

(框架) VUE.JS
(201) vue-cli

(實作) 160113

(實作) 160113

我是一個先做後想的人

邊做邊紀錄

分享給群友

只是想省下你10分鐘的時間

先看一下 **vue-cli** 是什麼東西。

沒講解的部份，當我不會，問我也不會，在群討論吧

vue-cli 中文說明: <http://segmentfault.com/a/1190000004219090>

最近有许多关于React项目工具的讨论，讨论了工具的使用问题。如果你现在正在使用Vue.js，当你构建一个原型的时候，你所需要做的通常就是通过 `<script>` 把Vue.js引入进就完事了。但是真实情况往往不是这样的。当我们真正开发一个应用的时候，我们不可避免的会用到一大堆的工具，模块化、预处理器、热模块加载、代码校验和测试。这些工个需要长期维护的大型应用是必须的，但是项目初始化将会是让人痛苦的事情。这就是为什么我们做了**vue-cli**，让一个简单的命令行工具来帮助你快速的构建一个拥有强大构建Vue.js项目。

仅仅为了初始化

下面是使用方法：

```
# 安装vue-cli
npm install -g vue-cli

# 使用vue-cli初始化项目
vue init webpack my-project

# 进入到目录
cd my-project

# 安装依赖
npm install

# 开始运行
npm run dev
```

上面的这些命令会从**vuejs-templates**拉取模板并安装，然后用NPM安装依赖，最后你只需要用个NPM脚本启动就能开始开发了。

官方模板

这些官方的模板存在的意义在于提供强大的项目构建能力，以至于用户可以尽可能快速的进行开发。然而能否真正的发挥作用还在于你如何组织你的代码和你使用的其他库。

所有的官方模板都可以在**vuejs-templates organization**找到。如果有一个新的模板放在了这里，你可以直接就用 `vue init <template-name> <project-name>` 使用。你也可以用 `list` 命令来看看现在有哪些官方模板是现在可用的。

目前有以下模板可供选择：

- **browserify** - 拥有高级功能的 Browserify + vueify 用于正式开发。
- **browserify-simple** - 拥有基础功能的 Browserify + vueify 用于快速原型开发。
- **webpack** - 拥有高级功能的 Webpack + vue-loader 用于正式开发。
- **webpack-simple** - 拥有基础功能的 Webpack + vue-loader 用于快速原型开发。

创建属于你的模板

如果你对官方的模板不感兴趣，你可以自己fork下来然后进行修改（或者重新写一个），然后用 **vue-cli** 来调用。因为 **vue-cli** 可以直接拉取 **git源**：

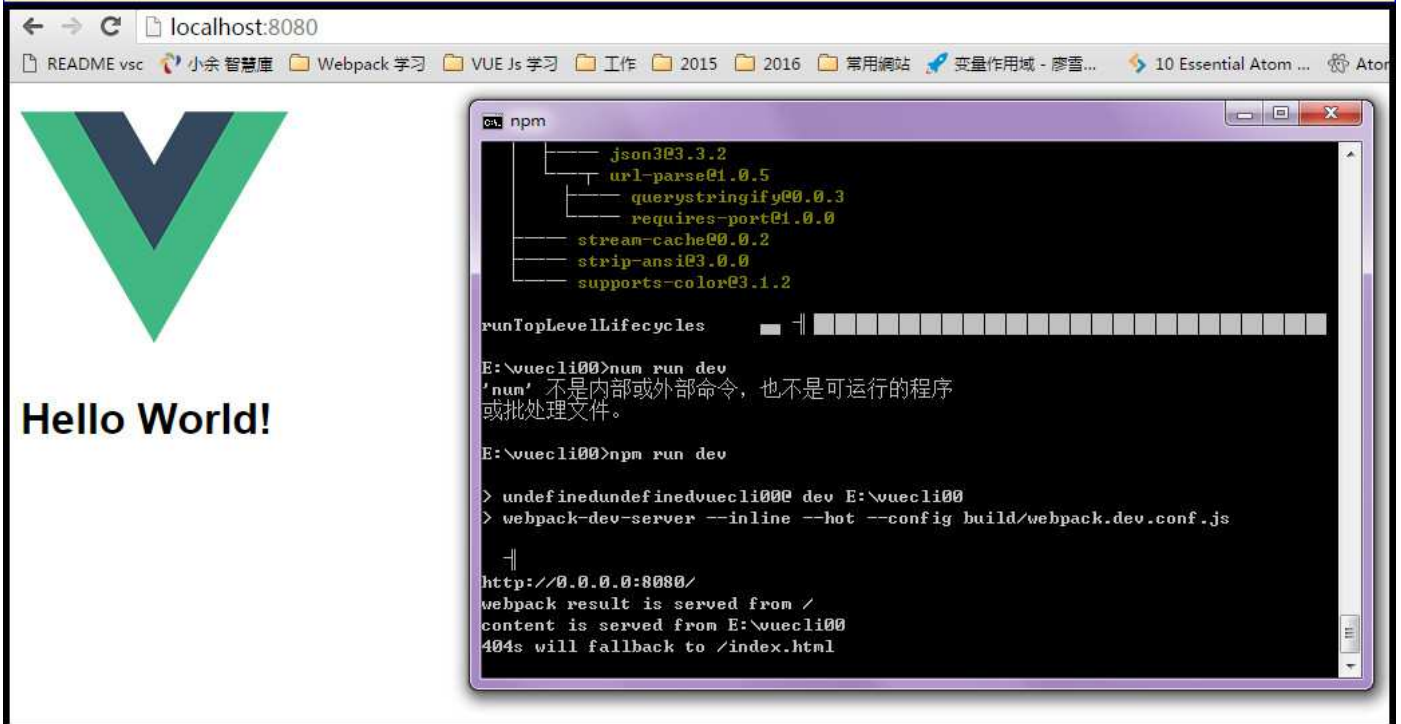
```
vue init username/repo my-project
```

无处不在的Vue组件

每一个模板都有自己的作用：基础版本用于快速构建原型，高级版本用于正式开发。这些模板有许多共同的功能，如都支持 ***.vue** 组件。这意味着任何第三方的Vue组件都可以用，并且可以轻而易举的发布在NPM上 - 让我们创建出更多的可重用组件！

vue-cli 官網: <https://www.npmjs.com/package/vue-cli>

160113
實驗結果



The screenshot shows a web browser at localhost:8080 displaying a large green 'V' logo and the text 'Hello World!'. An npm terminal window is open, showing the following commands and output:

```
runTopLevelLifecycles
E:\vuecli00>num run dev
'num' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
E:\vuecli00>npm run dev
> undefinedundefinedvuecli00@ dev E:\vuecli00
> webpack-dev-server --inline --hot --config build/webpack.dev.conf.js
+|
http://0.0.0.0:8080/
webpack result is served from /
content is served from E:\vuecli00
404s will fallback to /index.html
```

src下有兩個目錄
3個檔案



The screenshot shows a file explorer window for the directory 'E:\vuecli00\src'. It contains the following files and folders:

名称	修改日期	类型	大小
assets	2016/1/13 星期...	文件夹	
components	2016/1/13 星期...	文件夹	
App.vue	2016/1/13 星期...	VUE 文件	1 KB
index.html	2016/1/13 星期...	360 se HTML Do...	1 KB
main.js	2016/1/13 星期...	JS 文件	1 KB

main.js

1. 這是主要控制區
2. 使用 vue，並且 載入 app.vue
3. el body
4. components 組件，app



```
1 import Vue from 'vue'
2 import App from './App'
3
4 /* eslint-disable no-new */
5 new Vue({
6   el: 'body',
7   components: { App }
8 })
9
```

index.html
這里對初學者來說，太難以理解了

```
index.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>undefinedundefinedvuecli00</title>{% for (var css in o.htmlWebpackPlugin.files.css) { %}
6     <link href="{%=o.htmlWebpackPlugin.files.css[css] %}" rel="stylesheet">{% } %}
7   </head>
8   <body>
9     <app></app>{% for (var chunk in o.htmlWebpackPlugin.files.chunks) { %}
10    <script src="{%=o.htmlWebpackPlugin.files.chunks[chunk].entry %}"></script>{% } %}
11  </body>
12 </html>
13
```

App.vue
1.<template> 這個好理解
2.<script> ES6的語法
重點在 export, import 的使用

```
index.html x App.vue x
1 <template>
2   <div id="app">
3     
4     <hello></hello>
5   </div>
6 </template>
7
8 <script>
9 import Hello from '../components/Hello'
10
11 export default {
12   components: {
13     Hello
14   }
15 }
16 </script>
17
18 <style>
19 body {
20   font-family: Helvetica, sans-serif;
21 }
22 </style>
23
```

src/assets/logo.png
要注意，在 app.vue 怎麼寫



Hello.vue
特別之處，在這個組件，有 data

```
index.html App.vue Hello.vue
1 <template>
2   <div class="hello">
3     <h1>{{ msg }}</h1>
4   </div>
5 </template>
6
7 <script>
8 export default {
9   data () {
10    return {
11      // note: changing this line won't causes changes
12      // with hot-reload because the reloaded component
13      // preserves its current state and we are modifying
14      // its initial state.
15      msg: 'Hello World!'
16    }
17  }
18 }
19 </script>
20
```

components/Hello.vue

计算机 > 2 (E:) > vuecli00 > src > components

编辑(E) 查看(V) 工具(T) 帮助(H)

包含到库中 共享 新建文件夹

名称	修改日期	类型	大小
Hello.vue	2016/1/13 星期...	VUE 文件	1 KB

npm run build
出現 dist 目錄

名称	修改日期	类型
build	2016/1/13 星期...	文件夹
dist	2016/1/13 星期...	文件夹
node_modules	2016/1/13 星期...	文件夹
src	2016/1/13 星期...	文件夹
test	2016/1/13 星期...	文件夹
.babelrc	2016/1/13 星期...	BABELRC
.eslintrc	2016/1/13 星期...	ESLINTRC
.gitignore	2016/1/13 星期...	文件
package.json	2016/1/13 星期...	JSON 文件
README.md	2016/1/13 星期...	MD 文件

綠色是產出的 file

```
C:\windows\system32\cmd.exe
404s will fallback to /index.html
^C终止批处理操作吗(Y/N)? y
E:\vuecli00>npm run build

> undefinedundefinedvuecli00@ build E:\vuecli00
> rimraf dist && webpack --progress --hide-modules --config build/webpack.pro
onf.js

+|
Hash: 6f5cd831b8c36bc4e004
Version: webpack 1.12.11
Time: 4853ms

      Asset      Size  Chunks             C
k Names
  app.76bfbfec8de9fcd09357.js    89 kB      0  [emitted]  a
  app.6b590370d011a33cd6d0437256238024.css 106 bytes      0  [emitted]  a
  app.76bfbfec8de9fcd09357.js.map    686 kB      0  [emitted]  a
  app.6b590370d011a33cd6d0437256238024.css.map 572 bytes      0  [emitted]  a
    ../index.html    306 bytes      0  [emitted]

Child extract-text-webpack-plugin:

+|
E:\vuecli00>
```

dist 目錄
有 static 一個目錄
有 index.html 一個檔案

计算机 > 2 (E:) > vuecli00 > dist

编辑(E) 查看(V) 工具(T) 帮助(H)

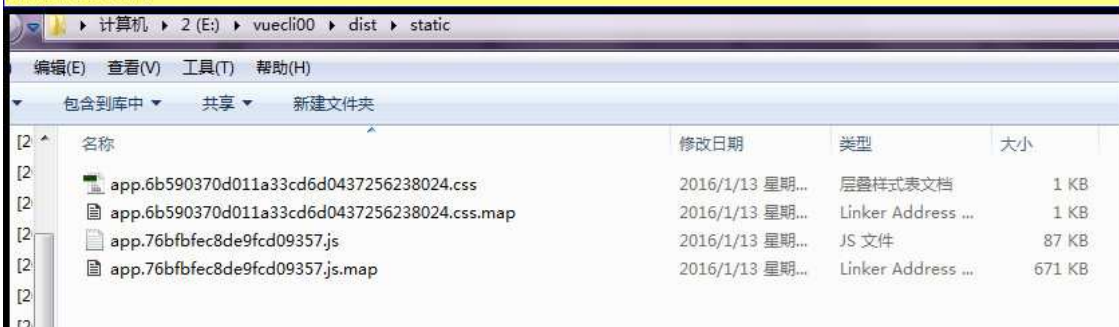
打开 包含到库中 共享 电子邮件 新建文件夹

名称	修改日期	类型	大小
static	2016/1/13 星期...	文件夹	
index.html	2016/1/13 星期...	360 se HTML Do...	1 KB

dist 目錄
index.html
重點在 被webpack, 編譯後的 檔名

```
index.html — src × App.vue × Hello.vue × index.html — dist ×
1 |<!DOCTYPE html>
2 |<html>
3 |   <head>
4 |     <meta charset="utf-8">
5 |     <title>undefinedundefinedvuecli00</title>
6 |     <link href="/static/app.6b590370d011a33cd6d0437256238024.css" rel="stylesheet">
7 |   </head>
8 |   <body>
9 |     <app></app>
10 |     <script src="/static/app.76bfbfec8de9fcd09357.js"></script>
11 |   </body>
12 | </html>
13 |
```

dist 目錄下
static 目錄
被編譯後的檔案



run build
依那個文件執行 工程化呢?
我打開 package.json

```
package.json ×
<No Schema Selected>
{
  "name": "undefinedundefinedvuecli00",
  "description": "A Vue.js project",
  "author": "vuecli00",
  "private": true,
  "scripts": {
    "dev": "webpack-dev-server --inline --hot --config build/webpack.dev.conf.js",
    "build": "rimraf dist && webpack --progress --hide-modules --config build/webpack.prod.conf.js",
    "test": "karma start build/karma.conf.js --single-run"
  },
  "dependencies": {
    "vue": "^1.0.0"
  }
}
```

build 目錄
webpack.prod.conf.js
跟 npm run build 有關係



webpack.prod.conf.js
—共 58行。

webpack.prod.conf.js x

```
1 var webpack = require('webpack')
2 var config = require('./webpack.base.conf')
3 var ExtractTextPlugin = require('extract-text-webpack-plugin')
4 var HtmlWebpackPlugin = require('html-webpack-plugin')
5
6 // naming output files with hashes for better caching.
7 // dist/index.html will be auto-generated with correct URLs.
8 config.output.filename = '[name].[chunkhash].js'
9 config.output.chunkFilename = '[id].[chunkhash].js'
10
11 // whether to generate source map for production files.
12 // disabling this can speed up the build.
13 var SOURCE_MAP = true
14
15 config.devtool = SOURCE_MAP ? 'source-map' : false
16
17 // generate loader string to be used with extract text plugin
18 function generateExtractLoaders (loaders) {
19   return loaders.map(function (loader) {
20     return loader + '-loader' + (SOURCE_MAP ? '?sourceMap' : '')
21   }).join('!')
22 }
23
24 config.vue.loaders = {
25   js: 'babel!eslint',
26   // http://vuejs.github.io/vue-loader/configurations/extract-css.html
27   css: ExtractTextPlugin.extract('vue-style-loader', generateExtractLoaders(['css'])),
28   less: ExtractTextPlugin.extract('vue-style-loader', generateExtractLoaders(['css', 'less'])),
29   sass: ExtractTextPlugin.extract('vue-style-loader', generateExtractLoaders(['css', 'sass'])),
30   stylus: ExtractTextPlugin.extract('vue-style-loader', generateExtractLoaders(['css', 'stylus']))
31 }
32
33 config.plugins = (config.plugins || []).concat([
34   // http://vuejs.github.io/vue-loader/workflow/production.html
35   new webpack.DefinePlugin({
36     'process.env': {
37       NODE_ENV: '"production"'
38     }
39   }),
40   new webpack.optimize.UglifyJsPlugin({
41     compress: {
42       warnings: false
43     }
44   }),
45   new webpack.optimize.OccurrenceOrderPlugin(),
46   // extract css into its own file
47   new ExtractTextPlugin('[name].[contenthash].css'),
48   // generate dist index.html with correct asset hash for caching.
49   // you can customize output by editing /build/index.template.html
50   // see https://github.com/ampedandwired/html-webpack-plugin
51   new HtmlWebpackPlugin({
52     filename: './index.html',
53     template: 'src/index.html'
54   })
55 ])
56
57 module.exports = config
58
```