

# 自定义窗体开发使用手册

同望·银弹谷·平台研发部

## ► 基础知识

高效的开发，离不开基础工程的搭建。在开始自定义窗体开发之前，有必要先了解以下基础知识：

Vue 教程 <https://cn.vuejs.org/v2/guide/>

Vue API <https://cn.vuejs.org/v2/api/>

Vue 组件 <https://cn.vuejs.org/v2/guide/components.html>

Vue 风格指南 <https://cn.vuejs.org/v2/style-guide/>

Vue 单页面开发 <https://segmentfault.com/a/1190000005667546>

Vui 控件标签使用指南 [http://service.yindangu.com:8018/module-operation!executeOperation?componentCode=divusermanual&windowCode=div\\_user\\_guide#top\\_nav\\_menu](http://service.yindangu.com:8018/module-operation!executeOperation?componentCode=divusermanual&windowCode=div_user_guide#top_nav_menu)

## ► 环境准备

- ◆ 窗体自定义开发基于nodejs环境，如果本机没有安装nodejs，请到官网（<https://nodejs.org/en/>）下载nodejs，并安装。
- ◆ 本地开发推荐编辑器VSCode（<https://code.visualstudio.com/>）



```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\chenf>node -v
v8.9.1

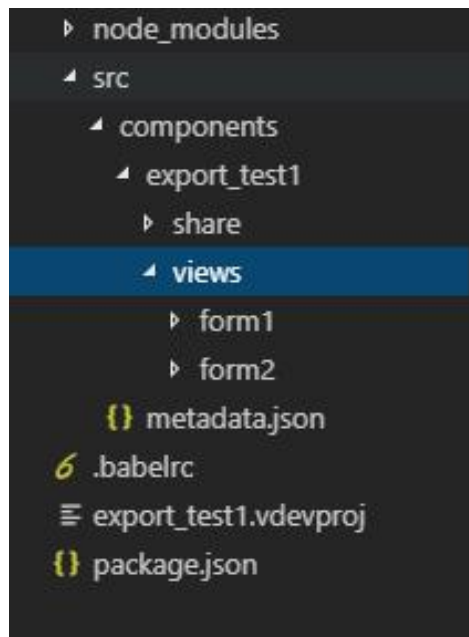
C:\Users\chenf>npm -v
5.8.0

C:\Users\chenf>
```

## ► 使用指令

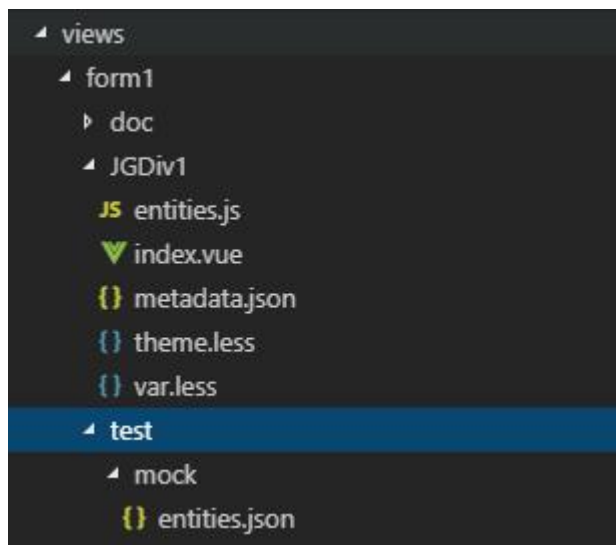
- 1、v3:package 使用webpack打包 `npm run v3:package`
- 2、v3:start 启动本地测试服务 `npm run v3:start`

## ► 工程目录结构



- 1、node\_modules：生成的相关依赖模块（无需关注）
- 2、src：源代码
- 3、export\_test1：构件名称
- 3、share：构件内共享资源文件 js、css、图片
- 4、views：窗体集合
- 5、form1、form2：窗体名称
- 6、package.json：配置说明

## ► 工程目录结构



- 1、entities.js: 数据实体定义 (无需关注)
- 2、index.vue: 窗体页面 (HTML、CSS、JS)
- 3、metadata.json: 工程元信息 (无需关注)
- 3、theme.less: 窗体样式定义
- 4、var.less: 窗体样式变量定义
- 5、form1、form2: 窗体名称
- 6、entities.json: 测试数据

## ▶ 已生成的测试数据及应用

```
entities.json x
1 {
2   "infolist": [
3     {
4       "item": "基本配置",
5       "content": "海纳百川精选全球的高品质软件与服务, 大咖云集, 知识分享的开发者技术社区",
6       "price": "99"
7     },
8     {
9       "item": "标准配置",
10      "content": "海纳百川精选全球的高品质软件与服务, 大咖云集, 知识分享的开发者技术社区",
11      "price": "199"
12    },
13    {
14      "item": "高端配置",
15      "content": "海纳百川精选全球的高品质软件与服务, 大咖云集, 知识分享的开发者技术社区",
16      "price": "299"
17    }
18  ]
19 }
```

```
<i-col :xs="24" :sm="12" :md="8" v-for="infolist in infolist" :key="infolist.id" style="padding-left: 10px;padding-right: 10px;">
  <div class="itemBox">
    <div class="content">
      <h3>{{infolist.item}}</h3>
      <p>{{infolist.content}}</p>
    </div>
    <div class="price">
      <p>¥{{infolist.price}}</p>
    </div>
    <div class="buybtn">
      <a>立即购买</a>
    </div>
  </div>
</i-col>
```

## ▶ 生成的theme.less和var.less

```
1 @import "../theme.less";
2 @import "../var.less";
3
4 .loginbg{
5     position: absolute;
6     top: 0;
7     left: 0;
8     right: 0;
9     bottom: 0;
10    min-height: 568px;
11    min-width: 980px;
12    background: no-repeat left center;
13    background-size: cover;
14    background-image: url(../../share/image_bg.jpg);
15 }
16 .photoBox{
17     text-align: center;
18     margin: 50px 0;
19 }
20 .photoBox img{
21     height: 120px;
22     width: 120px;
23 }
24 .photoBox span{
25     color: @base_color;
26     display: block;
27     font-size: 28px;
28     margin-top: 10px;
29 }
30 .rowBg{
31     background-color: rgba(0,0,0,0.3);
32     padding: 25px;
33 }
```

定义css样式，只作用于本窗体中

样式可以以less的形式定义，具体写法可参考less官方文档：

<http://lesscss.cn/>

```
1 @base_color: #fff;
2 @roundBg: #9ac664;
```

存放定义的样式变量，后期支持换肤功能



## ▶ 引入第三方UI框架、插件

### 方式1: npm安装相关依赖

npm i element-ui -S (安装element)

```
E:\窗体扩展培训\导出工程\div_export_demo>npm i element-ui -S
npm WARN ajv-keywords@3.1.0 requires a peer of ajv@^6.0.0 but none is installed.
  You must install peer dependencies yourself.
npm WARN iview@2.11.0 requires a peer of vue@^2.5.2 but none is installed. You m
ust install peer dependencies yourself.
npm WARN element-ui@2.3.2 requires a peer of vue@^2.5.2 but none is installed. Y
ou must install peer dependencies yourself.
npm WARN div_export_demo@1.0.0 No description
npm WARN div_export_demo@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 <node_modules\fse
vents>:
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.3: wanted <"os": "darwin", "arch": "any"> <current: <"os": "win32", "arch": "x64">
>

+ element-ui@2.3.2
added 5 packages in 48.742s
```

```
import vdk from 'v3-vdk';
import entities from './entities.js';
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
```

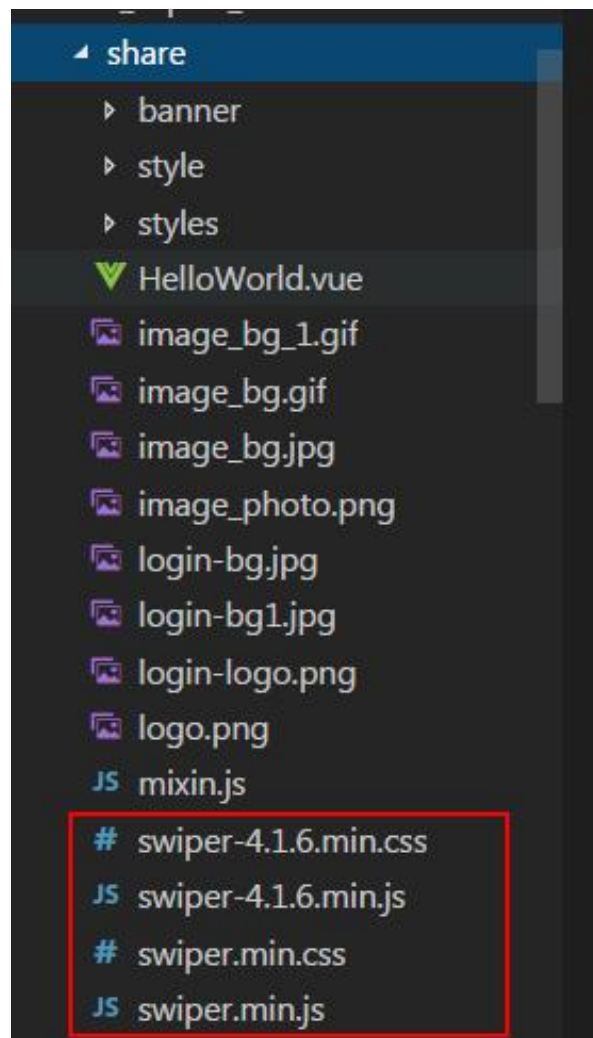
## ▶ 引入第三方UI框架、插件

### 方式2:

手动引入，将相关文件放入share文件夹下面，  
import直接引入相对路径下的文件

例：“@share/xx/xx”

```
import vdk from 'v3-vdk';  
import entities from './entities.js';  
import Swiper from '@share/swiper.min.js';  
import '@share/swiper.min.css';
```



## ▶ 引入vue组件

### vue组件放置于share文件夹

- 1.import引入vue文件
- 2.components定义

```
<template>
  <div class="app bg">
    
    <HelloWorld :blankurl="blankurl" />
  </div>
</template>
<script>
  import vdk from 'v3-vdk';
  import entities from './entities.js';
  import HelloWorld from "@share/HelloWorld.vue";
  export default {
    components: {
      HelloWorld
    },
    props: entities,
    data: function(){
      return {
        msg: "欢迎学习v3开发技术"
      };
    }
  };
</script>
```

## ► 使用指令

### npm run v3:package

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

PS F:\File\2018开发需求\02.窗体自定义开发\导出工程\export_test1> npm run v3:package

> export_test1@1.0.0 v3:package F:\File\2018开发需求\02.窗体自定义开发\导出工程\export_test1
> v3-package

开始打包项目...
开始解析资源依赖....
开始解析资源依赖....
打包完成!
PS F:\File\2018开发需求\02.窗体自定义开发\导出工程\export_test1> []
```

```
└─ export_test1
  └─ build
  └─ dist
  └─ node_modules
  └─ src
  └─ .babelrc
  └─ export_test1.vdevproj
  └─ package.json
```

```
└─ build
└─ dist
  └─ itop
  └─ resources
  └─ vendor
  └─ export_test1_form1_JGDiv1_metadata.json
  └─ export_test1_form1_JGDiv1_previewIndex.js
  └─ export_test1_form1_JGDiv1.html
  └─ export_test1_form2_JGDiv1_metadata.json
  └─ export_test1_form2_JGDiv1_previewIndex.js
  └─ export_test1_form2_JGDiv1.html
  └─ index.html
  └─ index.js
  └─ node_modules
  └─ src
  └─ .babelrc
  └─ export_test1.vdevproj
  └─ package.json
```

## ► 使用指令

**npm run v3:start**

```
PS F:\File\2018开发需求\02.窗体自定义开发\导出工程\export_test1> npm run v3:start  
  
> export_test1@1.0.0 v3:start F:\File\2018开发需求\02.窗体自定义开发\导出工程\export_test1  
> v3-start  
  
开始打包项目...  
开始解析资源依赖....  
开始解析资源依赖....  
启动Http服务器..
```



## ▶ 事件关联

### \$emit为标识

this.\$emit("XXX");

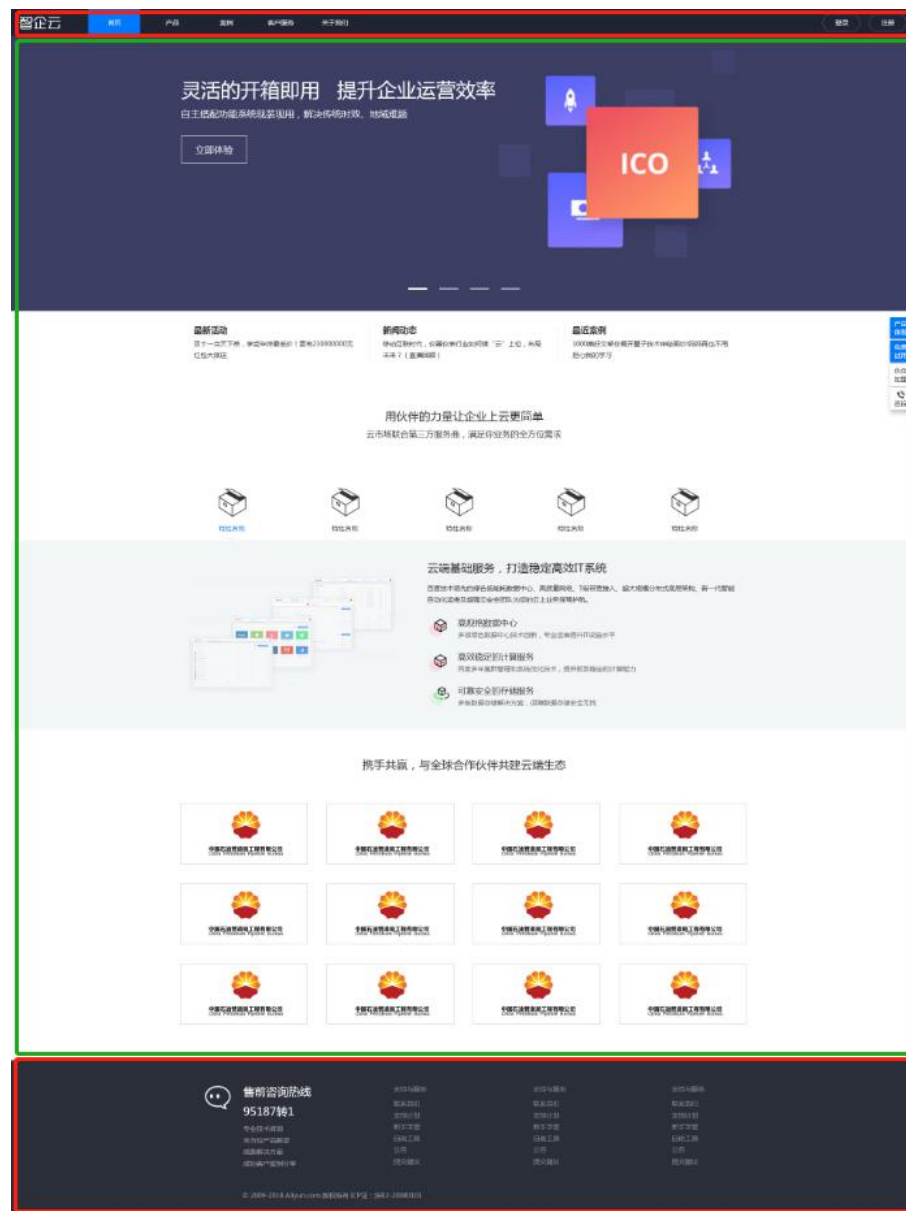
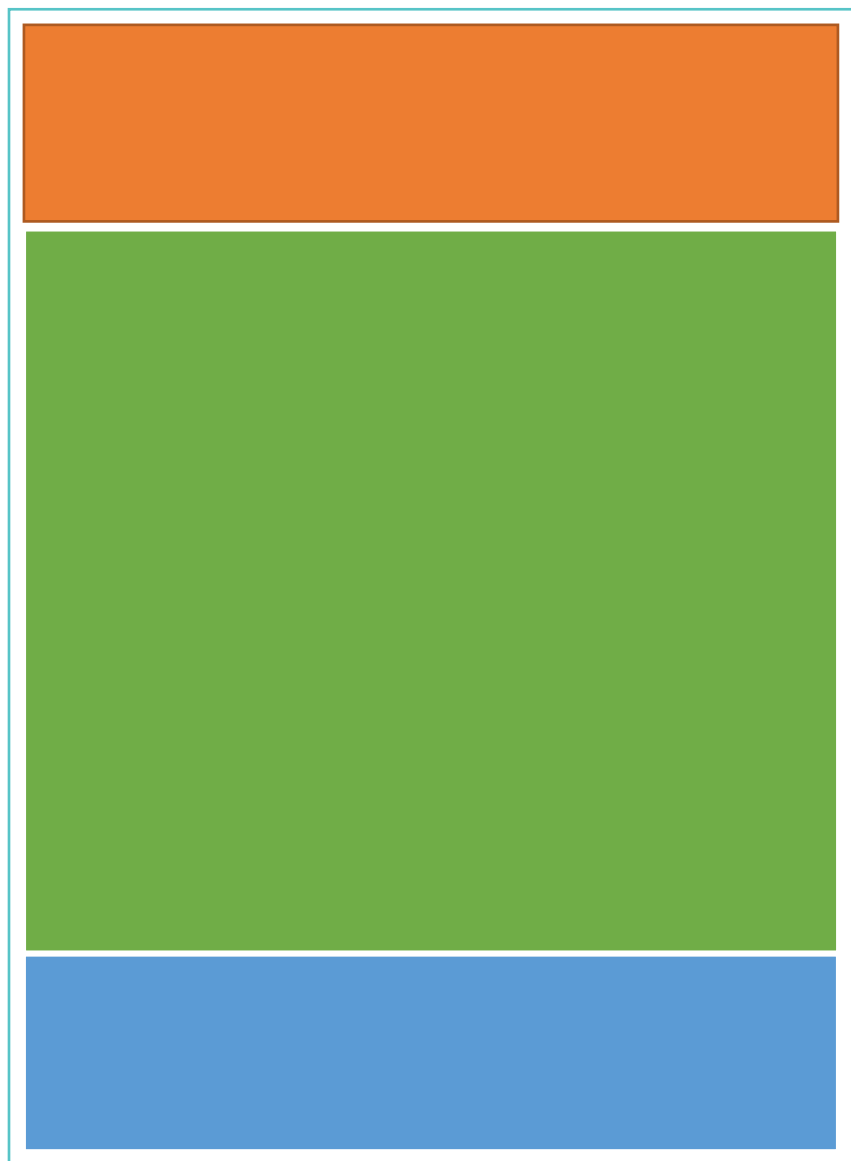
v-on:click="\$emit('XXX')"

```
handleClick : function(){  
  debugger;  
  alert('btn clicked!');  
  this.$emit("eventOne", 'a', 'b', 'c');
```

```
<i-button type="primary" @click="handleClick">Primary</i-button>  
<i-button type="success" v-on:click="$emit('eventTwo')">success</i-button>
```



## 开发规范

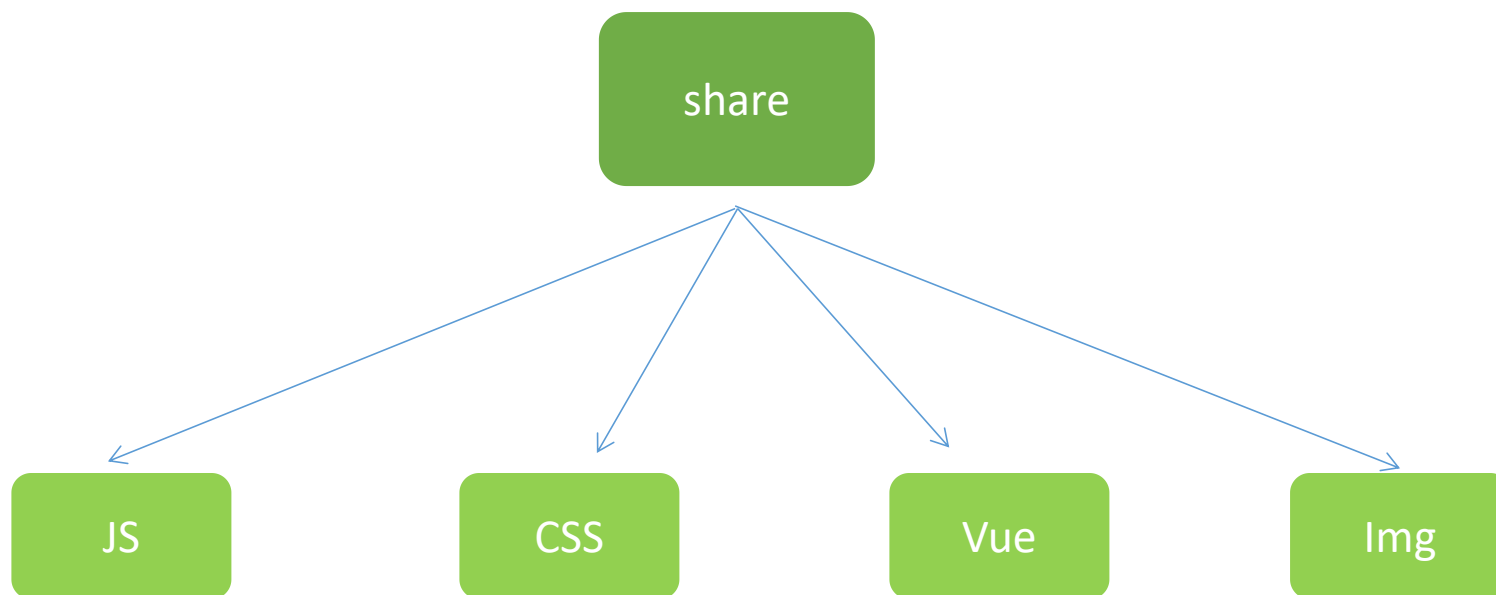


页面中可复用的  
区块都抽离成独  
立的Vue组件

## ► 开发规范

所有独立封装的Vue组件、引用的第三方js、css、本地资源图片全部放置于share目录之下。

引用share下面的资源统一使用路径：“@share/xx/xx”





## ► 开发规范

```
.box{  
  background: @v-primary-color;  
  height: 100px;  
  width: 100px;  
  color: @v-text-title-color;  
}
```

页面样式部分推荐使用平台本身提供的默认样式变量，如果在share下面的Vue组件使用样式变量，可在Vue组件的页面里面引入：

```
@import '~@vPlatformBaseSkinVar';
```

```
@import '~@vPlatformWebSkinVar';
```

```
@import '~@vPlatformMobileSkinVar';
```

针对网页窗体开发目前有以下约束和建议：

- 1、不可引用【Bootstrap】UI库及其相关组件
- 2、V平台已内置第三方Vue UI库【iView 2.0】，可直接使用iView标签
- 3、V平台有提供内置官方UI库【VUI】，建议在面向企业应用类场景使用

# THANK YOU

