(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)

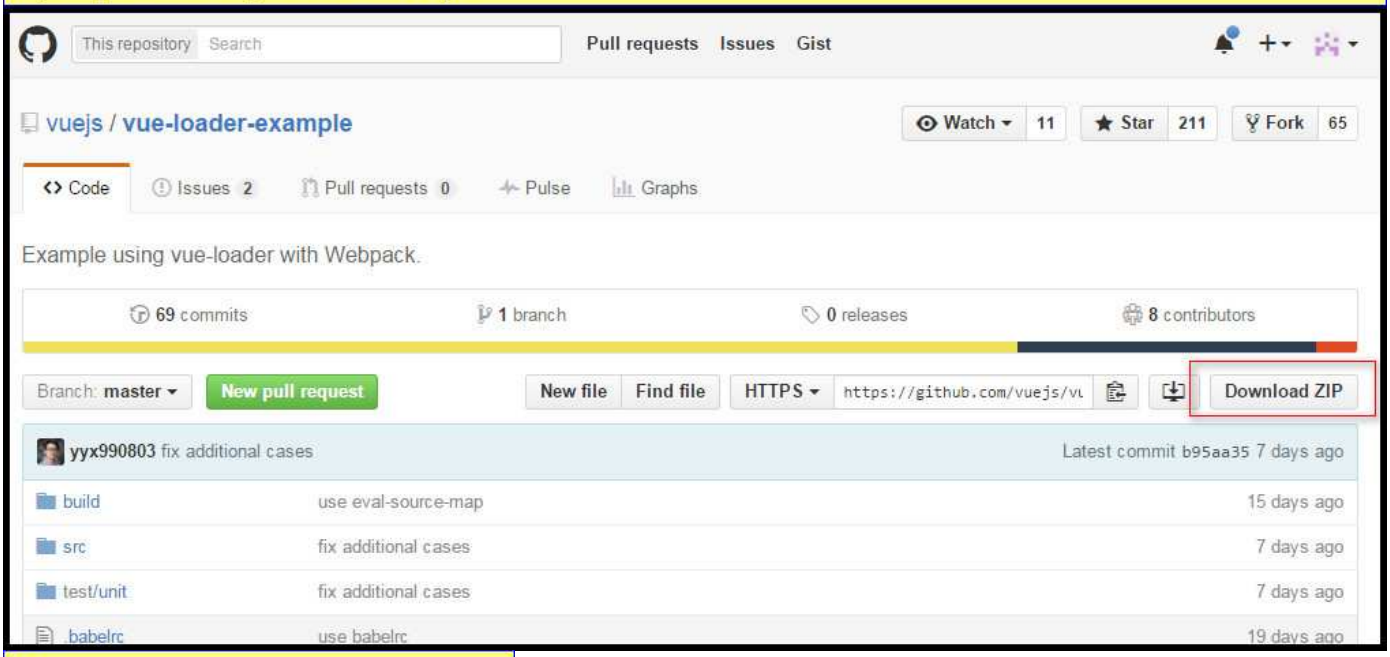
vue-load 结合 webpack，非常难以理解
直接先下载 zip，会比一个檔案一个檔案建立，来的有效率。

丁一鸣先生，辛苦写的文章，我无意修改，仅配上图片。

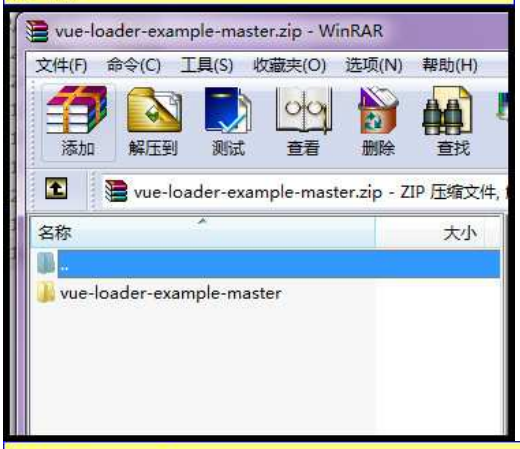
画蛇添足一下。

我的思想，以省时间为主，不懂的地方，还是以群里讨论为主。

我的想法：先下載 zip
https://github.com/vuejs/vue-loader-example



解壓縮

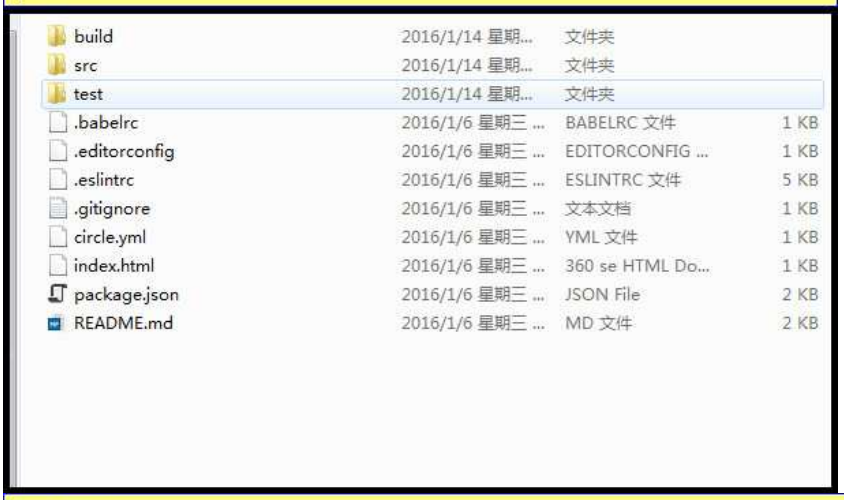


解壓縮後，檢查大小
100kb

解壓縮後的目錄長這樣

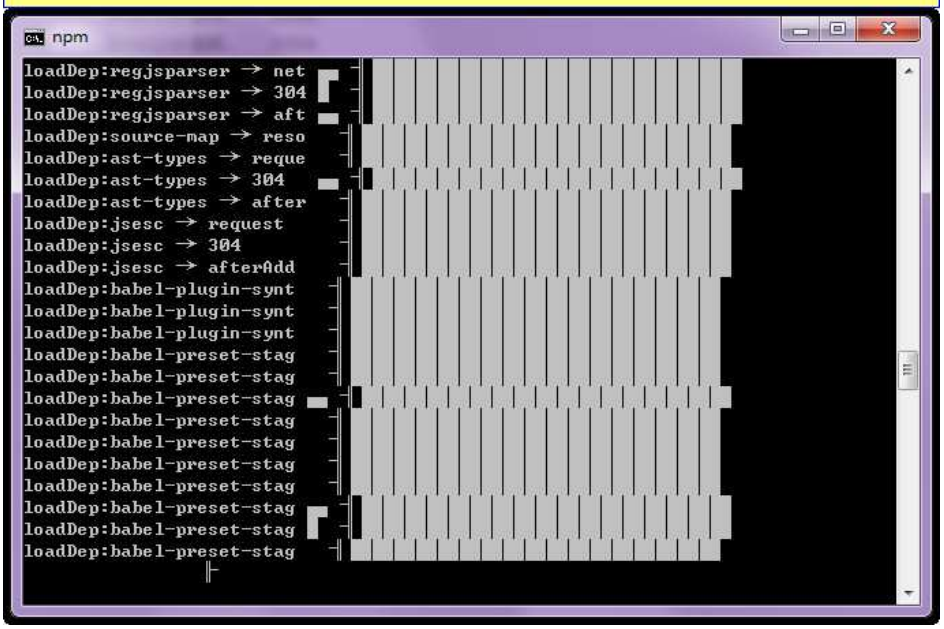
vue作者：安裝說明如下

```
# 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
```

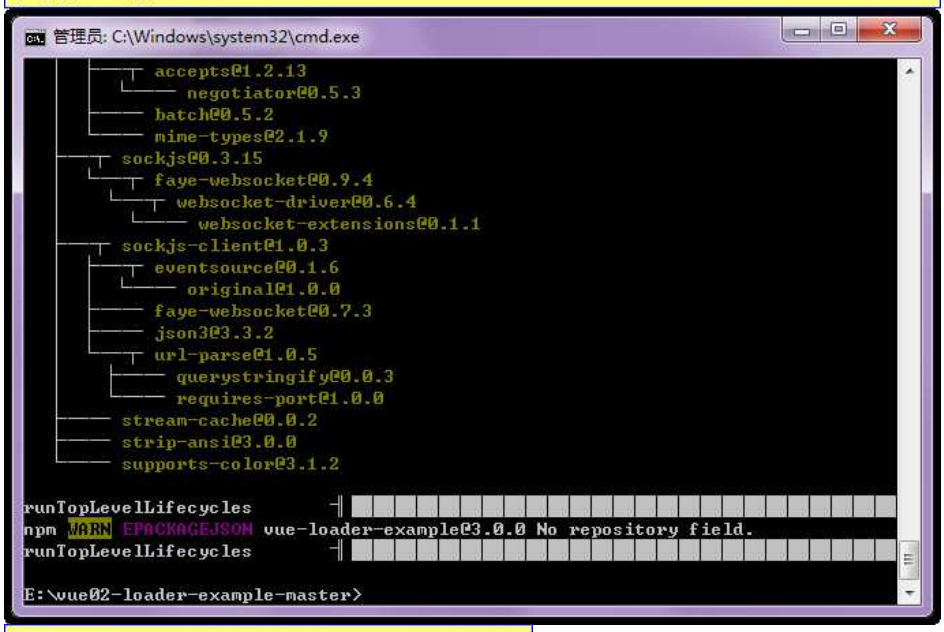



```
npm install
```

畫面長這樣



下載要 5、6分鐘



將近100mb

package.json - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Attach...

package.json

<No Schema Selected>

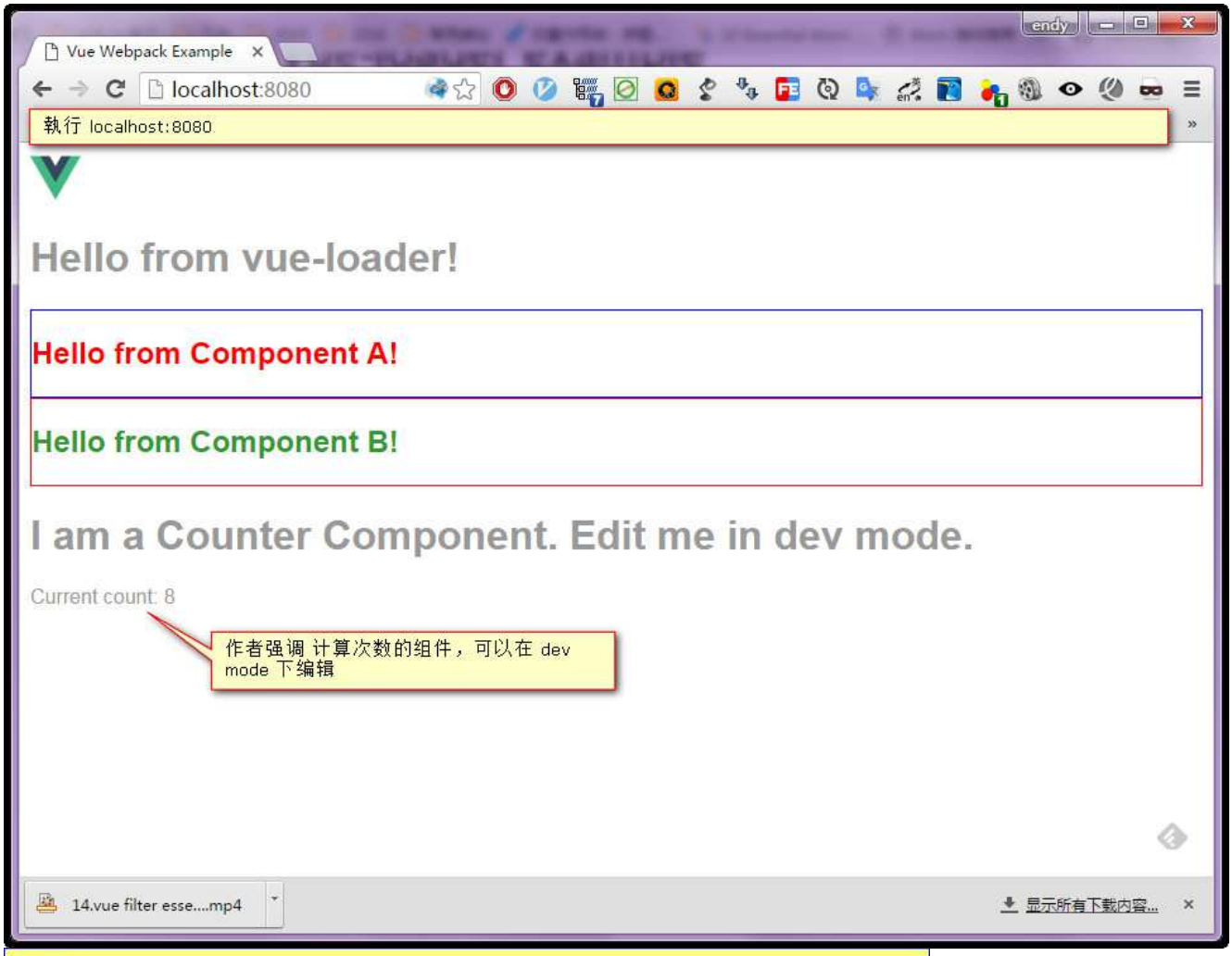
```
{
  "name": "vue-loader-example",
  "version": "3.0.0",
  "description": "Example using Webpack with vue-loader",
  "main": "index.js",
  "scripts": {
    "dev": "webpack-dev-server --inline --hot --config build/webpack.dev.config.js",
    "build": "webpack --progress --hide-modules --config build/webpack.prod.config.js",
    "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

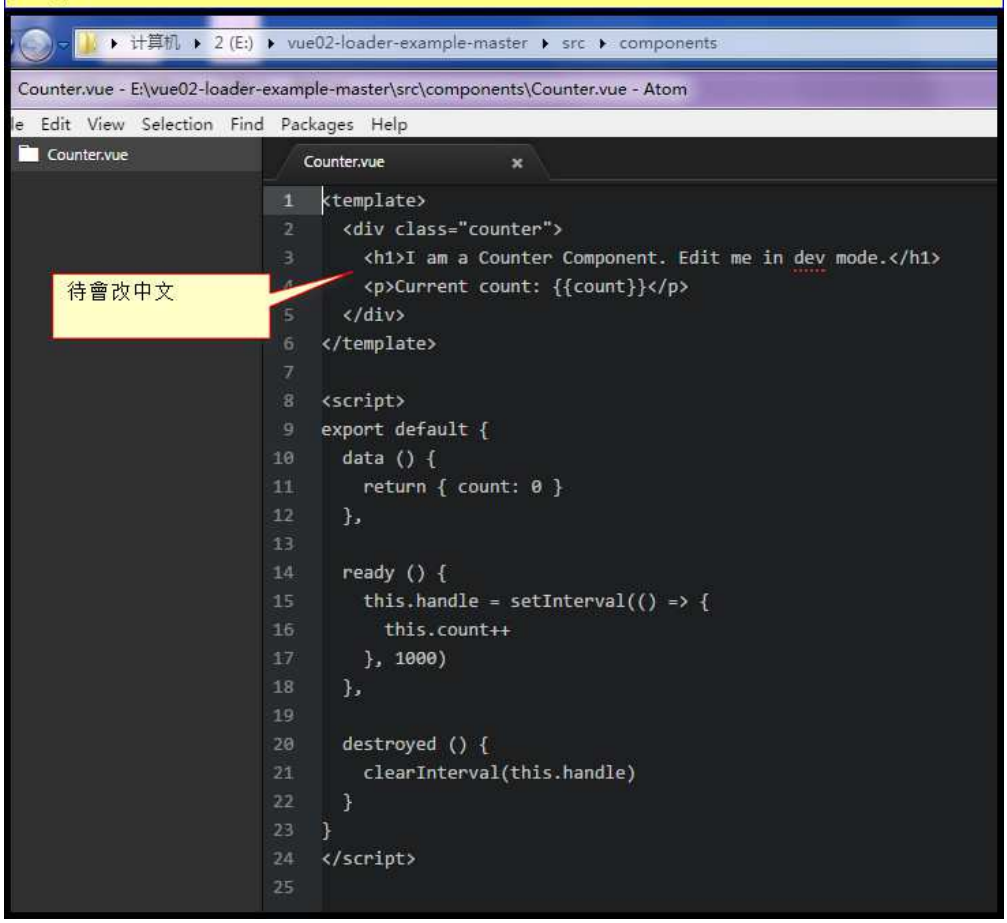
npm run dev
畫面如下

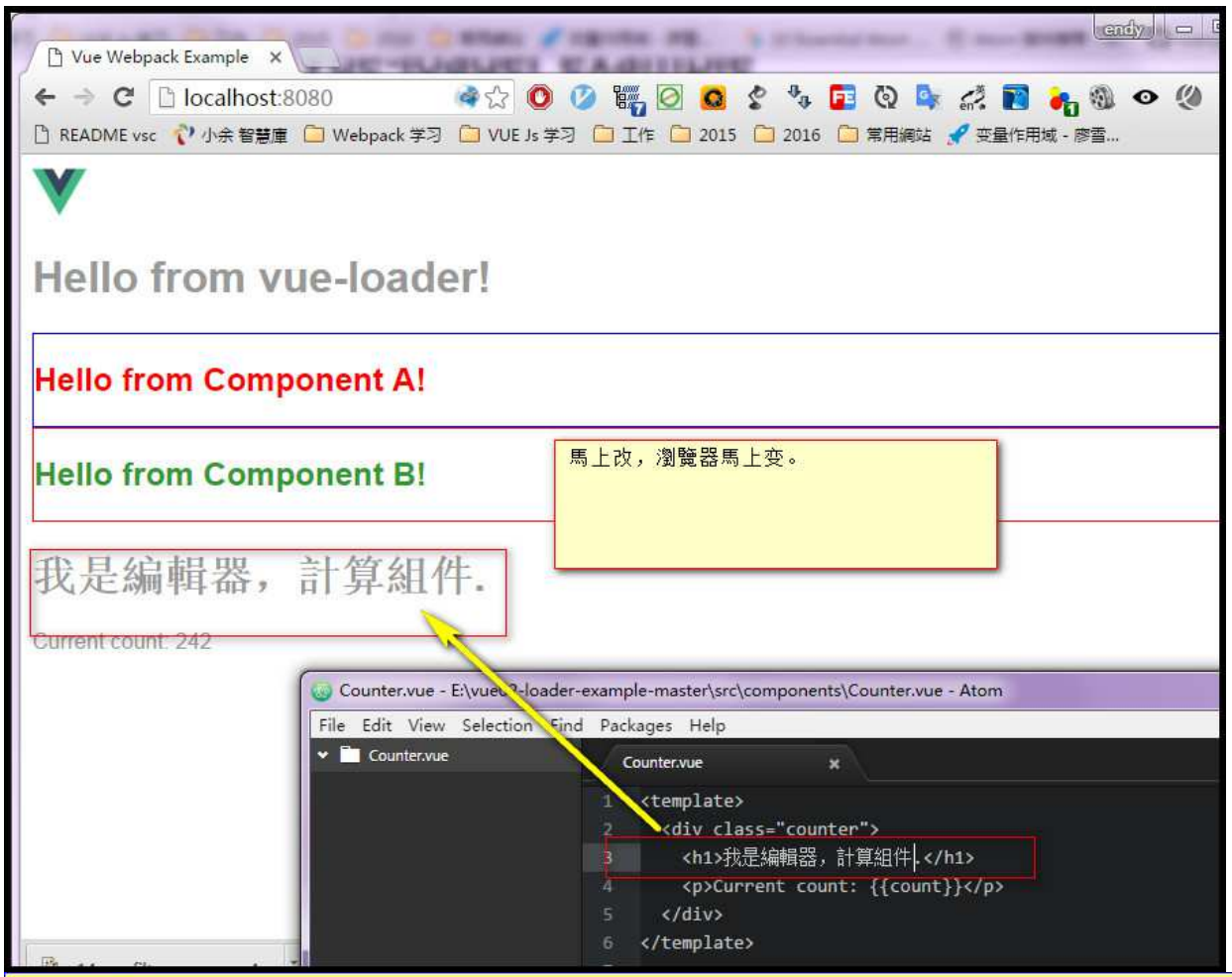
npm

```
├─┬ 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>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
```

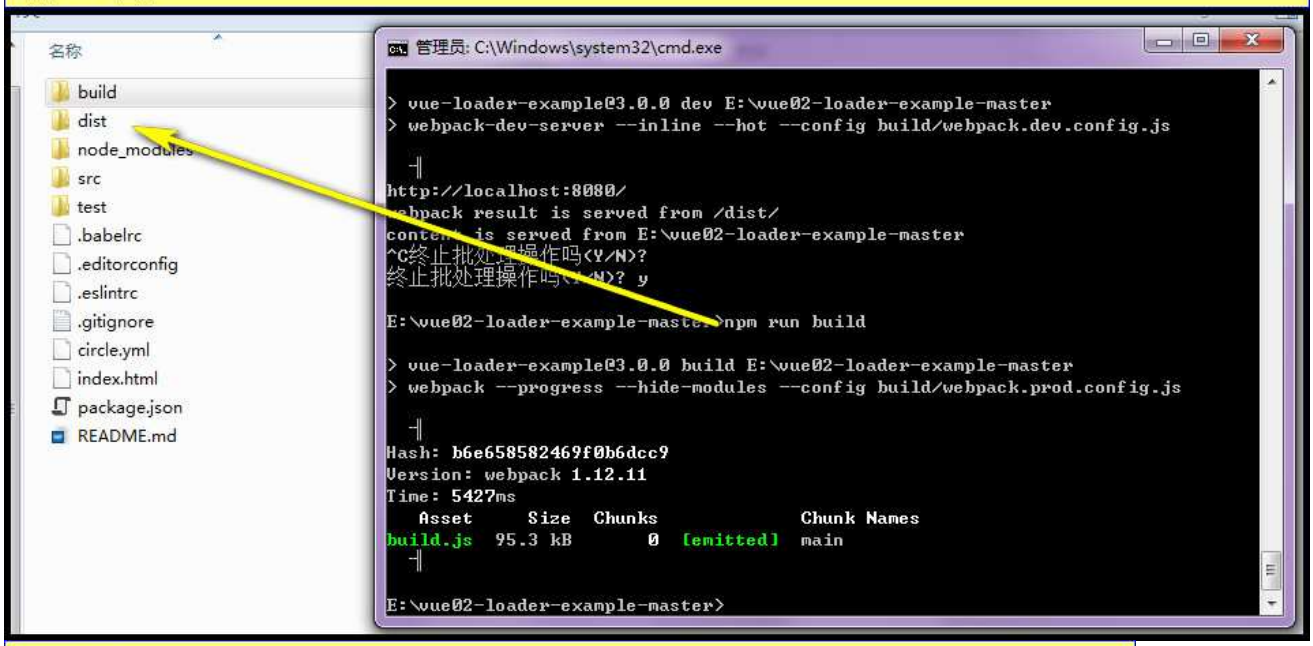


counter.vue
在 /src/components 目錄下
長這樣

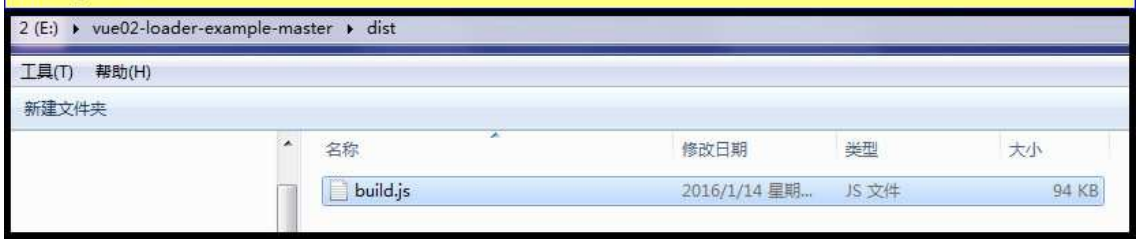


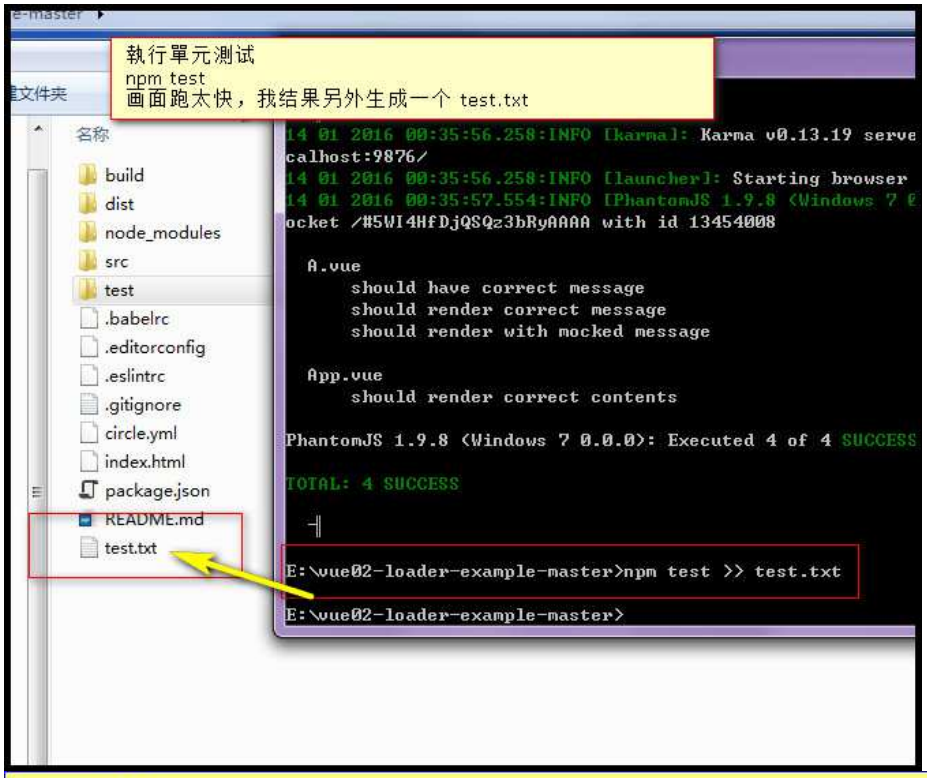


npm run build
生成 dist 目錄



dist 目錄
只有一個檔
build.js 94KB



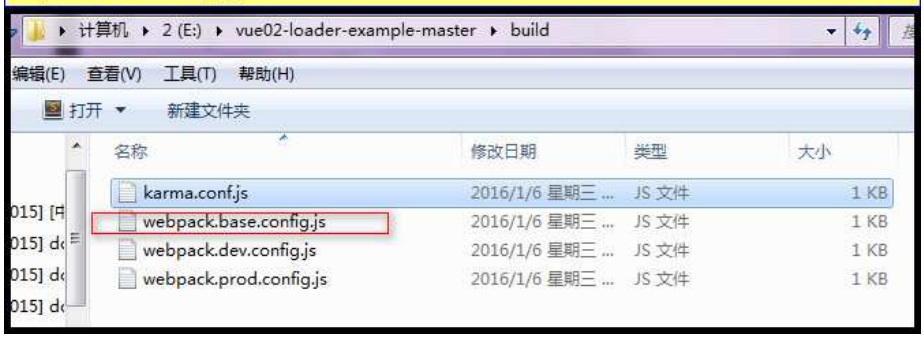


npm test 的結果
我也看不懂。
僅參考

build.js × test.txt ×

```
> vue-loader-example@3.0.0 test E:\vue02-loader-example-master
> karma start build/karma.conf.js

ESC[32m14 01 2016 00:36:18.245:INFO [karma]: ESC[39mKarma v0.13.19 server started at http://localhost:9876/
ESC[32m14 01 2016 00:36:18.245:INFO [launcher]: ESC[39mStarting browser PhantomJS
ESC[32m14 01 2016 00:36:19.184:INFO [PhantomJS 1.9.8 (Windows 7 0.0.0)]: ESC[39mConnected on socket /#gbdHERoUXRk2i2tRAAAA with id
82364883

  A.vue
    ESC[32m□ ESC[39mshould have correct message
    ESC[32m□ ESC[39mshould render correct message
    ESC[32m□ ESC[39mshould render with mocked message

  App.vue
    ESC[32m□ ESC[39mshould render correct contents

PhantomJS 1.9.8 (Windows 7 0.0.0): Executed 4 of 4ESC[32m SUCCESSESC[39m (0 secs / 0.017 secs)
ESC[32mTOTAL: 4 SUCCESSESC[39m
```

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



build.js | webpack.base.config.js | webpack.dev.config.js | test.txt | webpack.pro

```
module.exports = {
  entry: './src/main.js',
  output: {
    path: './dist',
    publicPath: 'dist/',
    filename: 'build.js'
  },
  module: {
    loaders: [
      {
        test: /\.vue$/,
        loader: 'vue'
      },
      {
        test: /\.js$/,
        loader: 'babel!eslint',
        // make sure to exclude 3rd party code in node_modules
        exclude: /node_modules/
      },
      {
        // edit this for additional asset file types
        test: /\.(png|jpg|gif)$/,
        loader: 'url',
        query: {
          // inline files smaller then 10kb as base64 dataURL
          limit: 10000,
          // fallback to file-loader with this naming scheme
          name: '[name].[ext]?[hash]'
        }
      }
    ]
  },
  // lint all JavaScript inside *.vue files with ESLint
  // make sure to adjust your .eslintrc
  vue: {
    loaders: {
      js: 'babel!eslint'
    }
  }
}
```

webpack.config.js
我覺得要先看 loader, 才能了解怎麼將檔案交給那个模组

vue > vue 模组
js > babel, eslint 模组
png jpg gif > url 模组

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

工作 | 2015 | 2016 | 常用網站 | 变量作用域 - 廖雪...

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')
```

计算机 ▸ 2 (E:) ▸ vue02-loader-example-master

文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)

组织 | 打开 | 包含到库中 | 共享 | 新建文件夹

名称

build
dist
node_modules
src
test
.babelrc
.editorconfig
.eslintrc
.gitignore
circle.yml
index.html
package.json

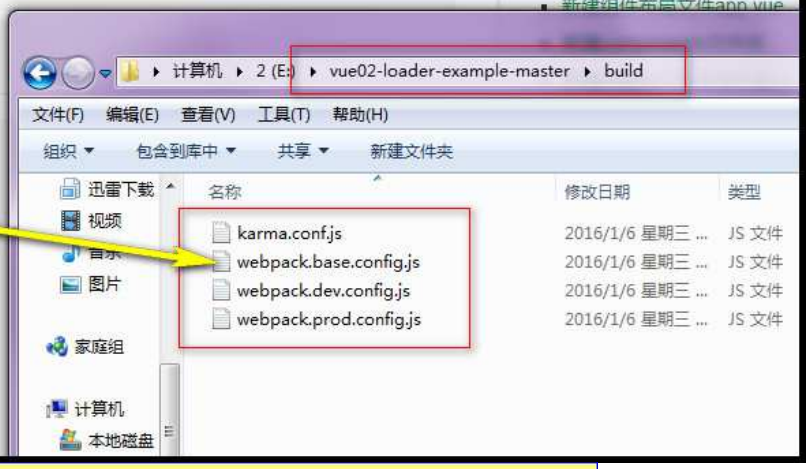
src 修改日期: 2016/1/14 星期四 上午 12:17
文件夹

/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
是這个



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
件之间可以尽可能地解耦，便于开发维护