

需求：

1. 能够使用npm安装，支持amd规范
2. 能够按需引入
3. 能够按需打包
4. 能够编写单元测试用例

步骤：

1. 搭建一个属于民生内部的npm仓库，自己管理包的同时借助npm的命令行工具快速复用业务组件。
2. 创建组件库工程
3. 发布组件到npm
4. 测验组件库效果
5. 调试组件库

遇到的问题

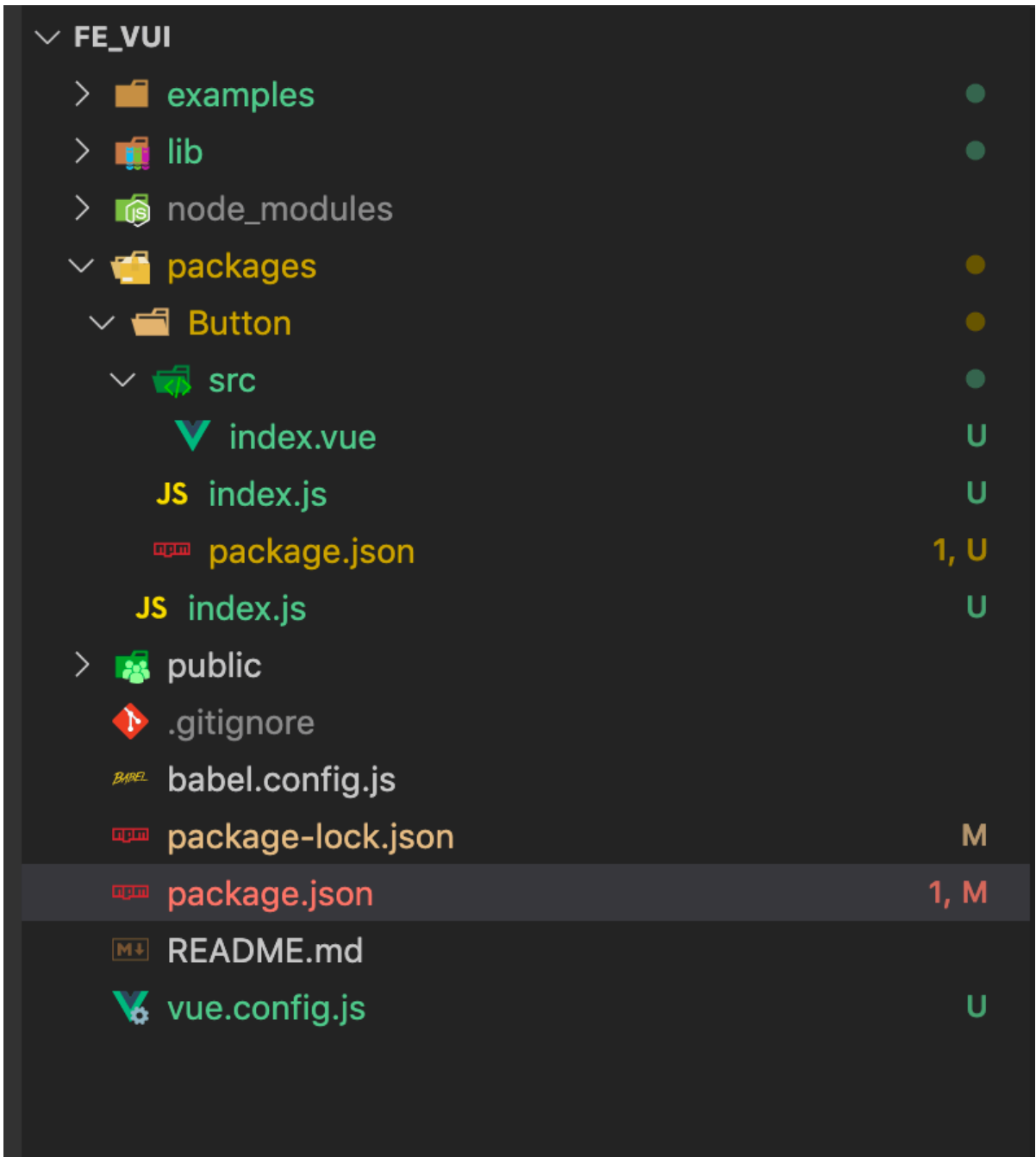
1. 如何在最少依赖下快速开发一个vue组件
2. 如何管理所有包的依赖，减少包的体积
3. 如何快速创建组件实例

demo实例---全局导入

1. 项目初始化

```
vue init fe_vui
```

更改一下项目目录，更改的目录如下



将生成的src改成examples，用于测试自定义的组件

新建packages文件夹，用于存放自定义组件，例如Button，Button>src>index.vue组件相关代码，Button>index.js 组件的入口文件，即该文件导出该组件（可以单个导出该组件），其内容如下：packages>index.js所有组件的入口文件

2. 编写第一个组件

Button>src>index.vue

```
JS index.js .../Button ● JS index.js packages index.vue ●
packages > Button > src > index.vue > {} "index.vue"
1 <template>
2   <div class="x-button">
3     <slot></slot>
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: 'x-button',
10    props: {
11      type: String
12    }
13  }
14 </script>
15 <style>
16   .x-button {
17     display: inline-block;
18     padding: 3px 6px;
19     background: #000;
20     color: #fff;
21   }
22 </style>
```

Button>index.js

```
JS index.js .../Button • JS index.js packages V index.vue npm
packages > Button > JS index.js > ...
1  // 导入组件, 组件必须声明 name
2  import XButton from './src'
3  // 为组件提供 install 安装方法, 供按需引入
4  XButton.install = function (Vue) {
5    Vue.component(XButton.name, XButton)
6  }
7
8  // 导出组件
9  export default XButton
10
11
```

packages>index.js

```
JS index.js .../Button • JS index.js packages X V index.vue package.json .../Button JS main.js V App.vue
packages > JS index.js > [?] default
1  // 导入button组件
2  import XButton from './Button'
3  // 组件列表
4  const components = [
5    XButton
6  ]
7
8  // 定义 install 方法, 接收 Vue 作为参数。如果使用 use 注册插件, 那么所有的组件都会被注册
9  const install = function (Vue) {
10    // 判断是否安装
11    if (install.installed) return
12    // 遍历注册全局组件
13    components.map(component => Vue.component(component.name, component))
14  }
15
16  // 判断是否是直接引入文件
17  if (typeof window !== 'undefined' && window.Vue) {
18    install(window.Vue)
19  }
20
21  export default {
22    // 导出的对象必须具有 install, 才能被 Vue.use() 方法安装
23    install,
24    // 以下是具体的组件列表
25    XButton
26  }
```

3. 测试组件

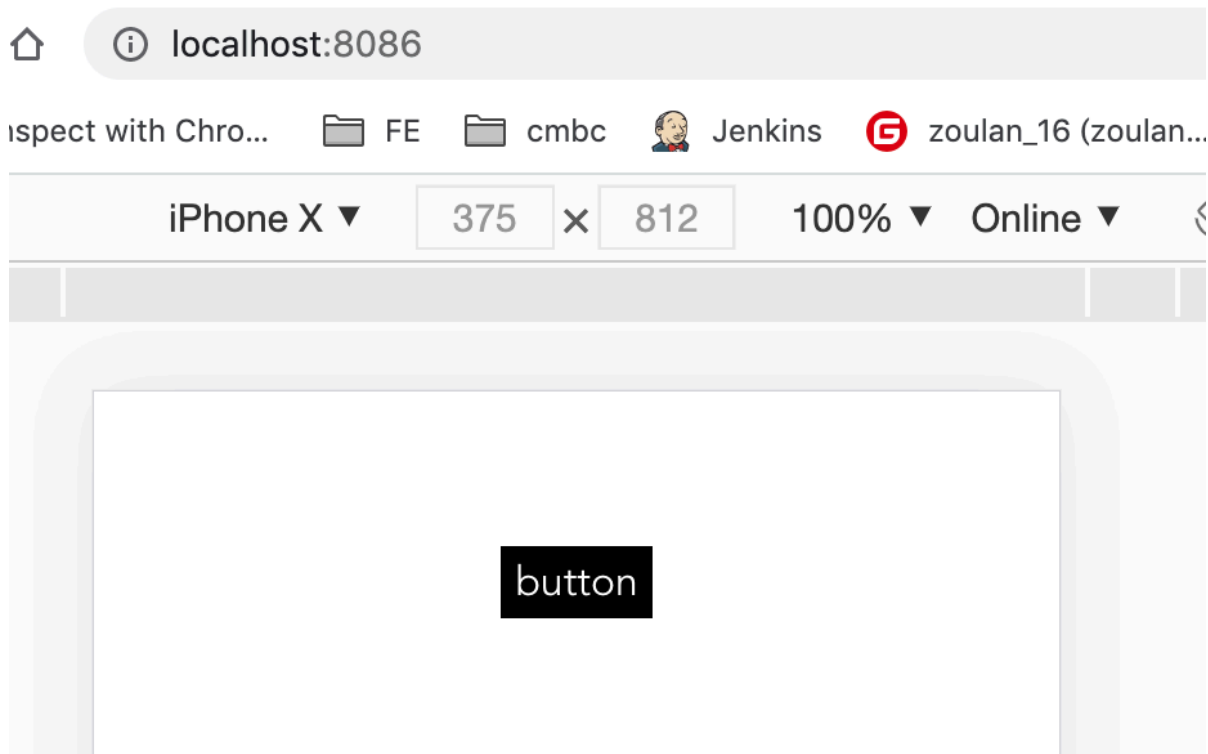
在fe_vui的入口文件main.js中引入

```
JS index.js .../Button • JS index.js packages ✓ index.vue • package.json .../Button JS main.js ×
examples > JS main.js > ...
1  import Vue from 'vue'
2  import App from './App.vue'
3
4  // 导入组件库
5  import fe_vui from '../packages'
6
7  // 注册组件库
8  Vue.use(fe_vui)
9
10 Vue.config.productionTip = false
11
12 new Vue({
13   render: h => h(App),
14 }).$mount('#app')
```

引入后在app.vue中引入

```
JS index.js .../Button • JS index.js packages ✓ index.vue • package.json .../Button JS main.js ✓ App.vue × packa
examples > ✓ App.vue > {} "App.vue" > script > default
1 <template>
2   <div id="app">
3     <x-button type="primary">button</x-button>
4   </div>
5 </template>
6
7 <script>
8
9 export default {
10   name: 'App',
```

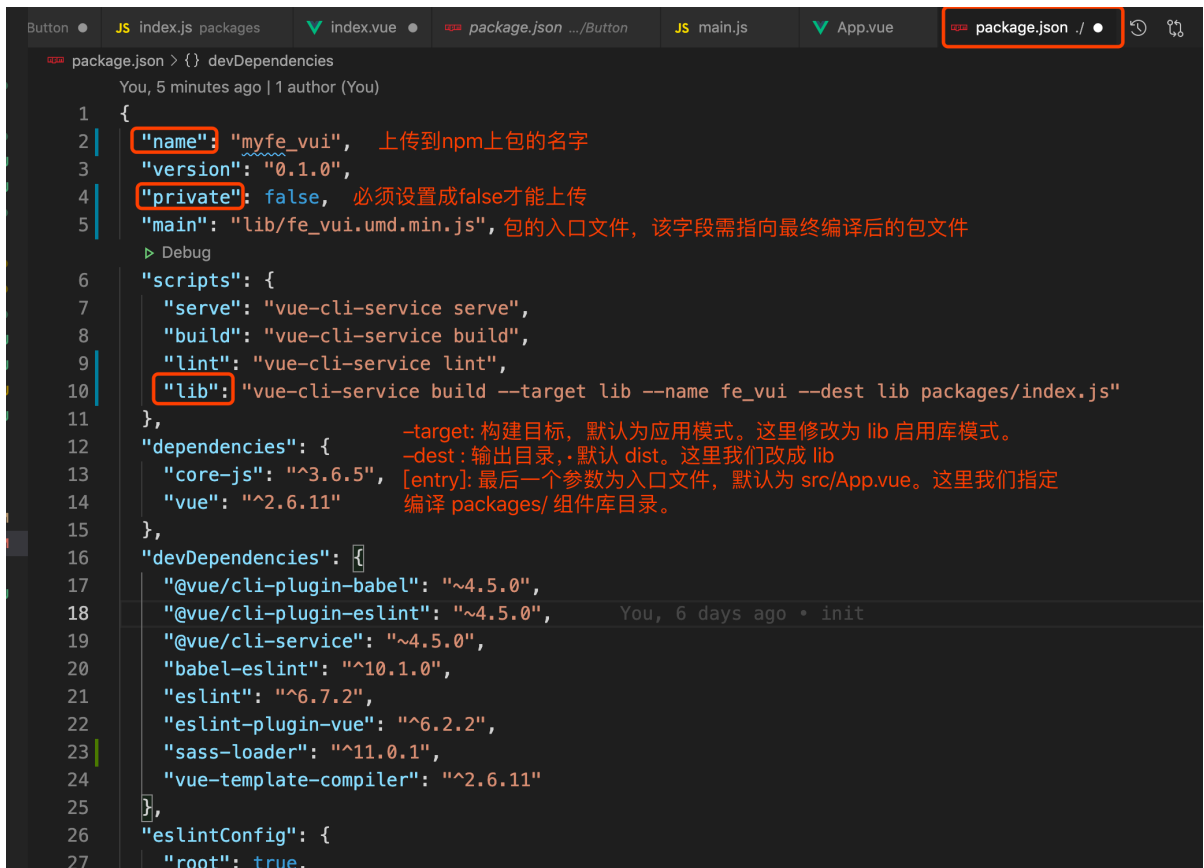
测试结果如下：



在本地测试组件没有问题之后就可以打包上传到npm。

4. 发布第一个npm包

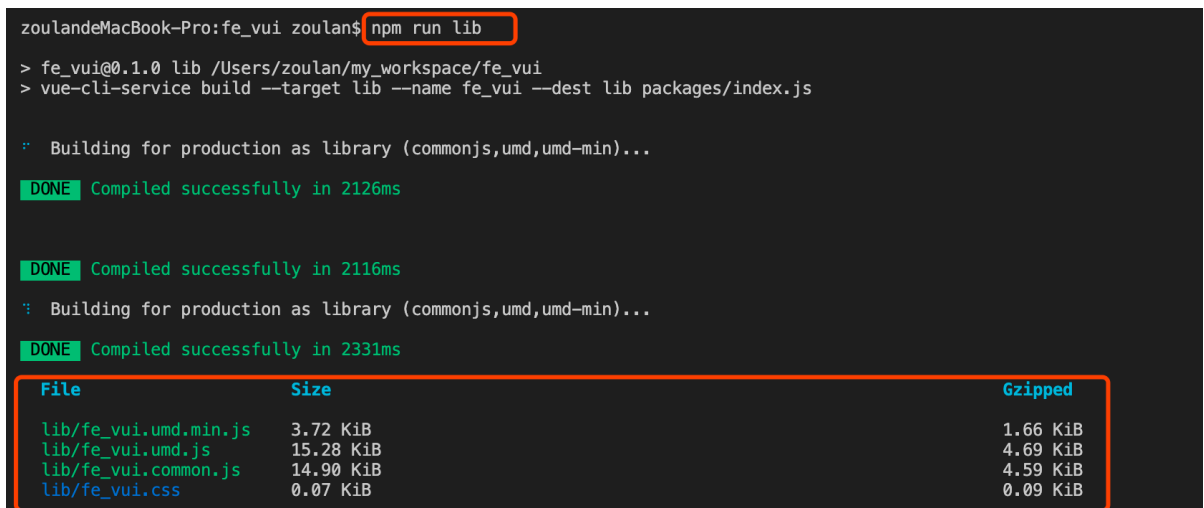
在上传之前先配置好fe_vui工程的fe_vui>package.json文件



```
package.json > {} devDependencies
You, 5 minutes ago | 1 author (You)
1 {
2   "name": "myfe_vui", 上传到npm上包的名字
3   "version": "0.1.0",
4   "private": false, 必须设置成false才能上传
5   "main": "lib/fe_vui.umd.min.js", 包的入口文件, 该字段需指向最终编译后的包文件
6   "scripts": {
7     "serve": "vue-cli-service serve",
8     "build": "vue-cli-service build",
9     "lint": "vue-cli-service lint",
10    "lib": "vue-cli-service build --target lib --name fe_vui --dest lib packages/index.js"
11  },
12  "dependencies": {
13    "core-js": "^3.6.5",
14    "vue": "^2.6.11"
15  },
16  "devDependencies": {
17    "@vue/cli-plugin-babel": "~4.5.0",
18    "@vue/cli-plugin-eslint": "~4.5.0",
19    "@vue/cli-service": "~4.5.0",
20    "babel-eslint": "^10.1.0",
21    "eslint": "^6.7.2",
22    "eslint-plugin-vue": "^6.2.2",
23    "sass-loader": "^11.0.1",
24    "vue-template-compiler": "^2.6.11"
25  },
26  "eslintConfig": {
27    "root": true,
```

将package.json配置完成后, 执行编译库名=命令

npm run lib



```
zoulandeMacBook-Pro:fe_vui zoulan$ npm run lib
> fe_vui@0.1.0 lib /Users/zoulan/my_workspace/fe_vui
> vue-cli-service build --target lib --name fe_vui --dest lib packages/index.js

Building for production as library (commonjs,umd,umd-min)...
DONE Compiled successfully in 2126ms

DONE Compiled successfully in 2116ms
Building for production as library (commonjs,umd,umd-min)...
DONE Compiled successfully in 2331ms
```

File	Size	Gzipped
lib/fe_vui.umd.min.js	3.72 KiB	1.66 KiB
lib/fe_vui.umd.js	15.28 KiB	4.69 KiB
lib/fe_vui.common.js	14.90 KiB	4.59 KiB
lib/fe_vui.css	0.07 KiB	0.09 KiB

编译完成之后就会看到lib文件下有如下内容:



编译完成之后，接下来就可以开始上传了

首先查看一下自己npm源

npm get registry 如果是淘宝的镜像源，就需要修改一下源

npm config set registry <http://registry.npmjs.org>

更改之后，先登录

```
zoulandeMacBook-Pro:fe_vui zoulan$ npm login
Username: zoulan
Password:
Email: (this IS public) 417747041@qq.com
Logged in as zoulan on http://registry.npmjs.org/.
```

发布

```

zoulandeMacBook-Pro:fe_vui zoulan$ npm publish --access=public
npm notice
npm notice 📦 fec_vui@0.1.0
npm notice === Tarball Contents ===
npm notice 975B package.json
npm notice 73B babel.config.js
npm notice 318B README.md
npm notice 451B vue.config.js
npm notice 392B examples/App.vue
npm notice 6.8kB examples/assets/logo.png
npm notice 2.0kB examples/components/HelloWorld.vue
npm notice 227B examples/main.js
npm notice 176B lib/demo.html
npm notice 15.3kB lib/fe_vui.common.js
npm notice 15.7kB lib/fe_vui.common.js.map
npm notice 74B lib/fe_vui.css
npm notice 15.6kB lib/fe_vui.umd.js
npm notice 16.3kB lib/fe_vui.umd.js.map
npm notice 3.8kB lib/fe_vui.umd.min.js
npm notice 19.3kB lib/fe_vui.umd.min.js.map
npm notice 243B packages/Button/index.js
npm notice 220B packages/Button/package.json
npm notice 302B packages/Button/src/index.vue
npm notice 683B packages/index.js
npm notice 4.3kB public/favicon.ico
npm notice 611B public/index.html
npm notice === Tarball Details ===
npm notice name: fec_vui
npm notice version: 0.1.0
npm notice package size: 27.0 kB
npm notice unpacked size: 103.9 kB
npm notice shasum: 8a304ebf30b43499c83f77f1820986775fa112d2
npm notice integrity: sha512-s4oBFZrac9zkg[...]Tt5DWp60RX0mw==
npm notice total files: 22
npm notice
+ fec_vui@0.1.0

```

发布成功之后就能在npm中看到自己上传的包了。

2 packages

fec_vui

Project setup `` npm install ``

zoulan published 0.1.0 • 4 hours ago

myfe_vui

Project setup `` npm install ``

zoulan published 0.1.0 • 3 hours ago

至此，包已成功上传至npm，接下来就是在项目中引用自己定义的组件

5. 使用自定组件

首先将包下下来

```
npm i myfe_vui --save
```

成功之后在依赖包node_modules目录下边就有自定义的组件包

下下来之后就是使用了

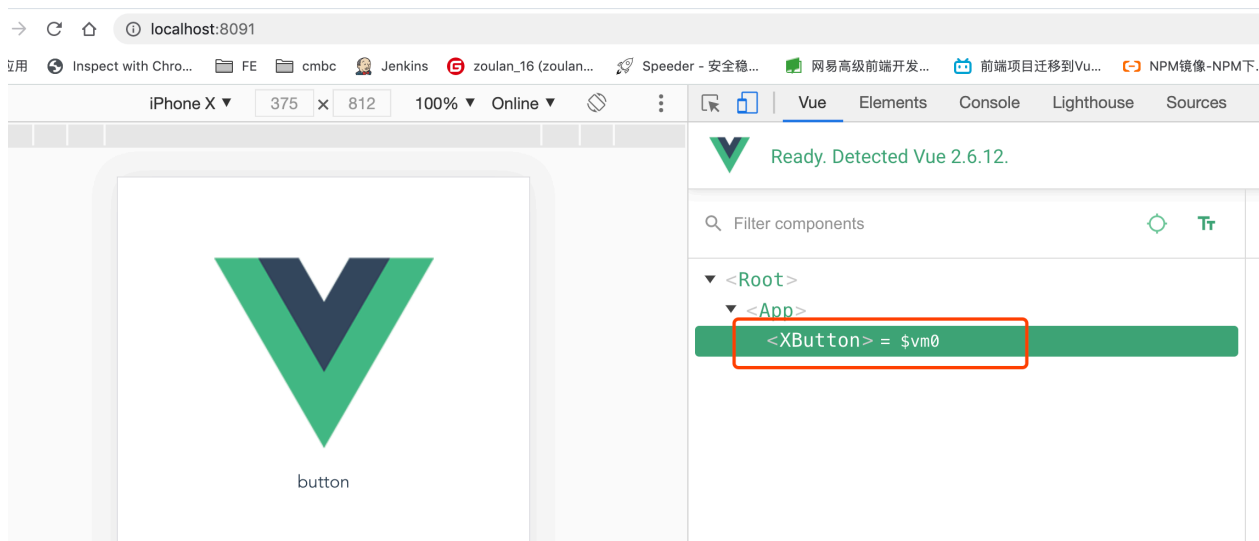

```
EXPLORER
TEST_UI_DEMO
  node_modules
  src
  testui
    node_modules
    public
    src
      assets
      components
        HelloWorld.vue
        App.vue
        JS main.js
      .gitignore
      babel.config.js
      package-lock.json
      package.json
      README.md
      package-lock.json
      package.json

main.js
1 import Vue from 'vue'
2 import App from './App.vue'
3 import myfe_vui from 'myfe_vui'
4 Vue.use(myfe_vui)
5 Vue.config.productionTip = false
6
7 new Vue({
8   render: h => h(App),
9 }).$mount('#app')
10
```

```
EXPLORER
TEST_UI_DEMO
  node_modules
  src
  testui
    node_modules
    public
    src
      assets
      components
        HelloWorld.vue
        App.vue
        JS main.js
      .gitignore
      babel.config.js
      package-lock.json
      package.json
      README.md
      package-lock.json
      package.json

App.vue
1 <template>
2   <div id="app">
3     
4     <x-button type="primary">button</x-button>
5     <!-- <HelloWorld msg="Welcome to Your Vue.js App"/> -->
6   </div>
7 </template>
8
9 <script>
10 // import HelloWorld from './components/HelloWorld.vue'
11
12 export default {
13   name: 'App',
14   components: {
15     // HelloWorld
16   }
17 }
18 </script>
19
```

结果如下：



至此，全局引入组件的实现已完成，接下来研究每个组件的可配型。

