目录

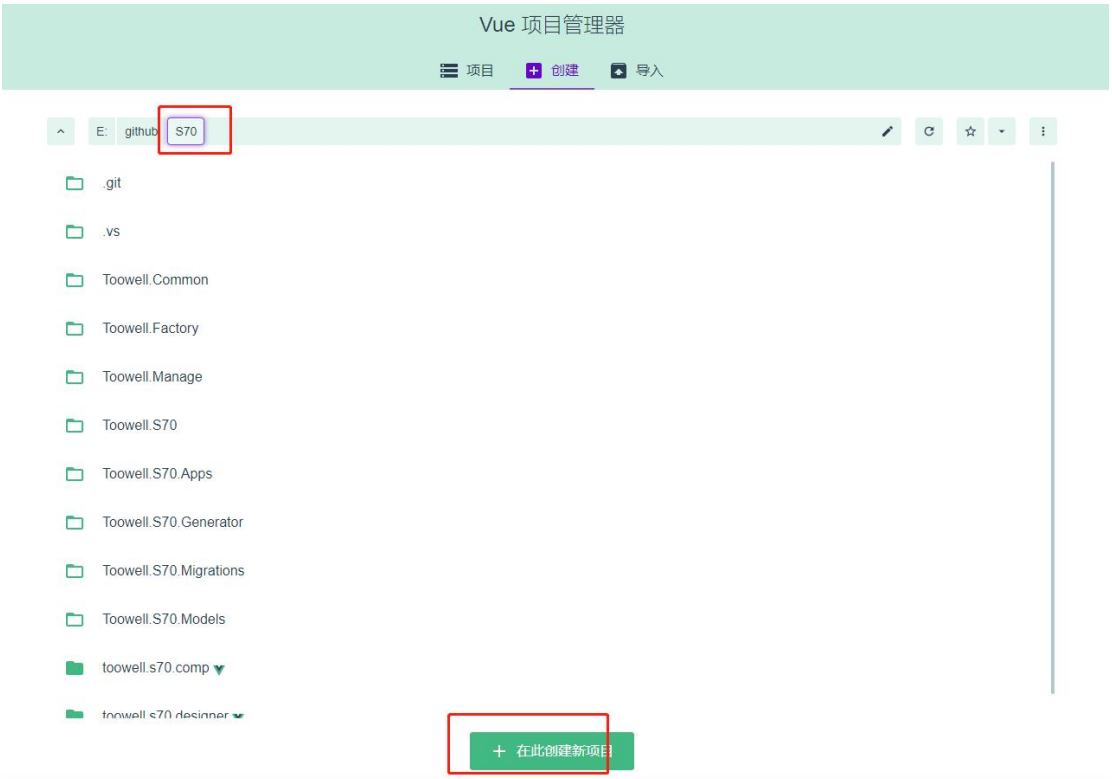
S70 自定义组件介绍

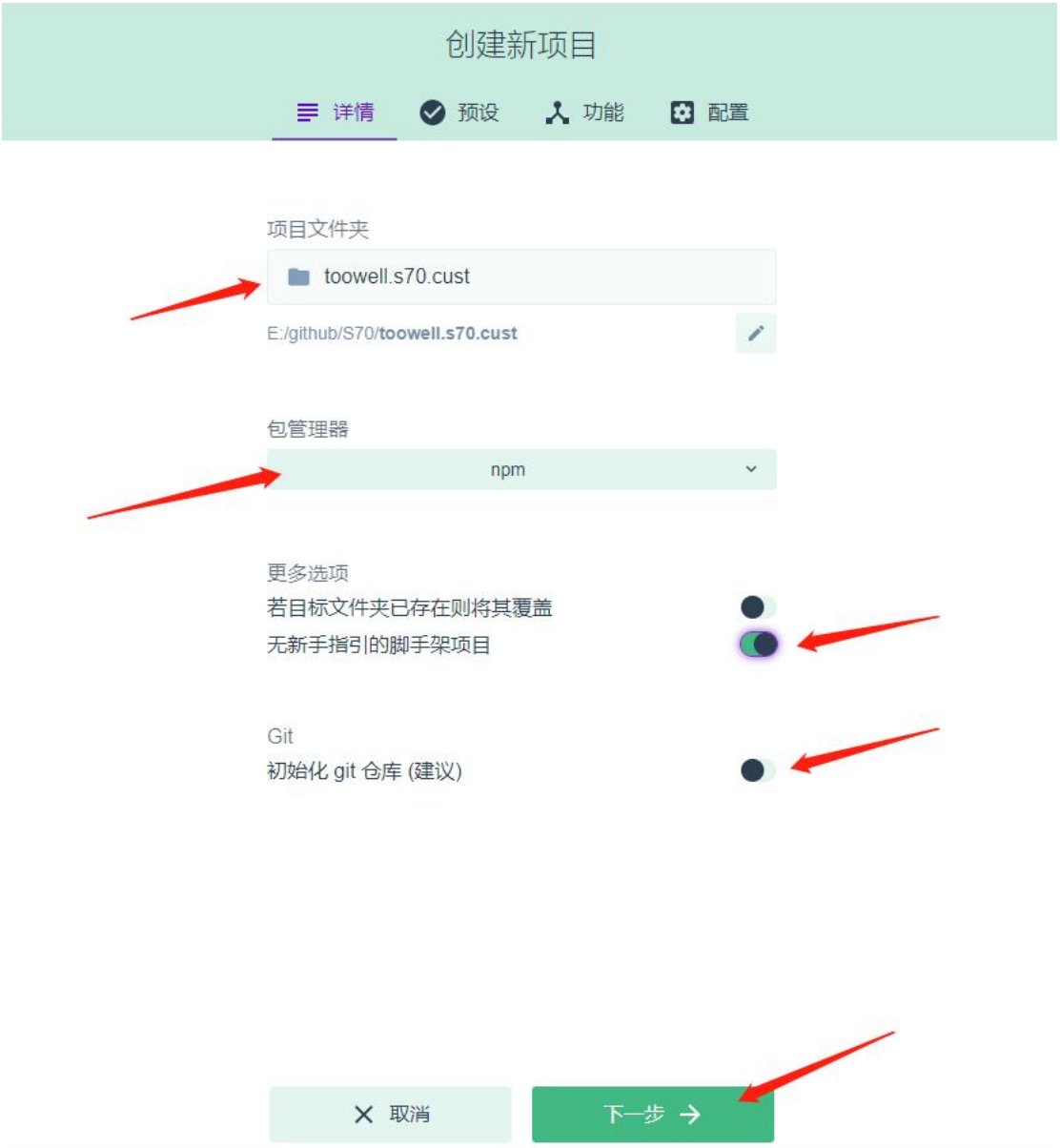
1. 启动 Vue 项目管理器

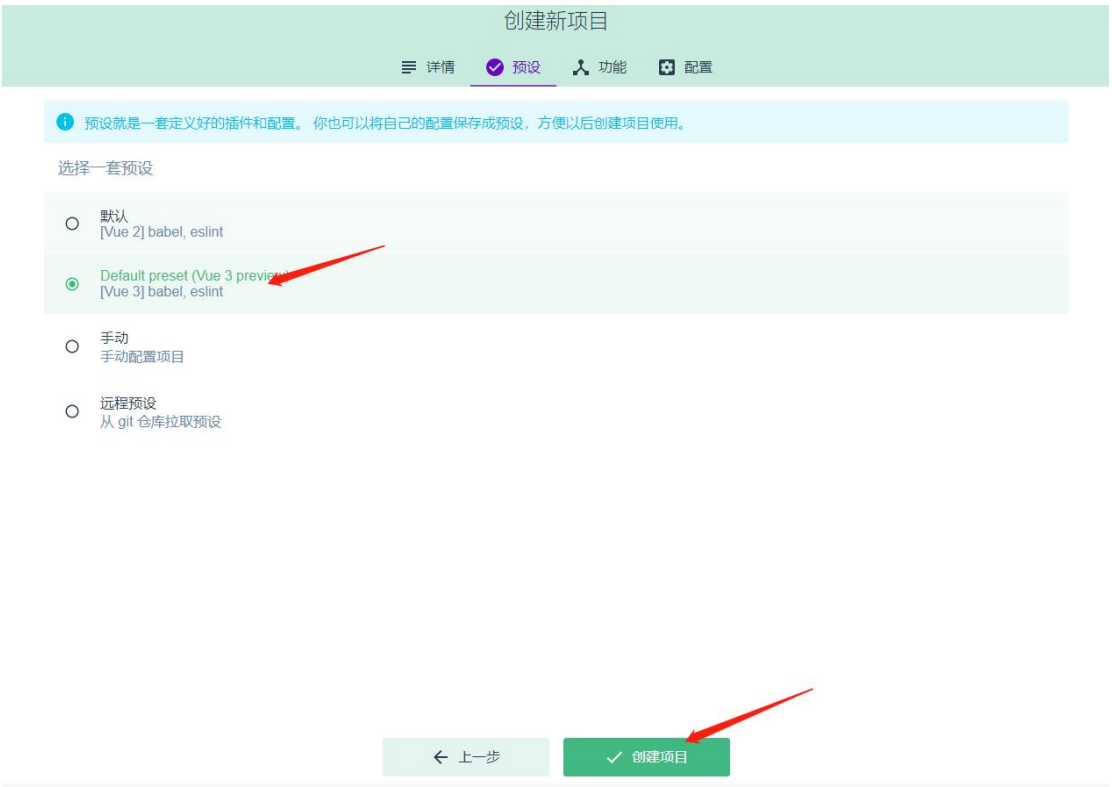
CMD 中输入 vue ui 启动 Vue 项目管理器

2. 创建自定义前端组件项目


Vue 项目管理器
项目
创建
导入

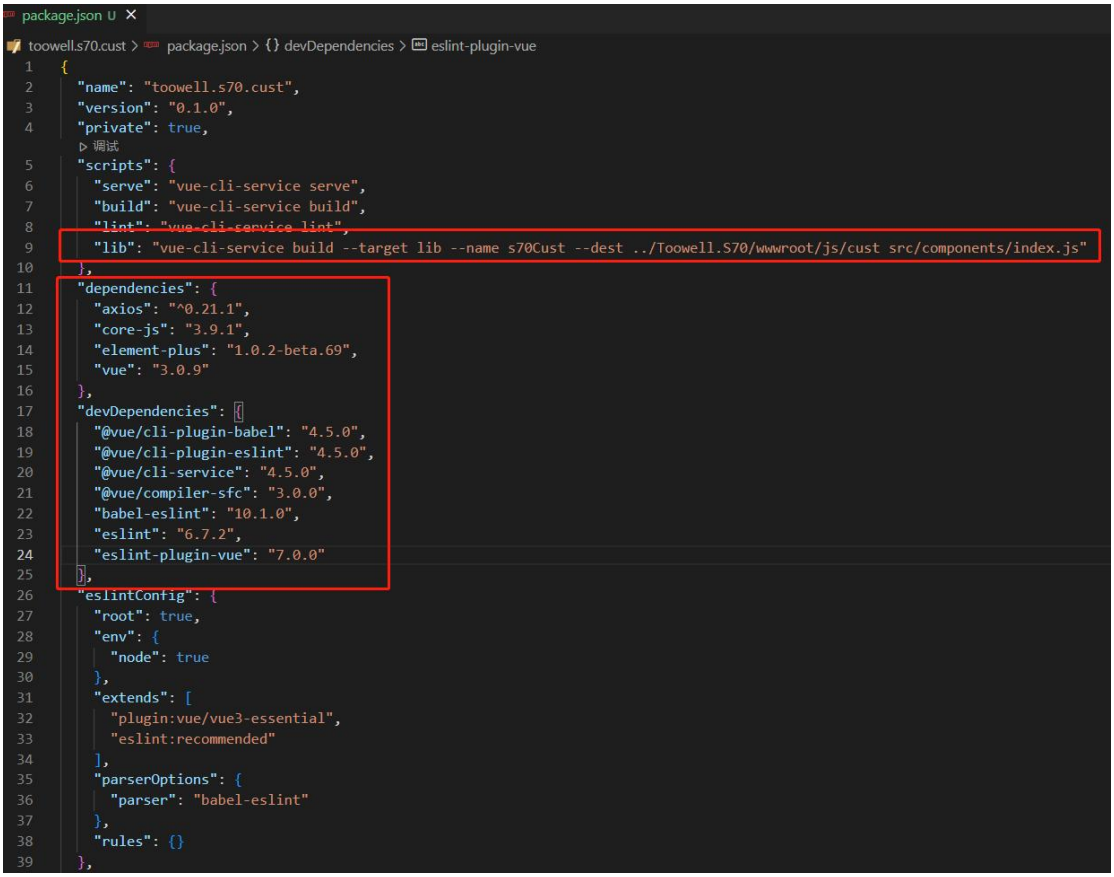

Vue 项目管理器
项目
创建
导入
E:
github
S70
.git
.vs
Toowell.Common
Toowell.Factory
Toowell.Manage
Toowell.S70
Toowell.S70.Apps
Toowell.S70.Generator
Toowell.S70.Migrations
Toowell.S70.Models
toowell.s70.comp
+ 在此创建新项目


创建新项目
详情
预设
功能
配置
项目文件夹
toowell.s70.cust
E:/github/S70/toowell.s70.cust
包管理器
npm
更多选项
若目标文件夹已存在则将其覆盖
无新手指引的脚手架项目
Git
初始化 git 仓库 (建议)
取消
下一步



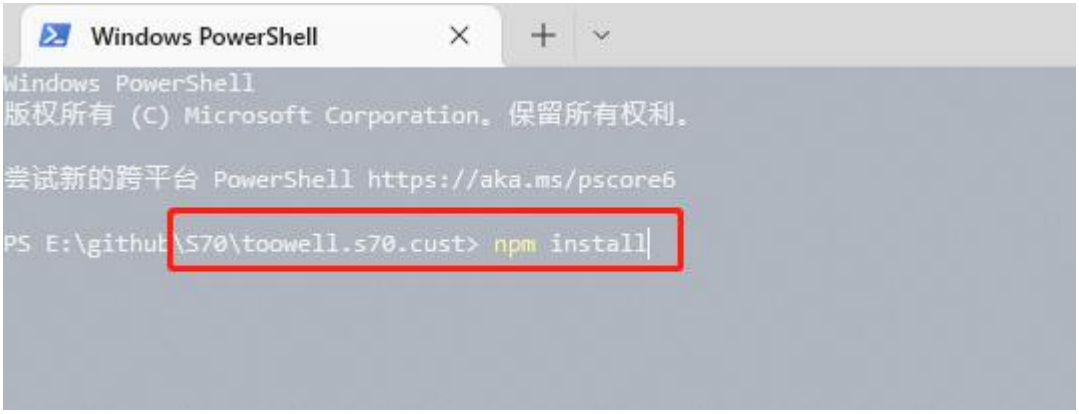
3. 项目依赖配置

修改 package.json，增加 lib 编译配置及依赖保持和 S70Comp 项目一致，配置如下



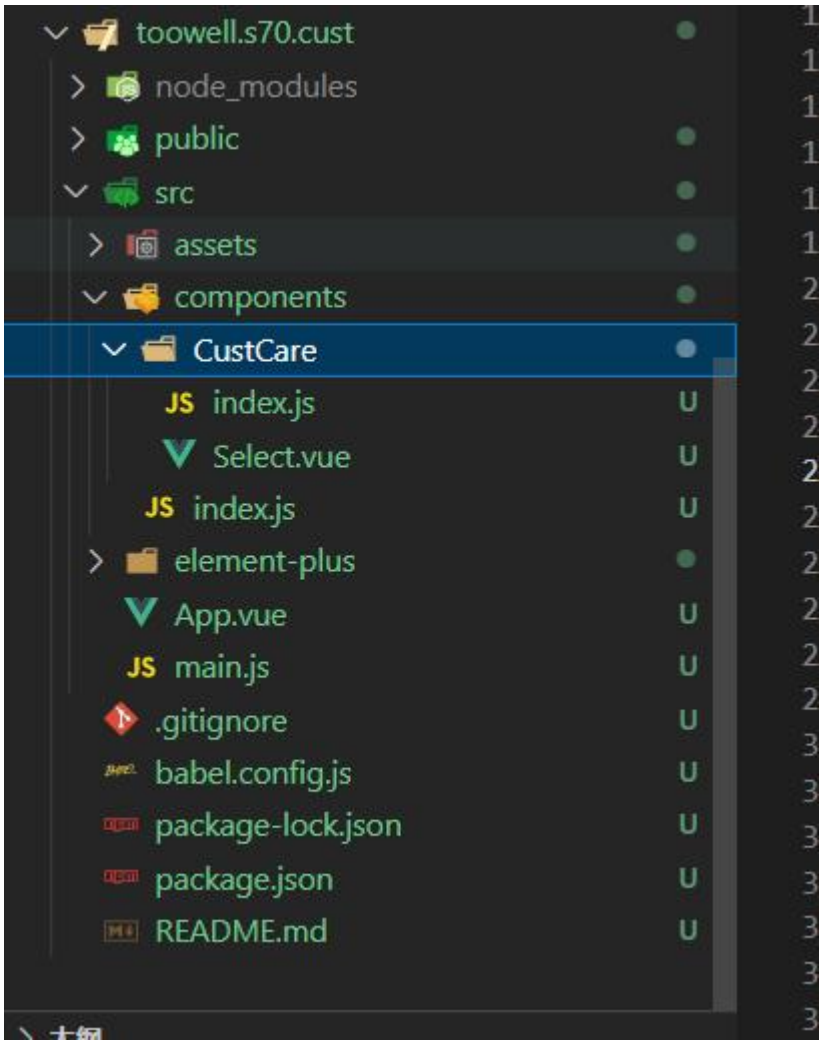
4. 安装项目依赖

CMD 进入项目目录，执行 npm install 安装项目依赖



5. 创建自定义组件

使用 VS Code 打开自定义组件项目创建自定义组件



在 components 目录中创建 index.js 和 CustCare 目录，在 CustCare 目录创建.vue 和 index.js 文件：

index.js 为组件注册和导出方法，参考

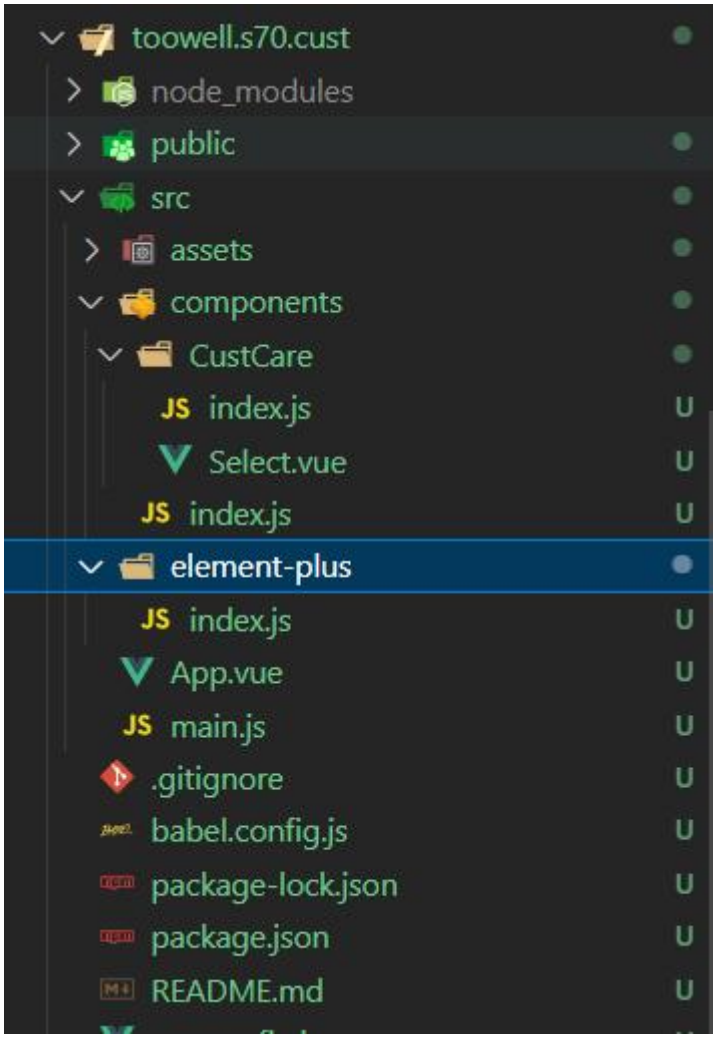
https://v3.cn.vuejs.org/guide/component-registration.html

.vue 文件为单文件组件， 参考
https://v3.cn.vuejs.org/guide/single-file-component.html#%E4%BB%8B%E7%BB%8D

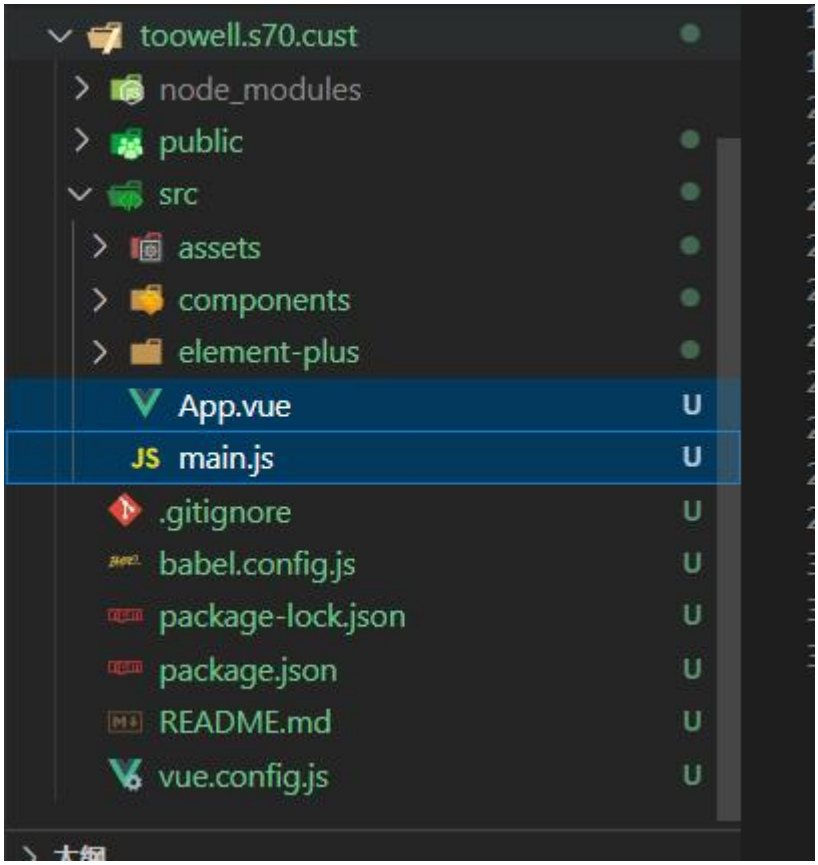
CustCare 中 index.js 为编写插件公开方法， 参考
https://v3.cn.vuejs.org/guide/plugins.html#%E7%BC%96%E5%86%99%E6%8F%92%E4%BB%B6

Element-plus js 版文档示例，参考
https://github.com/element-plus/element-plus/tree/1.0.2-beta.69/website/docs/zh-CN

6. 组件前端调测
创建 element-plus 目录和 index.js，index.js 为按需引入 element-plus 组件内容



修改 App.vue,main.js 文件测试组件



package.json U App.vue U

toowell.s70.cust > src > App.vue > {} "App.vue" > script > default > data > custcare

```vue
<template>
  <mx-cust-care-select v-model="custcare" placeholder="请选择"></mx-cust-care-select>
</template>

<script>
export default {
  name: "App",

  data: function () {
    return {
      custcare: "",
    };
  },
};
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

App.vue U JS main.js U

toowell.s70.cust > src > JS main.js > ...

```
1   import {
2       createApp
3   } from 'vue'
4   import ElementPlus from '@/element-plus'
5   import 'element-plus/lib/theme-chalk/index.css'
6   import S70Cust from '@/components'
7   import App from './App.vue'
8
9   /* 前后端联调时需要token认证
10  S70Cust.axios.post(`http://localhost:5000/api/v1/token`, {
11      "UserName": "admin",
12      "Password": "123456"
13  }).then((res) => {
14      let token = res.data.token;
15      console.log(token);
16      const app = createApp(App)
17      app.use(ElementPlus)
18      app.use(S70Cust, {
19          baseURL: "http://localhost:5000/api/comp/data/",
20          token: token
21      })
22      app.mount('#app')
23  })
24  */
25
26  const app = createApp(App)
27  app.use(ElementPlus)
28  app.use(S70Cust, {})
29  app.mount('#app')
```

7. 启动调测

8. 编译自定义组件

package.json 的 lib 指定组件输出目录

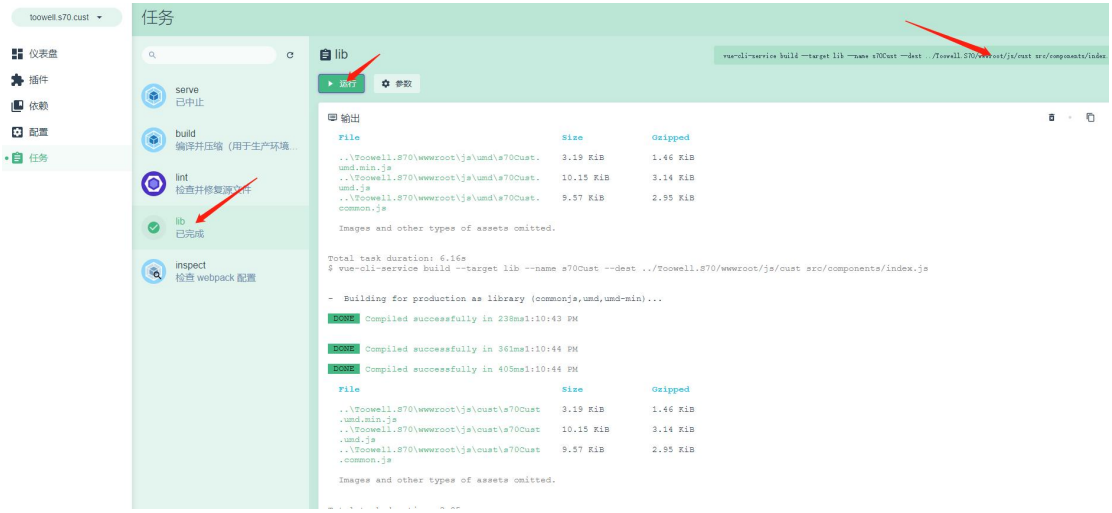
```
{
  "name": "toowell.s70.cust",
  "version": "0.1.0",
  "private": true,
  ▷ 调试
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint",
    "lib": "vue-cli-service build --target lib --name s70Cust --dest ../Toowell.S70/wwwroot/js/cust src/components/index.js"
  },
```

src 目录增加 vue.config.js 文件，内容如下：

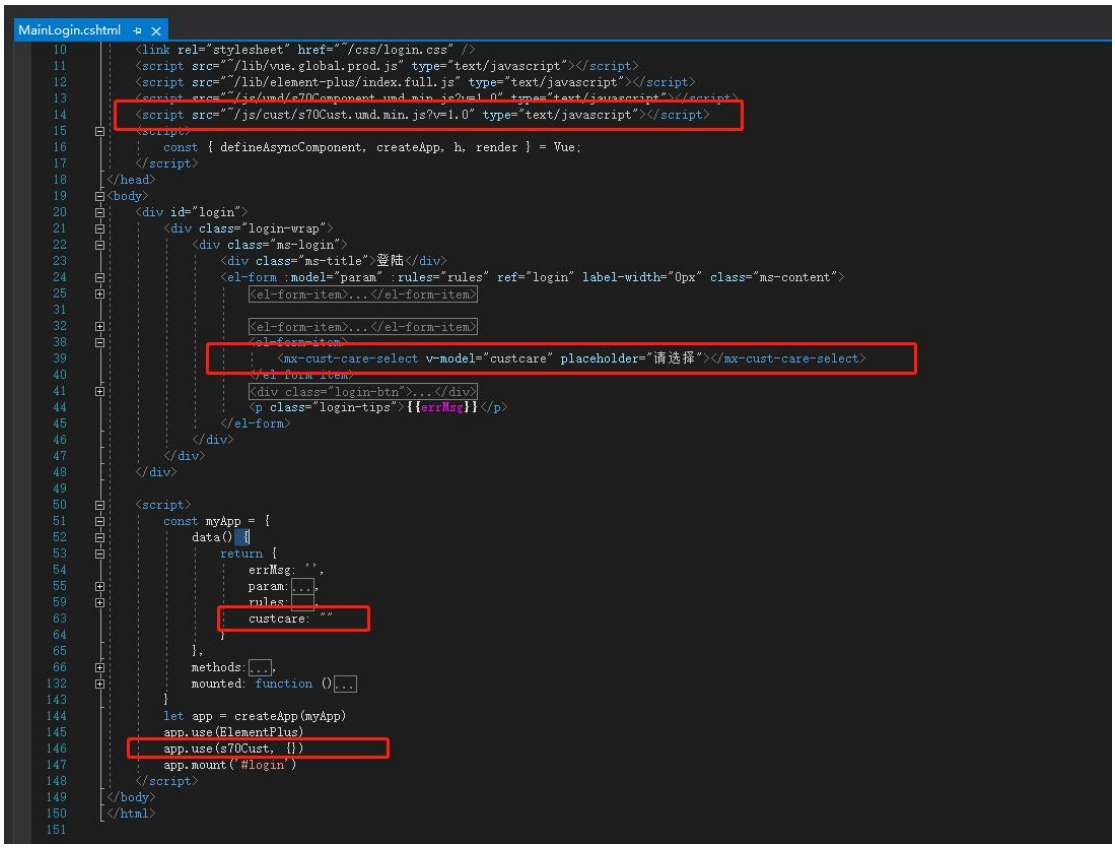
package.json U | App.vue U | vue.config.js U

toowell.s70.cust > vue.config.js > ...

```
function getProdExternals() {
    return {
        // axios: "axios",
        vue: "Vue"
    };
}
module.exports = {
    productionSourceMap: false,
    configureWebpack: {
        externals: process.env.NODE_ENV === 'production' ?
            getProdExternals() : {}
    },
};
```



9. 后端项目引用自定义组件


MainLogin.cshtml
<link rel="stylesheet" href="~/css/login.css" />
<script src="~/lib/vue.global.prod.js" type="text/javascript"></script>
<script src="~/lib/element-plus/index.full.js" type="text/javascript"></script>
<script src="~/js/cust/s70Cust.umd.min.js?v=1.0" type="text/javascript"></script>
<script>
const { defineAsyncComponent, createApp, h, render } = Vue;
</script>
</head>
<body>

登陆
<el-form :model="param" :rules="rules" ref="login" label-width="0px" class="ms-content">
<el-form-item>...</el-form-item>
<el-form-item>...</el-form-item>
<el-form-item>
<mx-cust-care-select v-model="custcare" placeholder="请选择"></mx-cust-care-select>
</el-form-item>
...
{{errMsg}}
</el-form>

<script>
const myApp = {
data() {
return {
errMsg: '',
param:...,
rules:...,
custcare: ""
}
},
methods:...,
mounted: function ()...
}
let app = createApp(myApp)
app.use(ElementPlus)
app.use(s70Cust, {})
app.mount('#login')
</script>
</body>
</html>