

控件自定义开发指南

同望·银弹谷·平台研发部

功能需求背景

- ◆ 脱离V平台开发系统独立开发
- ◆ 开发自主设计的控件
- ◆ 灵活引用第三方插件
- ◆ 满足项目实际需求

- ◆ 1 搭建本地开发环境、初始化控件工程
- ◆ 2 本地开发自定义控件，本地编译测试
- ◆ 3 部署到vstore或指定服务地址
- ◆ 4 V3页面使用已安装自定义控件

▶ 环境准备

- ◆ 控件自定义开发基于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>
```

▶ 开发控件规范

在开始自定义控件开发之前，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>

单文件组件 <https://cn.vuejs.org/v2/guide/components.html>

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

以下概念贯穿整个开发前后，建议开发者花点时间来了解。

props 传递数据

slot 内容分发

events \$emit @click 事件

你也可以通过 【[学习组件基础内容](https://segmentfault.com/l/1500000009448056)】 来快速入门 <https://segmentfault.com/l/1500000009448056>

▶ 初始化工程

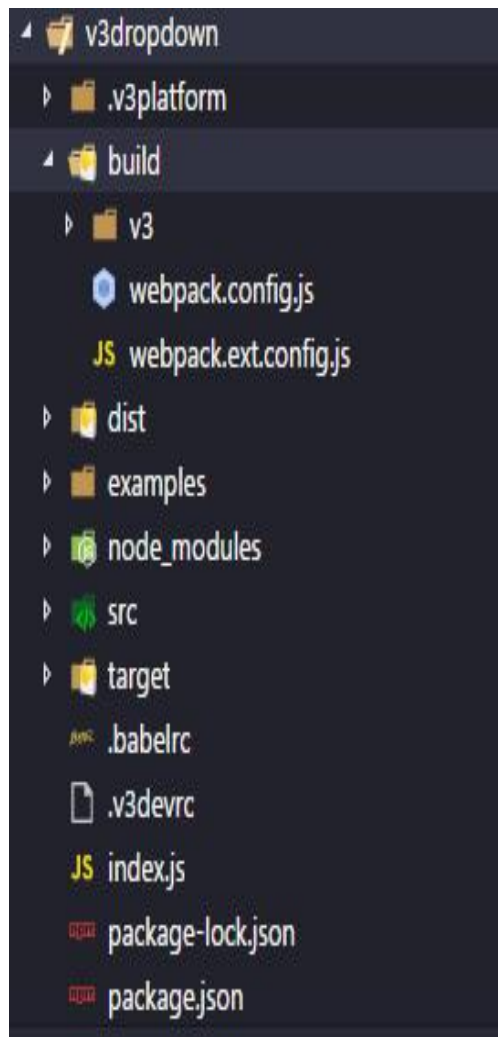
- 1、新建一个目录，如：example
- 2、在cmd命令行中进入目录 `cd example`
- 3、执行命令：`npm install v3-cli -save-dev`
- 4、根据指引设置控件信息，设置完成后，v3-cli脚手架会自动生成模板工程

► 初始化配置信息

```
V3 WARN: 当前项目不存在.v3devrc配置文件!  
V3 WARN: 当前项目不存在.v3devrc配置文件!  
? Vstore仓库[1.dev(开发库), 2.test(测试库),3.其他私库]类型: 2DEV  
? 插件类型( widget:二次开发控件 resource:资源构件 ): widget  
? 项目(构件)名称: v3dropdown  
? 组织id: com.toone.v3.platform  
? 匹配版本号: 1  
? 应用范围: 1.all(全部), 2.web(网页版), 3.mobile(移动版): all  
初始化package.json  
V3二次开发插件初始化成功!
```

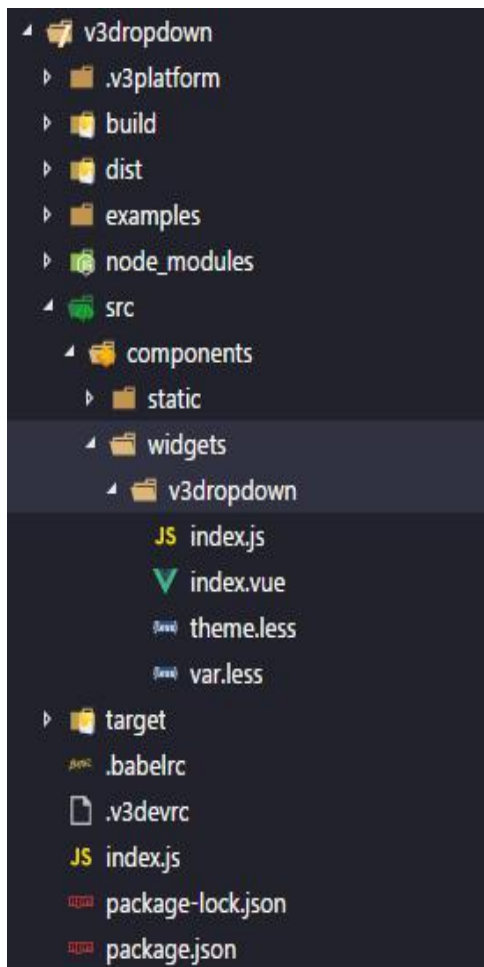
- 1、vstore仓库: 发布至vstore的指定仓库
- 2、插件类型: 生成的自定义控件的类型, widget为二次开发控件, resource为资源构件
- 3、项目(构件)名称: 自定义
- 4、组织id: 默认平台id, 可自拟
- 5、匹配版本号: 默认1
- 6、应用范围: V平台适用窗体类型(全部、网页、移动)

▶ 模板工程目录说明



- 1、**build**文件夹：存放webpack配置文件信息
 - A、**webpack.config.js** webpack打包配置主入口（**请不要更改**）
 - B、**webpack.ext.config.js** 控件额外配置文件，如果控件需要额外定制打包配置，请调整此文件。
 - C、**v3**文件夹：存放v3平台webpack打包配置文件（**请不要更改**）
- 2、**dist**文件夹：存放输出文件
- 3、**examples**文件夹：存放样例文件
- 4、**node_modules**：生成的相关依赖
- 5、**.babelrc**：Babel转译配置项
- 6、**.v3devrc**：v3平台二次开发配置信息
- 7、**package.json**：配置说明
- 8、**package-lock.json**：当前安装的包相关的依赖

▶ 模板工程说明



src文件夹：存放源码文件

A、static文件夹：存放静态文件

B、widgets文件夹：存放控件定义信息

① index.js文件：控件注册信息（**请不要更改**）

② index.vue文件：控件定义

③ theme.less文件：控件样式class定义

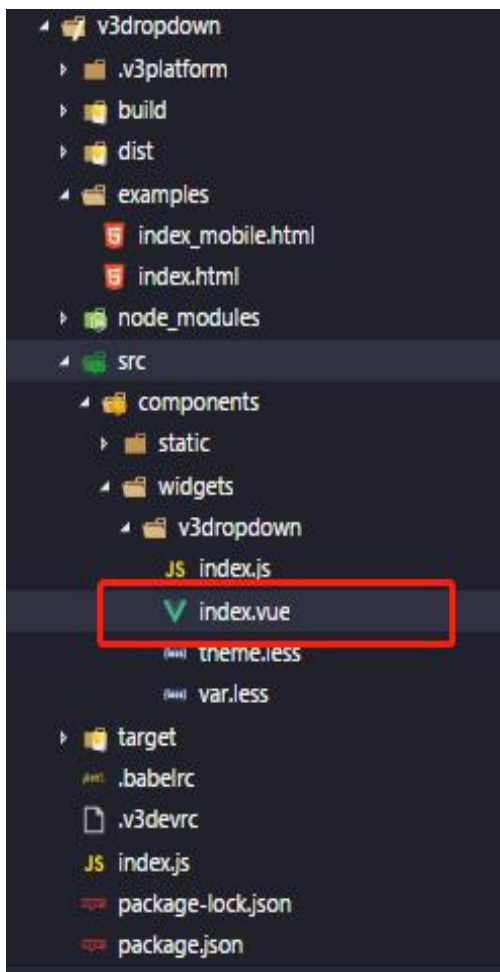
④ var.less文件：控件样式变量定义

▶ 可执行命令

- 1、v3:package 使用webpack打包控件 `npm run v3:package`
- 2、v3:test/v3:start 启动本地测试服务 `npm run v3:start`
- 3、v3:pack 将控件打包成vstore构件 `npm run v3:pack`
- 4、v3:publish 将控件部署到vstore仓库 `npm run v3:publish`
- 5、v3:apply 将控件安装至指定的执行系统 `npm run v3:apply`
- 6、v3:init 更新控件的webpack配置文件 `npm run v3:init`
- 7、v3:install 给控件安装平台的资源构件 `npm run v3:install xxx`
- 8、v3:uninstall 将控件中安装的资源构件选择性卸载 `npm run v3:uninstall xxx`

▶ 控件开发实例

widgets文件夹 index.vue, 编写控件代码

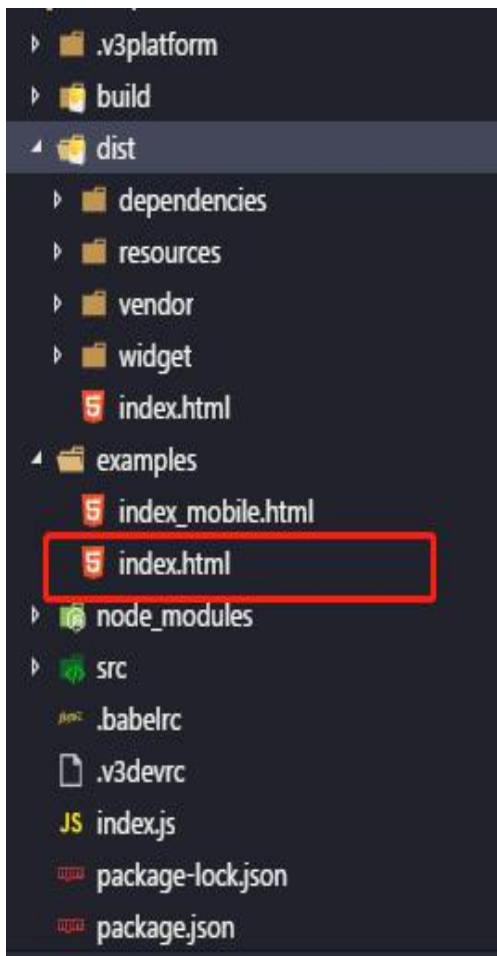


```
1 <template>
2   <div id = "v3Dropdown">
3     <button @mouseover = "isHidden = false" >
4       {{title}}
5       <span class = "downBtn"></span>
6     </button>
7     <div class = "dropdown-box" :class = "{hidden:isHidden}" @mouseout = "isHidden = true">
8       <ul>
9         <li v-for = "(item,index) in items" @click = "onclick(index)" :key = "index">{{item.text}}</li>
10      </ul>
11    </div>
12  </div>
13 </template>
14 <script>
15 export default {
16   name : "v3dropdown",
17   data(){
18     return {
19       isHidden : true,
20       items: []
21     }
22   },
23   props : {
24     itemSource: {
25       type: Array,
26       default: function() {
27         return [];
```

V3脚手架已内置iview 2.0,
可直接使用iview 标签, 具体
可参考:
<http://v2.iviewui.com/>

编写本地样例

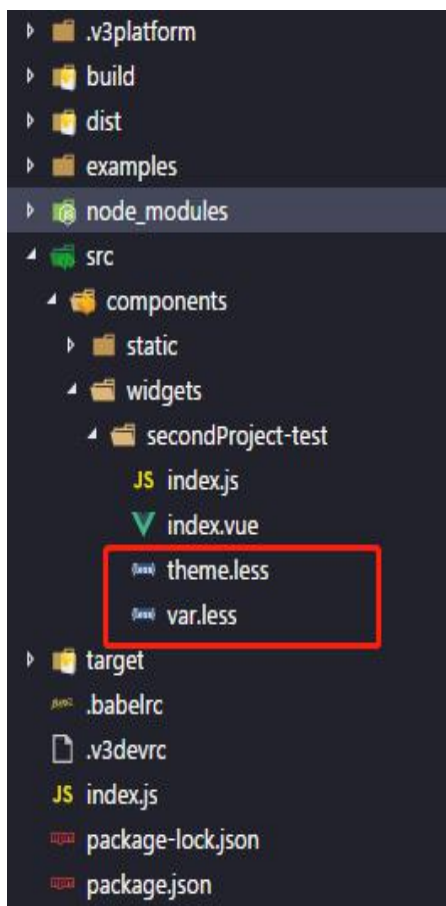
examples文件夹 index.html 本地写测试代码页面



```
<? index.html x
6 </head>
7 <body>
8   <div id="app">
9   </div>
10   <% var scripts = htmlWebpackPlugin.options.resources.scripts;if(scripts){for(var i = 0,len=scripts.
11 </body>
12 <script type="text/javascript">
13   new Vue({
14     el : "#app",
15     template : '<v3dropdown :title = "title" :dropSource = "dataSource" @on-click="click1"></v3dro
16     data : {
17       dataSource : [
18         {
19           'title' : 'test1',
20           'id' : '1'
21         },
22         {
23           'title' : 'test2',
24           'id' : '2'
25         },
26         {
27           'title' : 'test3',
28           'id' : '3'
29         },
30         {
31           'title' : 'test4',
32           'id' : '4'
33         }
34       ],
35       title : '下拉菜单'
36     },
37   },
38
```

▶ 控件样式引入皮肤变量

在var.less文件中引入皮肤变量：`@vPlatformMobileSkinVar` 或 `@vPlatformWebSkinVar`，在theme.less中使用

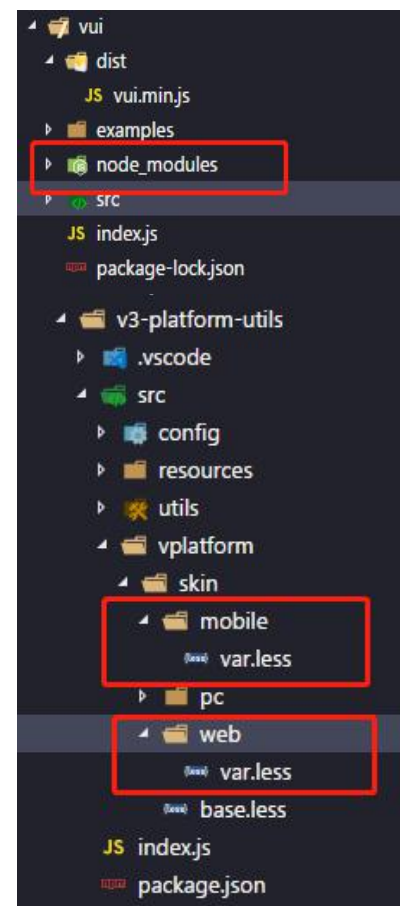


```
var.less x
1
2 //引入平台基础皮肤变量
3 @import '~@vPlatform/skin/base.less';
4
5 @import '~@vPlatformMobileSkinVar';
6 @import '~@vPlatformWebSkinVar';
7
8
9
10
11 .test1{
12   width:100px;
13   height:100px;
14   background:@primary-color;
15 }
16
17 .test2{
18   width:100px;
19   height:100px;
20   background:@m-success-color;
21 }
```

移动版皮肤变量

网页版皮肤变量

皮肤变量内容：在项目的node_modules目录下v3-platform-utils中，mobile下的var.less文件为移动版，web下的var.less为网页版



▶ 控件安装资源构件

执行 `npm run v3:install xxx`，有本地安装和从vstore仓库安装两种方式

```
PS D:\work\test\v3dropdown> npm run v3:install "D:\work\test\v3dropdown\src\components\static\com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-m-f7Resource-1.0.0.jar"
```

> v3dropdown@1.0.0 v3:install D:\work\test\v3dropdown
> v3 install "D:\work\test\v3dropdown\src\components\static\com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-m-f7Resource-1.0.0.jar"

开始安装本地v3平台插件，请稍候...
正在安装 vui-m-f7Resource ...
vui-m-f7Resource 安装完成。
安装本地v3平台插件完成！
安装本地v3平台插件完成！
PS D:\work\test\v3dropdown>

本地安装：资源构件生成的jar包所在的绝对路径

```
PS D:\work\test\v3dropdown> npm run v3:install vui-m-f7Resource
```

> v3dropdown@1.0.0 v3:install D:\work\test\v3dropdown
> v3 install "vui-m-f7Resource"

? 库类型: other
? 请输入库类型: vuiM

? 共检索到 1 个构件, 请选择要安装的构件: com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-m-f7Resource

开始从VStore中安装v3平台插件，请稍候...
下载插件: com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-m-f7Resource
正在安装 vui-m-f7Resource ...
vui-m-f7Resource 安装完成。
从VStore中安装v3平台插件完成！
安装本地v3平台插件完成！

Vstore仓库安装：资源构件的构件名

选择资源构件所在的Vstore仓库，二次开发构件要与资源构件在同一个仓库中

▶ 控件卸载资源构件

执行 `npm run v3:install xxx`

```
PS D:\work\test\v3dropdown> npm run v3:uninstall vui-m-f7Resource  
  
> v3dropdown@1.0.0 v3:uninstall D:\work\test\v3dropdown  
> v3 uninstall "vui-m-f7Resource"  
  
? 找到本项目中的以下相关依赖插件,请选择需要卸载的插件: com.toone.v3.platform-vjs.framework.extension.platform.custom.resource.vui-m-f7Resource  
卸载插件vui-m-f7Resource...  
插件vui-m-f7Resource已卸载.  
插件卸载完成!
```

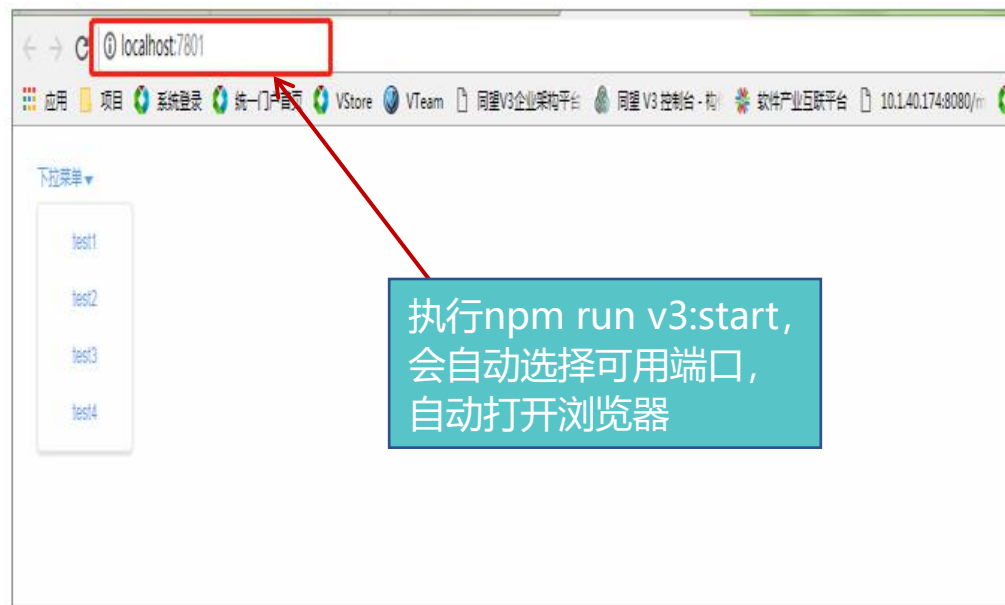
需要卸载的资源构件的构件名称

本地测试服务编译预览

执行 `npm run v3:start`，可以启动本地测试服务实时预览效果

```
D:\work\v平台二次开发 vui\测试\v3dropdown>npm run v3:start
> v3dropdown@1.0.0 v3:start D:\work\v平台二次开发 vui\测试\v3dropdown
> v3 test

正在启动本地测试服务, 请稍候...
Starting server on http://localhost:7801
10% building modules
11% building modules
71% sealing
72% optimizing
73% basic module optimization
74% module optimization
75% advanced module optimization
76% basic chunk optimization
77% chunk optimization
78% advanced chunk optimization
79% module and chunk tree optimization
80% chunk modules optimization
81% advanced chunk modules optimization
82% module reviving
83% module order optimization
84% module id optimization
85% chunk reviving
86% chunk order optimization
87% chunk id optimization
88% hashing
89% module assets processing
90% chunk assets processing
91% additional chunk assets processing
92% recording
94% asset optimization
95% emitting
100%
```



本地修改代码之后, 无需重新执行`npm run v3:start`, 保存代码之后刷新浏览器即可更新预览界面

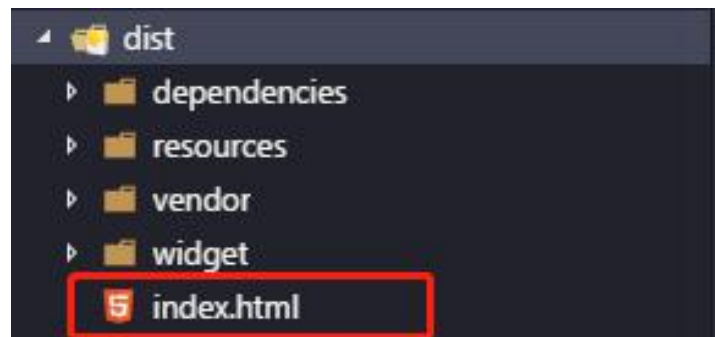
本地编译预览

执行 `npm run v3:package`, `dist`文件夹下输出的`index.html`可以直接预览效果

```
E:\File\2018开发需求\控件自定义开发\demo\v3dropdown>npm run v3:package
> v3dropdown@1.0.0 v3:package E:\File\2018开发需求\控件自定义开发\demo\v3dropdown
> v3 webpack

正在使用webpack打包控件,请稍候...
webpack打包控件完成!

E:\File\2018开发需求\控件自定义开发\demo\v3dropdown>
```



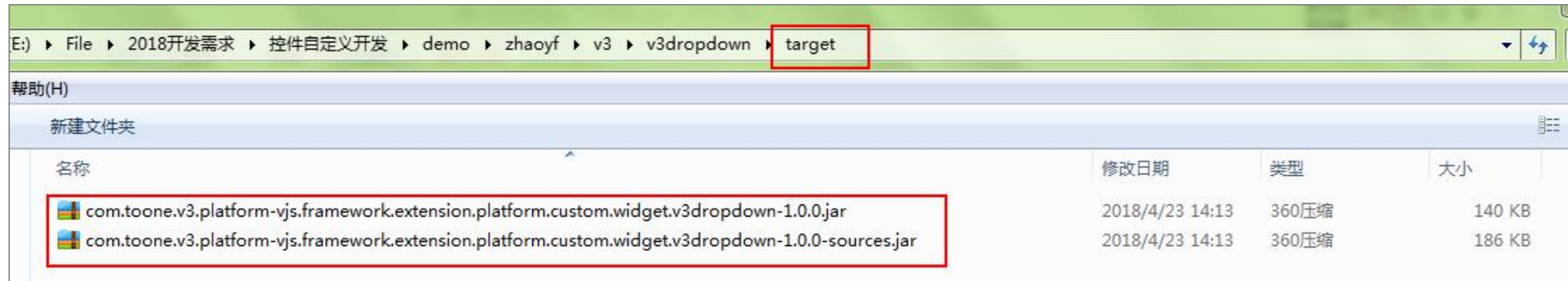
▶ 打包生成vstore构件

执行 `npm run v3:pack`

```
E:\File\2018开发需求\控件自定义开发\demo\v3dropdown>npm run v3:pack
> v3dropdown@1.0.0 v3:pack E:\File\2018开发需求\控件自定义开发\demo\v3dropdown
> v3 pack

正在使用webpack打包控件,请稍候...
webpack打包控件完成!
正在生成UStore构件,请稍候...
UStore构件生成完成,总共:189985字节!
UStore构件生成成功!

E:\File\2018开发需求\控件自定义开发\demo\v3dropdown>
```



名称	修改日期	类型	大小
com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.v3dropdown-1.0.0.jar	2018/4/23 14:13	360压缩	140 KB
com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.v3dropdown-1.0.0-sources.jar	2018/4/23 14:13	360压缩	186 KB

部署到vstore仓库

执行 `npm run v3:publish`

```
D:\work\v平台二次开发 vui\测试\v3dropdown>npm run v3:publish
> v3dropdown@1.0.0 v3:publish D:\work\v平台二次开发 vui\测试\v3dropdown
> v3 publish

正在使用webpack打包控件, 请稍候...
webpack打包控件完成!
正在将V3平台插件部署到VStore仓库, 请稍候...
VStore构件生成完成, 总共: 91018字节!
开始部署插件到VStore, 请稍候...
? VStore账号: zhaoyf
? Vstore密码: [hidden]
部署插件到VStore仓库完成!
```

第一次部署到vstore仓库时, 会要求输入Vstore账号和密码。

库分类	私库	构件库类型	开发库	库名	二次开发 (DEV)	构件类型	v3drop	构件标识	构件类型	构件名称	分组	产品标识
1								com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.v3dropdown	RuntimeJava	v3dropdown	com.toone.v3.platform	vjs.framework.extension...
								com.toone.v3.platform-vjs.framework.extension.platform.custom.widget.v3dropdown				

▶ 安装到指定服务

执行 `npm run v3:apply`

```
D:\work\v平台二次开发 vui\测试\v3dropdown>npm run v3:apply
> v3dropdown@1.0.0 v3:apply D:\work\v平台二次开发 vui\测试\v3dropdown
> v3 apply
...
webpack打包控件完成!
正在将V3平台插件安装到执行系统中,请稍候...
VStore构件生成完成, 总共: 91017字节!
? VStore账号: zhaoyf
? Vstore密码: [hidden]
? 执行系统地址: localhost:8888
获取执行系统VJS依赖信息...
V3平台插件安装到执行系统完成!
```

4710 Active	50 radiov3 (1.0.0.SNAPSHOT)
4713 Active	100 扩展自定义控件 (1.0.0)
4714 Active	50 v3dropdown (1.0.0.SNAPSHOT)
5938 Active	50 v3checkboxgroup (1.0.0.SNAPSHOT)
7176 Active	100 移动端插件demo (1.0.0)
8418 Active	100 布局 (1.0.0)
8419 Active	50 v3circleprogress (1.0.4.SNAPSHOT)
8420 Active	50 v3progress (1.0.0.SNAPSHOT)

第一次安装到服务时, 会要求输入Vstore账号密码, 和要安装到的指定的执行系统的地址。

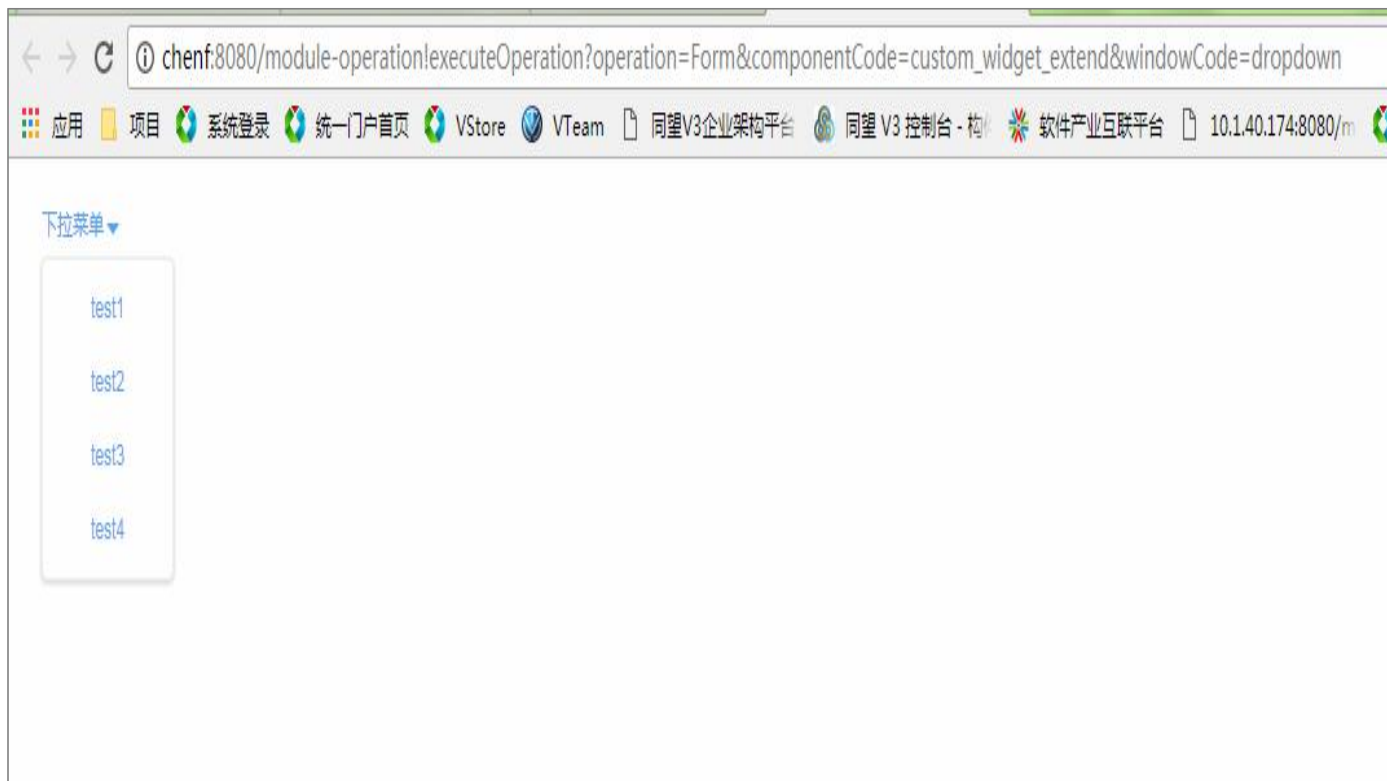
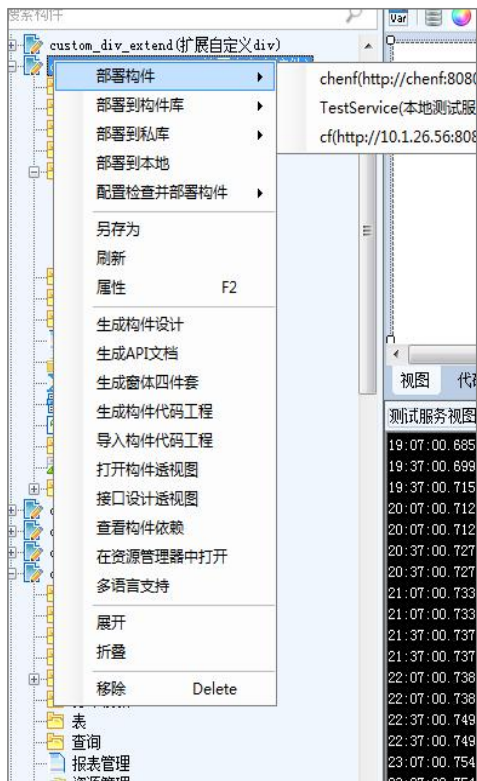
▶ 控件集成使用

以v3dropdown为例子，先在指定执行系统中安装v3dropdown控件，开发系统中创建一个自定义窗体，打开div编辑器

```
template  script  style  theme.less  var.less
1  <div style="margin:20px;">
2      <v3dropdown :title='btnTitle' :item-source='dataSource' item-value='value' item-text='title'
   @on-click='onClick'></v3dropdown>
3  </div>
```

▶ 控件集成使用

部署窗体预览效果



THANK YOU

