## 本地测试工程化打包

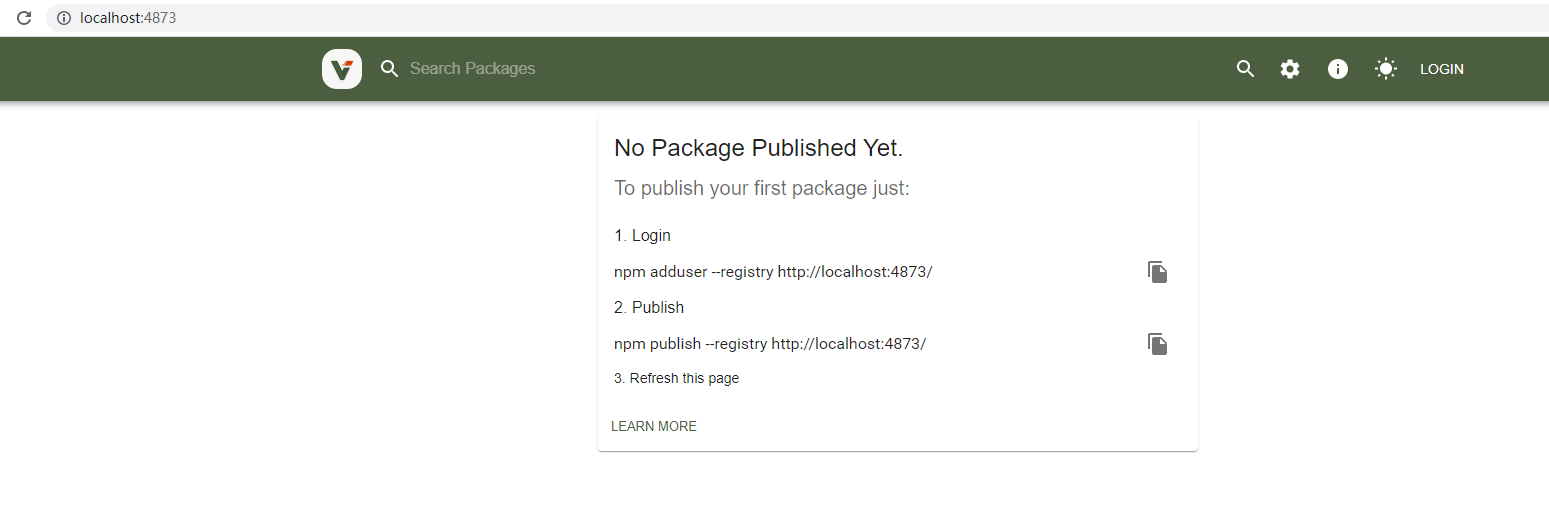
###### 一：安装verdaccio

1：全局安装verdaccio：npm i verdaccio -g

2：verdaccio （cmd运行）



在浏览器中输入：<http://localhost:4873/> 会出现以下界面



至此，第一步，准备工程化打包后，发布的本地测试环境已配置完毕

###### 二：基于tcore-smart已有的代码改造，并进行工程化的打包

1：将项目原来的src目录改为examples目录，用于存放原项目代码（也可以不改）

2：增加package目录，统一管理需要导出的组件代码文件

3：改造后的项目目录如下图：图2，图3，图4

4：由于项目启动默认入口是 src/main.js，第一步已将 src 文件名改成 examples 所以需要配置入口

在vue.config.js 文件下需要重新配置入口相关的参数：如图1

图1



图2

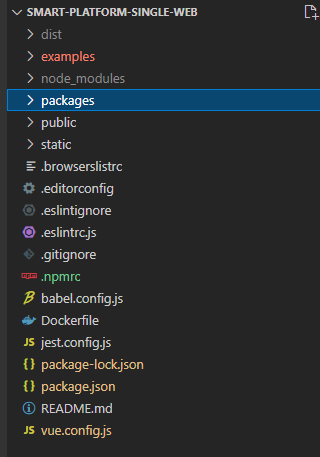


图3

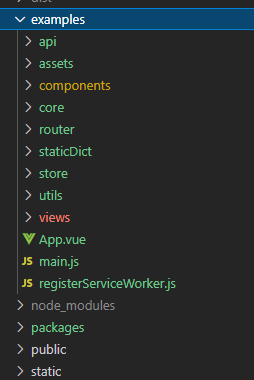
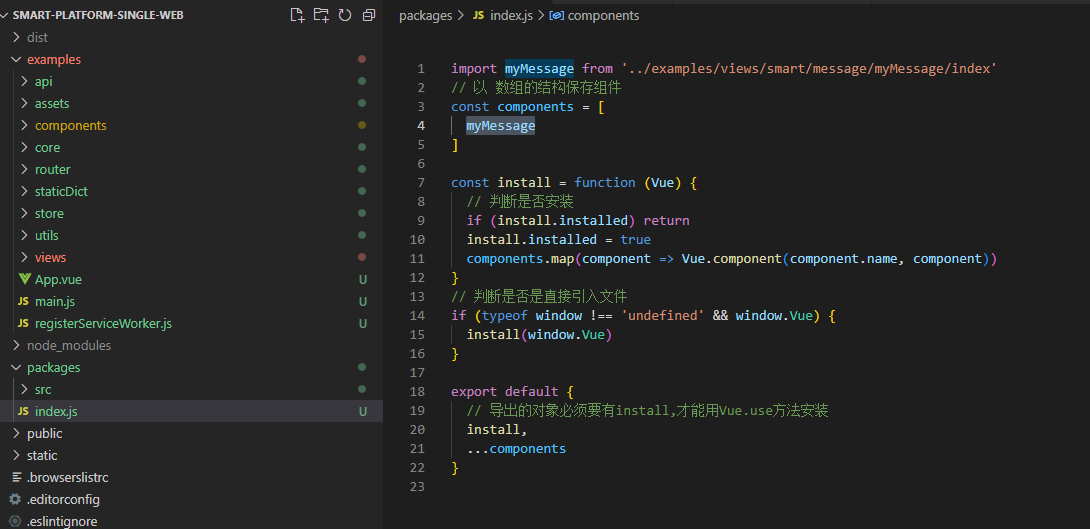


图4



###### 三：导出组件

之前已经创建了一个 packages 目录，用来存放组件  
该目录下存放一个 index.js 整合所有组件，并对外导出，便于外部引用

这里以“我的消息”为例：

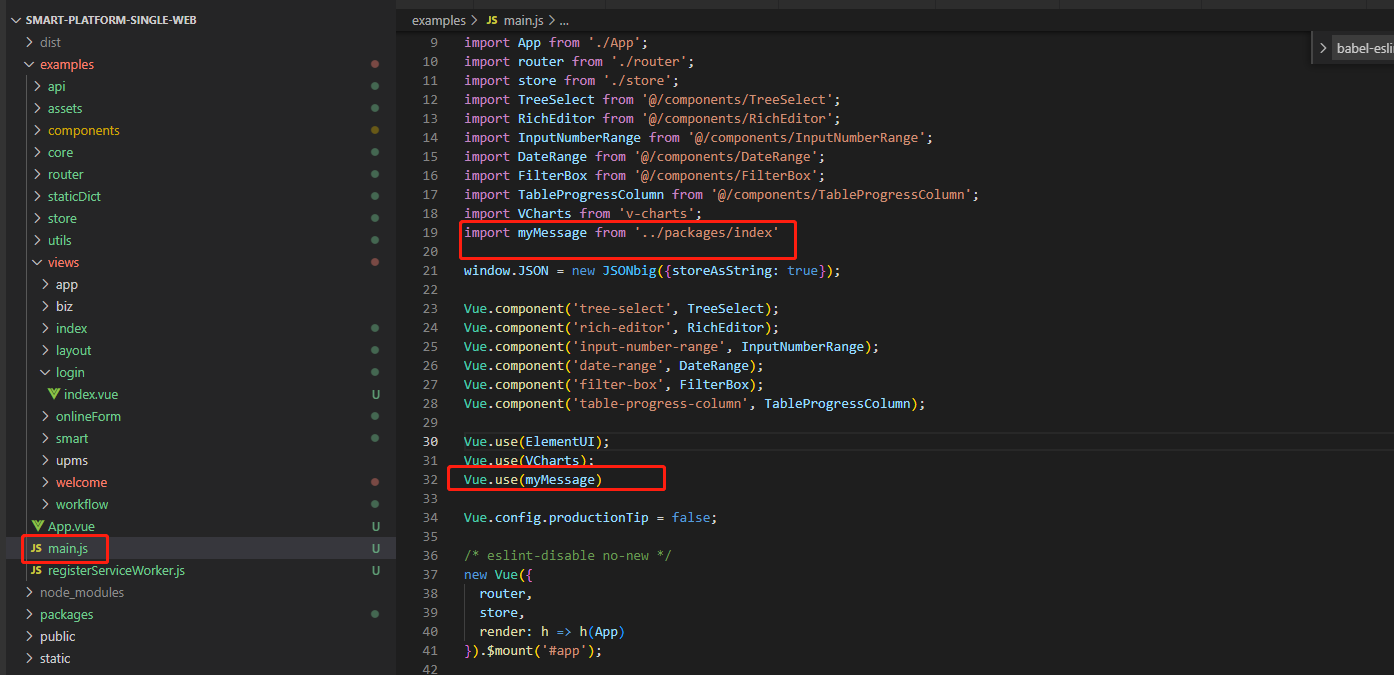
examples/views/smart/message/myMessage/index是组件的代码

注意点: 一定要声明name，name就是组件的标签

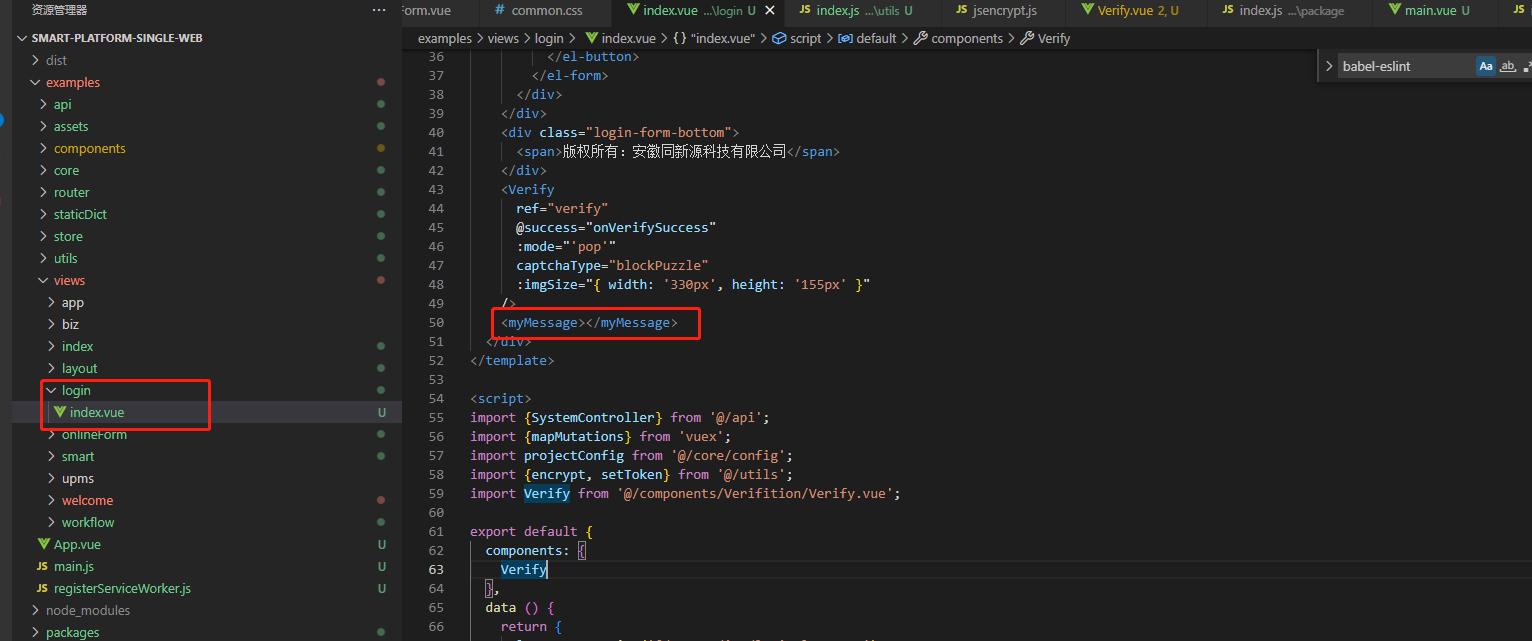
packages/index.js 整合并导出组件，实现组件全局注册：如图4

###### 四、测试组件是否导出成功

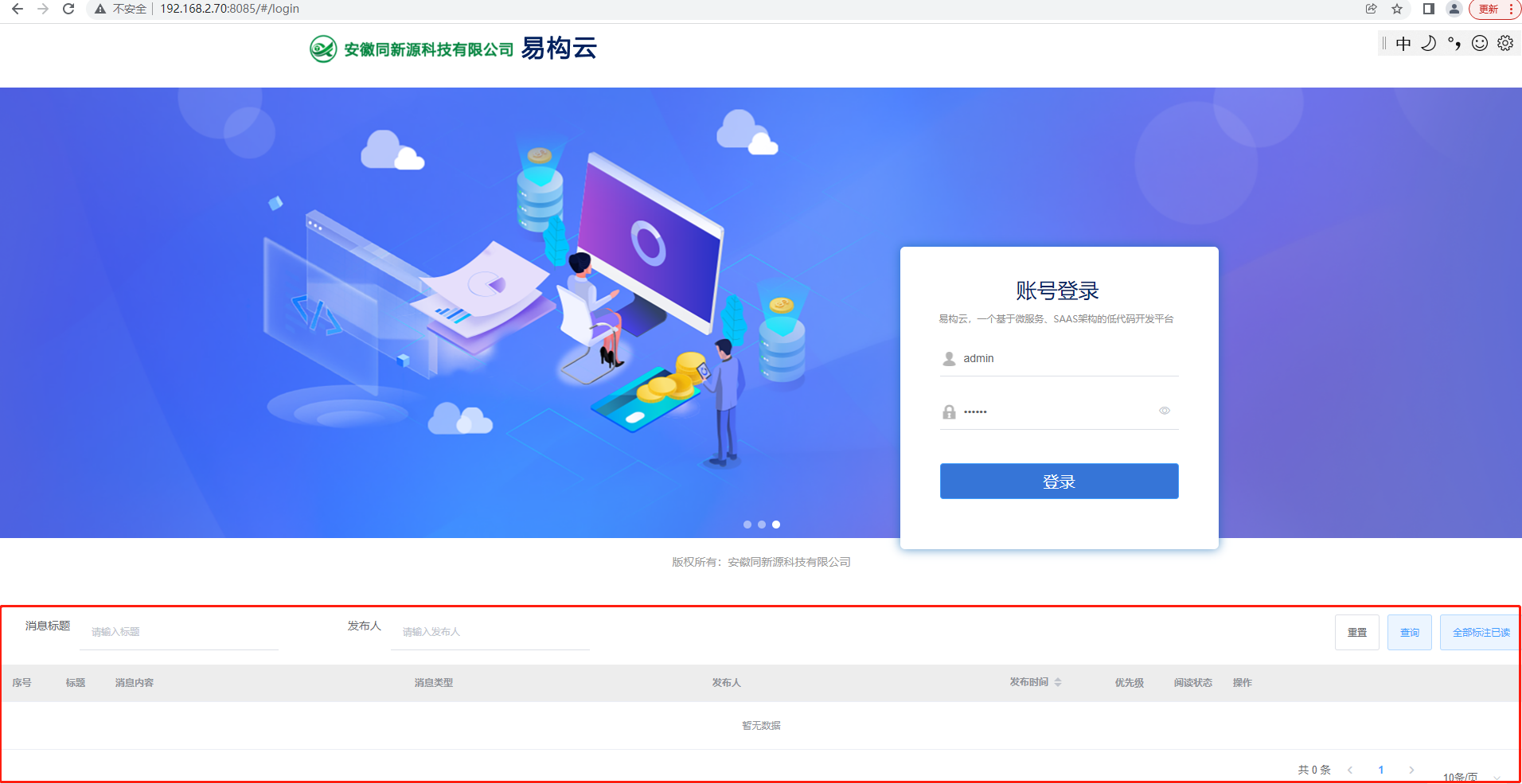
在examples/main.js中引入组件



在登录页使用组件



页面效果如下，（由于未登录进页面，只在登录页测试了一下，测试发现，“我的消息“页面已经被引入，但是接口调用返回code：401 token验证失败）

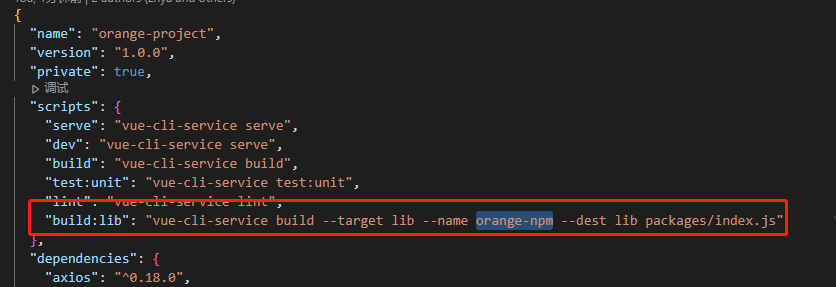


###### 五：打包组件

在 package.json 里的 scripts 添加一个 lib 命令

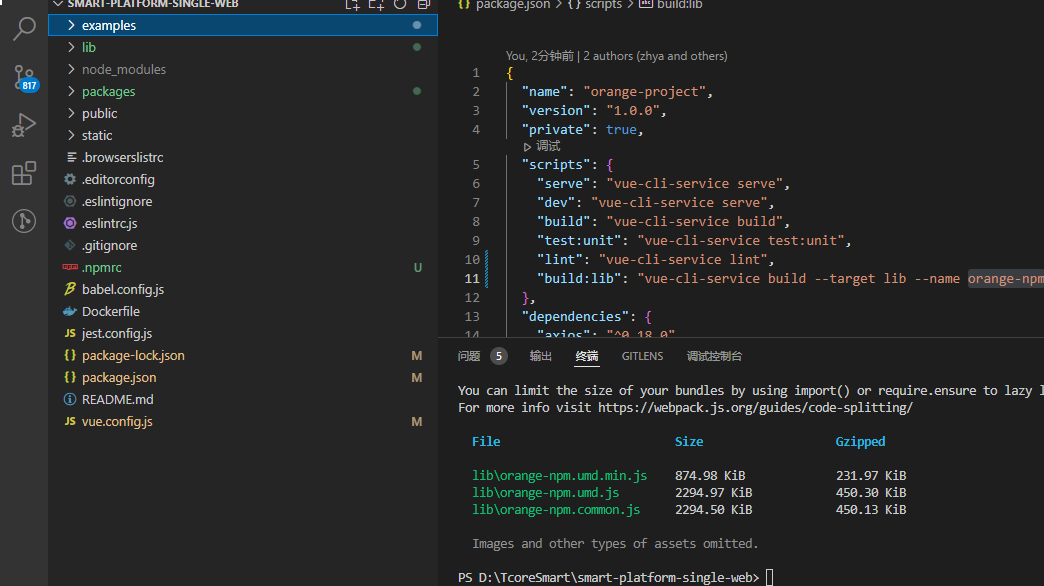
"build:lib": "vue-cli-service build --target lib --name 生成的库名 --dest lib packages/index.js"

这里我设置生成的库名为：orange-npm

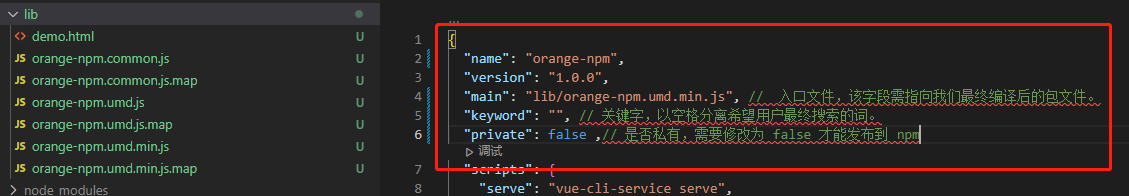


运行 npm run build:lib，编译组件，成功后会在根目录生成一个lib文件

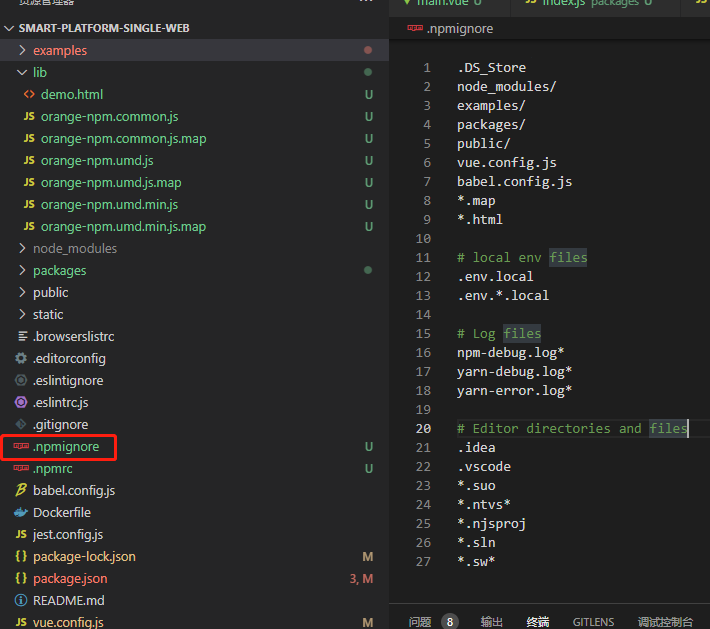




配置package.json文件中发布到npm的字段



在根目录下新建.npmignore文件，设置忽略发布文件

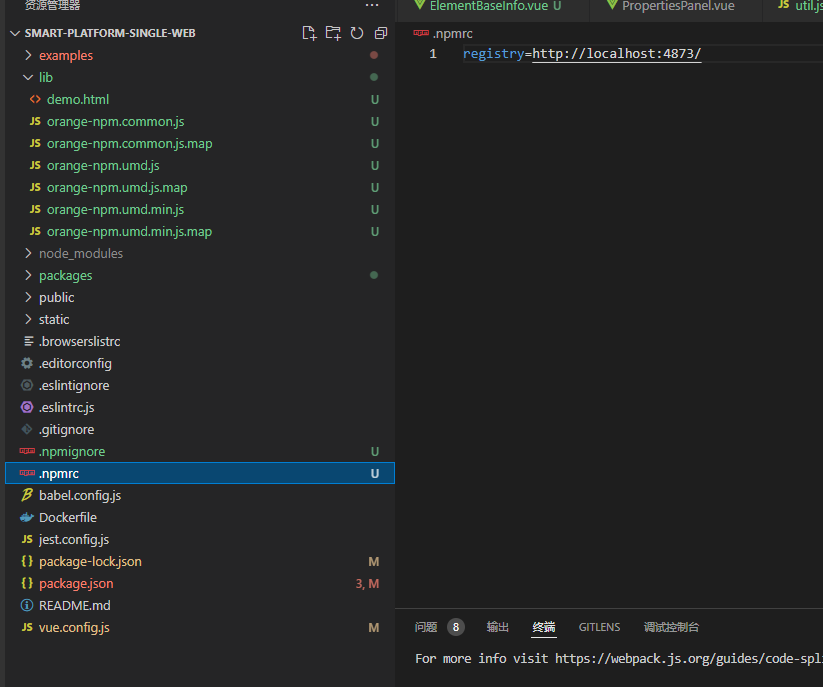


###### 六：发布npm

1：配置npm配置文件.npmrc

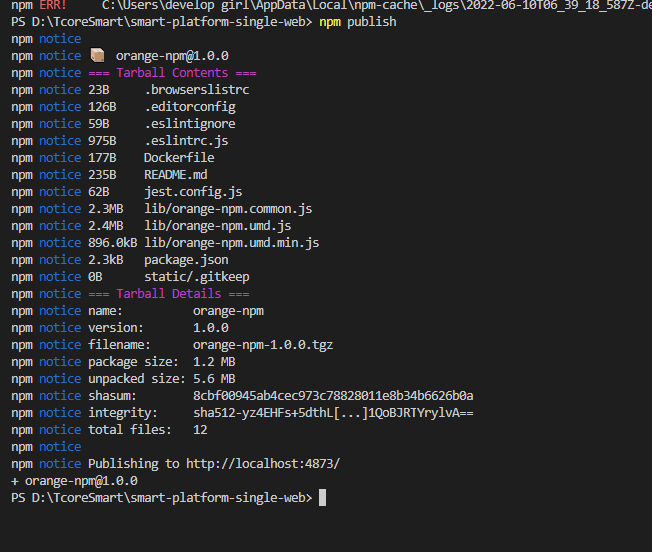
.npmrc 可以设置 package.json 中依赖包的安装来源，以及发布地址

例如：如下图所示，此时配置好的是第一步已经配置好的本地测试私服

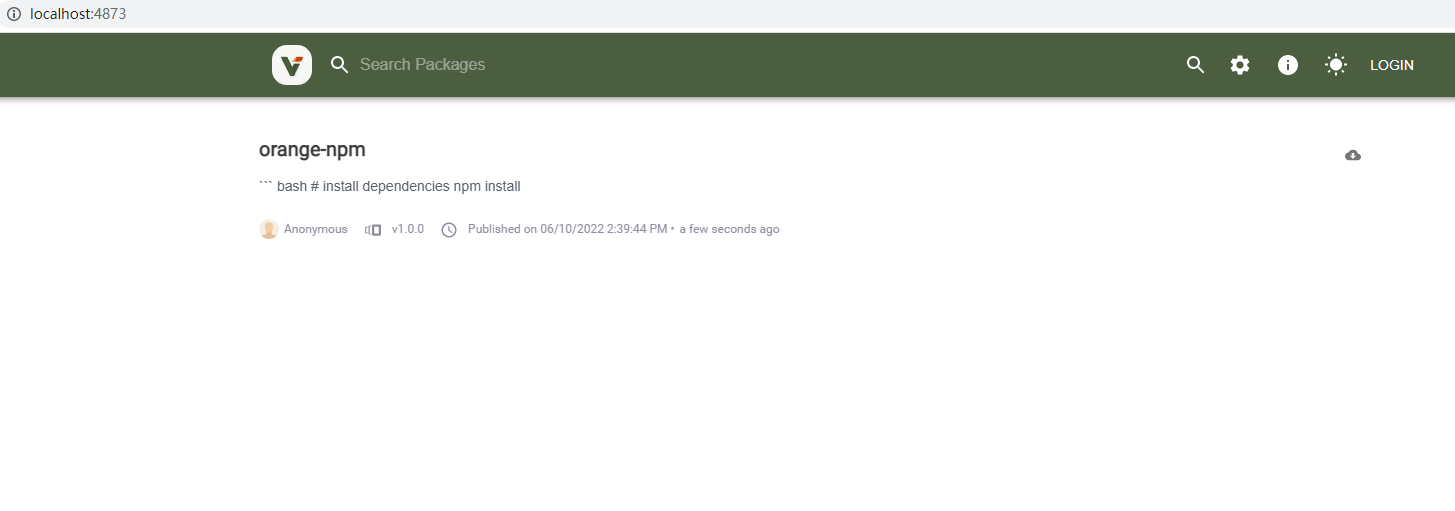


2：发布

执行npm publish 命令进行发布



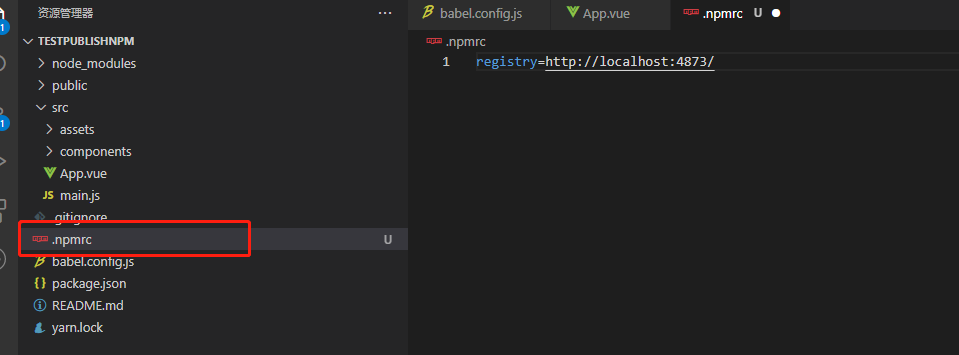
发布成功后，进入可视化界面，就可以看到orange-npm包已经存在



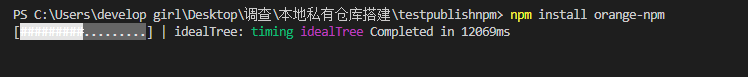
###### 七：本地新建vue项目，尝试下载已发布的npm包



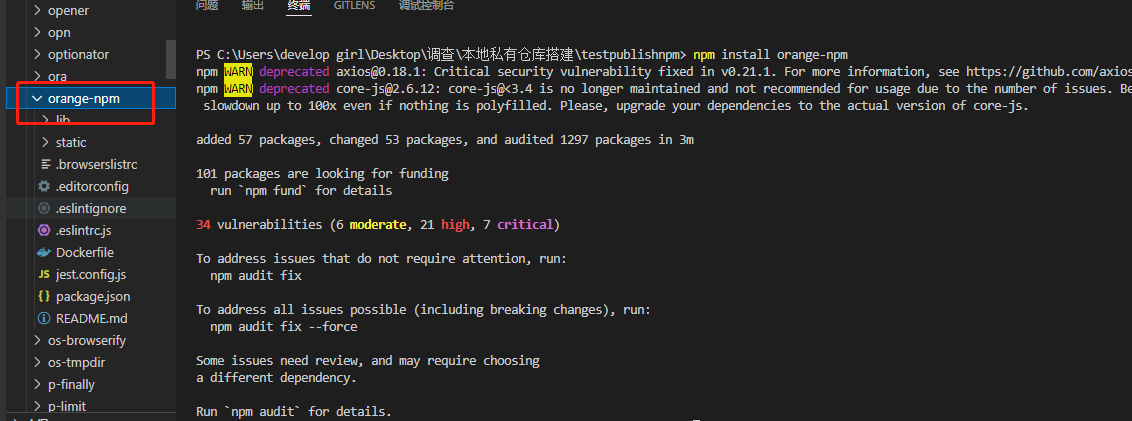
1：在新建的项目中配置.npmrc



2：执行命令：npm install orange-npm



3：安装成功，新项目依赖包中已经存在orange-npm包



4：在新建的项目中使用导出的组件myMessage