App.vue	2016/1/6 星期三 ...
main.js	2016/1/6 星期三 ...

vuejs.github.io/vue-loader/configurations/pre-processors.html

如果你先完成上述程序，再來看官網 vue-load. 你会看的比较有逻辑

Using Pre-Processors

In Webpack, all pre-processors need to be applied with a corresponding loader to use other Webpack loaders to process a part of a Vue component. loaders to use from the lang attribute of a language block.

CSS