## VUE 表单翻译器开发文档

#### 设计理念

表单设计器中通过拖拉拽生成的页面结构及组件，通过在翻译器中动态生成组件的形式，一一对应于vue表单翻译器的组件类型，属性等。

#### 控件类型

主要分为三种：**容器类组件**，**基础类组件**，**复合组件**；

1. 容器类： section、section-cols、toolbar
2. 基础类：textbox（文本框）、number（数字框）、textarea（文本域框）、dropdown（下拉框）、checkbox（复选框）、radio（单选框）、choosebox、datetime（时间）、label（标签）、htmlfield（html）、image（图片）、hypelink（超链接）；
3. 复合组件：datagrid（表格）、tab（tab切换）、file（上传附件）；

上述的每个组件都要对应封装好的vue控件，只需要根据服务端传过来的控件类型分别展示就行；

##### 表单的数据模型（smartDs）

this.store.getters.nebulogyForm.nebulogyForm是对smartDs的引用，在组件内部和各组件之间都是通过操作来this.store.getters.nebulogyForm.nebulogyForm来实现表单数据响应的。

#### 公共的属性与方法

全局公共方法：

this.getAccount() // 获取当前登录用户信息

this.getView() // 获取当前表单视图

this.getUrlData(key) // 获取url地址里面的query（参数），key代表想要获取的参数名称；eg：this.getUrlData(destInstId)

全局变量：

this.downloadUrl //获取文件下载地址

全局方法

// 设置表单基础配置

Vue.prototype.setBaseConfig = function(config) {

dataMap.baseConfig = config

}

// 获取表单基础配置

Vue.prototype.getBaseConfig = () => {

return dataMap.baseConfig || {}

}

Vue.prototype.setRequestData = function(config) {

dataMap.baseConfig = config

}

Vue.prototype.getBaseConfig = () => {

return dataMap.dataFields || {}

}

##### 4.1 公共属性

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 容器类 | 基础类 | 复合组件 |
| value | 无 | 有 | 有 |

##### 4.2 公共方法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法名 | 作用 | 容器类 | 基础类 | 复合组件 |
| show | 显示控件 | 有 | 有 | 有 |
| hide | 隐藏控件 | 有 | 有 | 有 |
| getValue | 获取数据 | 无 | 有 | 有 |
| setValue | 设置数据 | 无 | 有 | 有 |
| setDisabled | 控件禁用启用 | 无 | 有 | 有 |
| setProperty | 修改控件属性值 | 有 | 有 | 有 |
| getFieldLabelValue | 获取组件的fieldLabel | 无 | 有 | 有 |

Eg: this.$refs.componentName.show();

setProperty(key, val); // key:要修改的属性名，val: 期望修改的值

以下从表单的最外层form层开始解析表单：

说明：

1. 后面例子中用到的componentName是指组件的name属性的值，每个组件在渲染的时候都会有个ref属性，取值为改name对应的值
2. 监听的事件是指在表单设计器中可以给组件设置监听，然后在相应的触发条件下触发
3. this.$refs是vue框架提供的api，组件在定义了ref属性的前提下，都可以通过this.$refs获取到该组件，ref 被用来给元素或子组件注册引用信息。引用信息将会注册在父组件的 $refs对象上。如果在普通的 DOM 元素上使用，引用指向的就是 DOM 元素；如果用在子组件上，引用就指向组件实例。

#### 组件的属性与方法

##### 5.1 容器类

###### 5.1.1 Form

1. 表单监听的事件：

a: beforeInit //表单初始化之前，该方法返回true才会执行接下来的init事件

b: init //表单初始化事件，在页面数据从服务端加载完成时触发

c: beforeSubmit // 提交表单前事件，该方法返回true才会执行接下来的submit事件

d: submit // 表单提交成功触发该事件

e:beforeAction // 表单头部action按钮操作前事件，改方法返回true才会执行接下来的action事件

g:action // action 操作成功之后触发该事件

f:smartDsChange // smartDs发生改变是会触发该事件，返回值：bindPath（组件绑定smartDs的路径）,oldValue（上一次的值）,newValue（当前的值）

h:beforeDraft // 保存草稿之前触发事件，该方法返回true才会执行接下来的draft事件

i:draft // 草稿保存成功之后触发该事件

1. 外部可调用方法:

a:getCirculationRecord() // 获取表单的流程记录信息

B :showOpinionCollection() // 显示意见收集弹窗

C: toggleCirculationRecord(bool) // 是否隐藏流转记录模块，默认显示，bool为布尔类型true显示，false隐藏，参数为空是也是隐藏

###### 5.1.2 Section

1、组件监听事件：无

###### Toolbar

1、组件监听事件：无

###### 5.1.4 Window

1、组件监听事件：暂无

##### 5.2 基础类

###### 5.2.1 Textbox

1. 组件监听事件：

a: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

b:focus // 获取焦点时触发，返回值：value（组件当前的值）

c:blur // 失去焦点时触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val：string（字符串类型）或者number（数值类型），

eg: this.$refs.componentName.setValue(‘张三’)

b:getValue() // 获取组件的值，得到组件当前的值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.2 Number

1、组件监听事件：

a: change// 组件的值发生改变的时候触发，返回值：value（组件当前的值）

b:focus// 获取焦点时触发，返回值：value（组件当前的值）

c:blur// 失去焦点时触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val：number（数值类型），

eg: this.$refs.componentName.setValue(10)

b:getValue() // 获取组件的值，得到组件当前的值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.3 Textarea

1. 组件监听事件：

a: change// 组件的值发生改变的时候触发，返回值：value（组件当前的值）

b:focus// 获取焦点时触发，返回值：value（组件当前的值）

c:blur// 失去焦点时触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val：string（字符串类型）或者number（数值类型），

eg: this.$refs.componentName.setValue(10)

b:getValue() // 获取组件的值，得到组件当前的值,如果该组件用来作为填写审批意见时，获取该组件的值需要使用getText();因为审批意见没有做双向绑定，使用getValue反而获取不到值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.4 Dropdown

1. 组件监听事件：

a: change// 组件的值发生改变的时候触发，

返回值：value（组件当前保存的值），displayValue（组件显示的值）

1. 外部可调用的组件方法：

a: setValue(data) //给组件设置值，data：object（对象类型），

eg:

//label 用来显示的值，value：实际保存的值；

const data = {label: ‘张三’, value: ’23232432’} ;

this.$refs.componentName.setValue(data) // 下拉框单选时的赋值方式

eg:

//label 用来显示的值，value：实际保存的值；

const data = [{label: ‘张三’, value: ’23232432’},{label: 李四, value: ’23232432’}];

this.$refs.componentName.setValue(data) // 下拉框多选时的赋值方式

b: getValue() // 获取组件的值，得到组件当前所选的值

c: setData(arr, field) // 外部设置下拉框的值, arr :Array(数组类型)，field：Object(对象类型)；固定格式{ displayField :’, valueField : ‘}其中displayField对应的值就是arr中要显示的key，valueField对应的值就是arr中要保存的key。

eg：

const field = { displayField :’label’, valueField : ‘value’}；

const arr= [{label:’张三’，value: ‘12123’},{ label:’李四’， value: ‘12123’}]；

this.$refs.componentName.setData(arr, field);

3、该组件中用来保存下拉框里面数据的字段是selectionOptions

###### 5.2.5 Checkbox

1. 组件监听事件：

a: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setCheck(key, bool) // 给组件设置是否选中， key :复选框的index，bool: Boolean（布尔类型）；true:选中，false:未选中

eg: this.$refs.componentName.setCheck(0,true);

b: setValue(key, bool) // 设置组件选中的值，和 setCheck方法使用一致;

c: getValue() // 获取组件的值

d: disable(key)// 设置组件指定索引的复选框不可编辑状态，key：复选框的index

e: enable(key) // 设置组件指定索引的复选框为可编辑状态，key：复选框的index

f: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

g: setToolTip(index, bool, content) // index:number类型，指定第几个复选框使用提示语；bool:Boolean类型,是否启用提示语；content:string类型，提示语内容

###### 5.2.6 Radio

1. 组件监听事件

a: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val：string（字符串类型）,对应的是radio组件的中options中的value值，

eg: this.$refs.componentName.setValue(1)

b: getValue() // 获取组件的值，得到组件当前的值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

d: setToolTip(index, bool, content) // index:number类型，指定第几个复选框使用提示语；bool:Boolean类型,是否启用提示语；content:string类型，提示语内容

###### 5.2.7 Datetime

1. 组件监听事件

a: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

b: foucs // 获取焦点时触发，返回值：value（组件当前的值）

c: blur// 失去焦点时触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) // 给组件设置值，val：为日期值如：2019-7-23

b: getValue() // 获取组件的值

c: setMinValue(val) // 设置组件选日期的最小值

d: setMaxValue(val)// 设置组件选日期的最大值

f: clearPickerOptions() // 清除组件的选最小和最大日期的限制

g: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.8 Label

1. 组件监听事件

a: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

1. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val: string（字符串类型）或者number（数值类型）

b: getValue() // 获取组件值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.9 Htmlfield

1、组件监听事件：无

a: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2. 10 Image

1、组件监听事件：无

###### 5.2.11 Hypelink

1. 组件监听事件 :无
2. 外部可调用的组件方法：

a: setValue(val) //给组件设置值，val: 要跳转的网站完整地址（需要带http或者https）

b: getValue() // 获取组件值

c: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

###### 5.2.12 Choosebox

**（**用来做弹出框，弹出框的内容目前支持三种弹框：选人，选组织，选表单**）**

1. 组件监听事件

a: triggerClick //组件点击时触发，该事件需要设置以下参数才能显示弹框：

eg:

//选人：'personCom'，选组织：’ organizationCom’,选表单: ‘customDialog’

this.$refs.assistantName.activeName = 'personCom';

// 设置弹框的title

this.$refs.assistantName.templateTitle = '选择人员'

this.$refs.assistantName.templateProps= {

valueField: 'empId',//保存的字段名

displayField: 'employeeName'// 显示的字段名称

}

b: change // 组件的值发生改变的时候触发，返回值：value（组件当前的值）

c: clear // 清空控件选择的值

1. 组件外部可调用的方法：

a: setValue(obj) //给组件设置值

注意、：需要设置相关值的\_+component名字（例如：\_default:1）

eg:

单选: data{label: ‘显示值’, value: ‘保存的值’}

多选: data{

label: ‘显示值1,显示值2’,

value: [

{displayField: ‘显示值1’, valueField: ‘保存的值’},

{displayField: ‘显示值2’, valueField: ‘保存的值’}

],

propertyAttrField : {

displayField: templateProps.displayField,

valueField: templateProps.valueField

}

},



displayField对应的是triggerClick事件设置的templateProps中的displayField的值，valueField对应的是triggerClick事件设置的templateProps中的valueField的值,propertyAttrField对应的是组件的显示和保存的key，这个key要和triggerClick事件设置的templateProps的displayField和valueField保持一致

；

b: getValue() // 获取组件的值

c: setData(arr, field, propertyAttrField) // 如果choosebox的前半部分是下拉框的情况下可以使用该方法给下拉框设置下拉选项,arr：下拉选项的数据，是个数组，field：对应的是arr数据中的显示的key和保存的key，propertyAttrField对应的是组件的显示和保存的key，这个key要和triggerClick事件设置的templateProps的displayField和valueField保持一致

eg:

const arr = [{label: ‘张三’，value: ‘123123’},{label: ‘李四，value: ‘123123’}]；

const field = {displayField:’label’,valueField: ‘value’};

const propertyAttrField = {

displayField: templateProps.displayField,

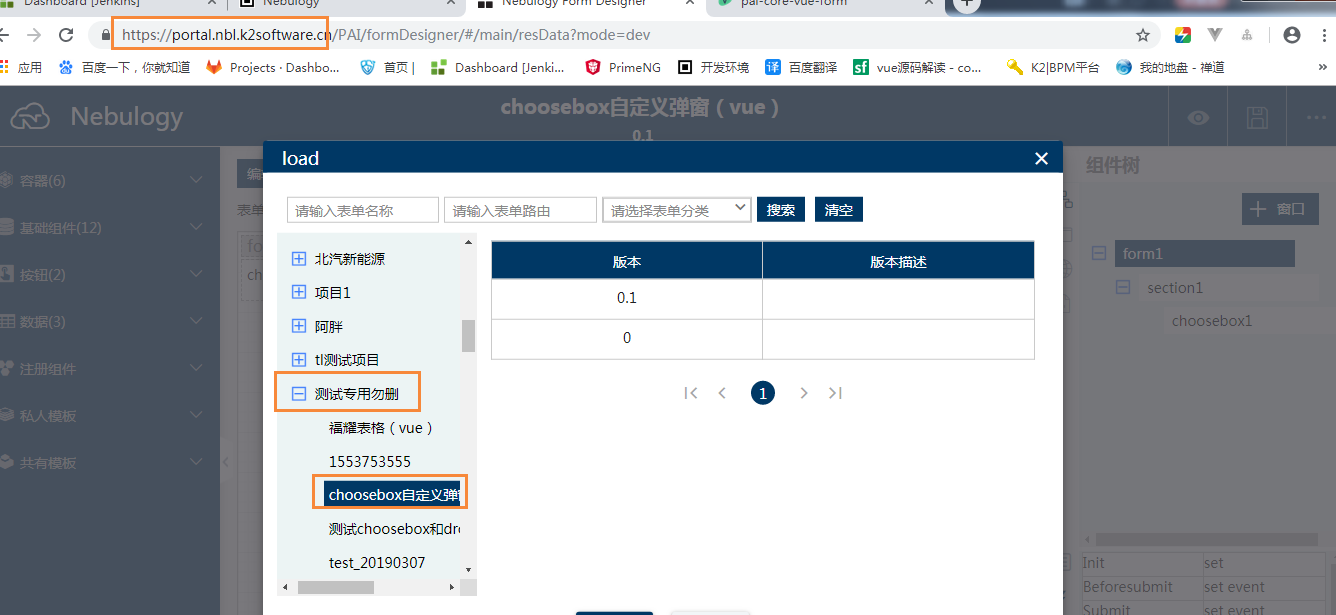
valueField: templateProps.valueField

}

this.$refs.componentName.setData(arr, field, propertyAttrField)

d: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

e: setEnableCustomDialog() // 设置choosebox是否启用自定义弹窗，自定义弹窗可通过动态组件的方式添加（也可以使用组件自带的属性IscustomDialog来设置），详情可参考阿里环境该表单：



1. 该组件的前半部分可以是input输入框也可以是下拉框，默认是input输入框，如果想要改为下拉框可以在表单设计器那边设置该组件的**isCombo**属性为true就能实现
2. hideModal():

设置弹窗遮罩的显示关闭如：

true 开启（默认），false 关闭

this.$refs.componentName.hideModal(false)

1. 该组件中用来保存下拉框里面数据的字段是property.dataSource.options

注意：和dropdown中用来保存下拉框数据的字段是有区别的。



##### 5.3 复合类

###### 5.3.1 Datagrid

1. 组件监听事件

a: itemClick // 某一行点击时触发(回调参数:{row, event,cell})

b: cellClick // 单元格点击时触发(回调参数:{row, column,cell,event})

c: DataChanged // 表格数据发生变化时触发(回调参数: data(当前表格的数据))

d: Beforeremoverecord // 删除某一行数据之前触发

e: Removerecord //删除某一行数据触发(回调参数：data(删除后的表格数据))

f: Beforeaddrecord // 增加记录之前触发

g: Addrecord // 增加记录触发

h: Beforeeditrecord // 编辑记录之前触发(回调参数:{row, column,cell,event})

i: editrecord // 编辑记录焦点离开触发 (回调参数:data(包含当前编辑行和列的信息以及当前单元格的值))

1. 组件外部可调用的方法：

a: setValue(val) // 给组件设置值； val: Array(数组类型);

eg:/\*

\*先定义一个数组，该数组里面的元素为对象形式，该对象以所需要设置的表格的业务模型对应的值为key，

\*value则是需要设置的值

\*\*/

let defaultValue1 = [

{

name: '飞机票',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '火车票',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '船票',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '出租车票',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name:'公交车票',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '住宿费',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '订/退票费',

number: '0',

count: '0.00',

selfPayment: '0.00',

desc: ''

},

{

name: '打包费',

number: '0',

count: '0.00',

selfPayment: '0.00',

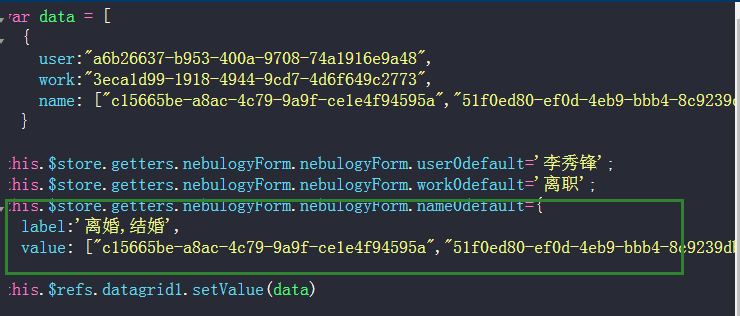
desc: ''

}

]

// 如果其中一列的editorType为hyperlink(超链接)的话，该列对应的字段的数据类型应为object，设置值的格式应为这样{label:’http://xxx.com(要跳转的地址)’,value:’要显示在表格中的值’}

// 给combo，multiCombo，choosebox赋值需要额外再加上该列的dataIndex+i+default作为key，i是行号，值就是你要赋的，然后把那这个字段添加到this.$store.getters.nebulogyForm.nebulogyForm上，multiCombo对应的dataIndex+i+default是个对象{label：展示值，value：对应的数组的值}



// 有两种方式可以给表格设置值

// 方法一，直接修改表格对应绑定的业务模型（smartDs）；

This.$store.getters.nebulogyForm.nebulogyForm.datagrid2List=defaultValue1 ;

// 方法二，翻译的表格组件有对应的setValue方法，可以直接调用

// this.$refs.componentName.setValue(defaultValue1)

}

b: getValue() // 获取组件当前的值

c: setPropertyAttr(property) //设置表格的propertyAttr属性，property：object（对象类型）; eg：const property = {editable: false};

d: setColumn(columns) // 设置表格的column,cloumns:Array（数组类型）,参数中必须存在的属性dataIndex；

eg:const columns = [{dataIndex: ‘对应表单设计器中表格的dataIndex’，otherProperty: ‘想要设置的值’}]

e: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

f: getCurSelection() // 表单开启复选框的情况下，可以通过该方法获取当前选中的值

###### 5.3.2 Tab

1、组件监听事件

a:beforetabchange (tab切换的之前触发，如果该事件返回false则阻止切换，仍然在当前tab)

b:tabchange(tab切换成功时触发，返回当前tabItem和上一次的tabItem的name)

###### File

1、组件外部可调用的方法：

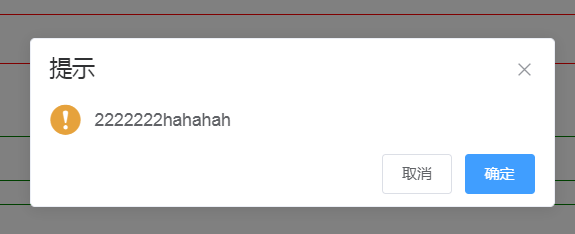
a: setDisabled(disabled) // disabled: Boolean(boolean类型)，true:禁用；false:启用

##### 表单头部

##### 自定按钮的添加

在form的buttonConfig里面定义header头部提交按钮提示信息自定义：

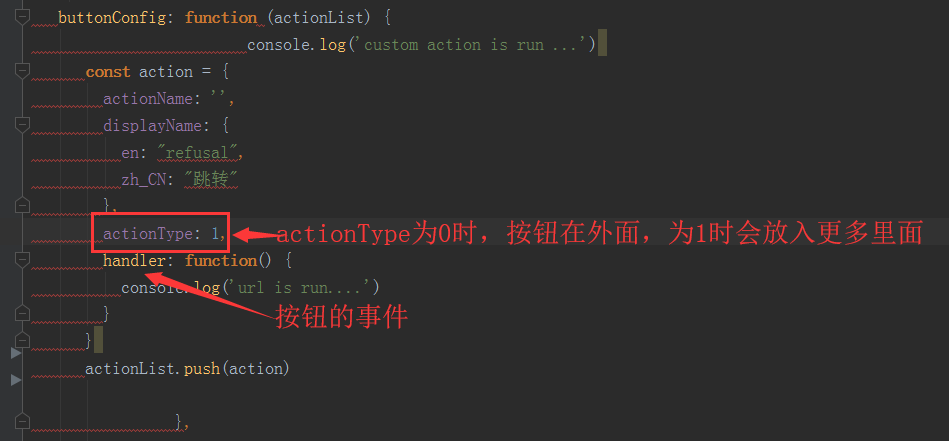
this.$refs.header.setActionTip('自定义的提示信息')



this.$refs.header.setSuccessTip('这是提交成功的提示语')



如图：（自定义按钮）

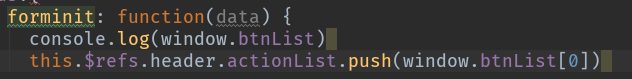


表单自带btnconfig事件，给参数actionList push数据就可以自定义审批按钮。

**【自定义按钮控制显示隐藏方法】**

1. **在buttonConfig中清空header组件中所有的按钮，如图：**
2. **actionList中有所有的按钮，如图：**
3. **将所有按钮挂载到window的一个变量上面，以便在其他方法中能访问的到，如图：**
4. **在forminit里面，将挂载到window上面的按钮数组，根据项目需要放到header的actionList组件中，如图：**

****

****

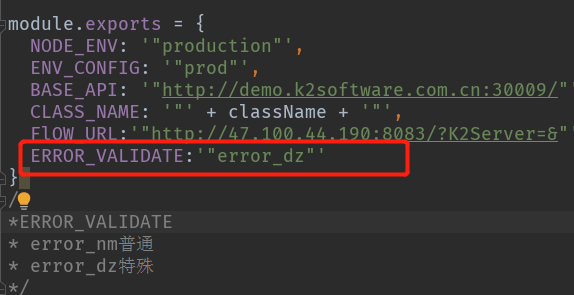
##### 8、使用意见收集功能（待澄清）

先通过添加自定义按钮（待澄清按钮）,该按钮绑定的click事件：

this.$refs.form1.showOpinionCollection() // 显示待澄清弹窗

##### 9、验证信息自定义提示信息使用方式

设置验证使用方式：



/\*

\*ERROR\_VALIDATE

\* error\_nm普通

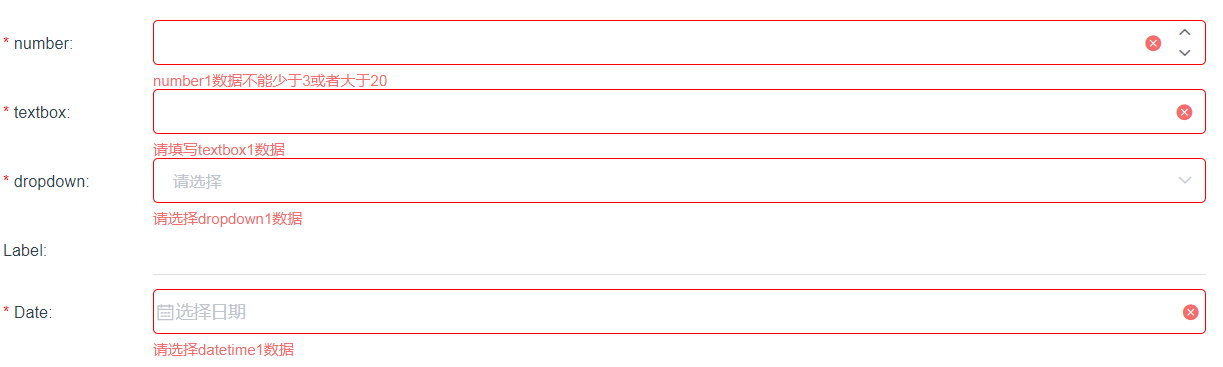
\* error\_dz特殊

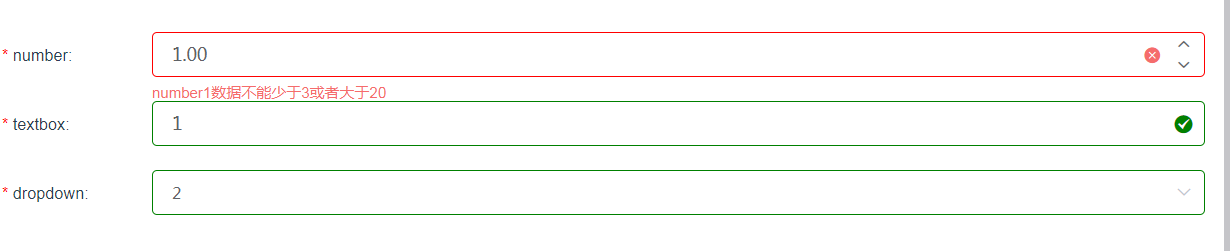
\*/

根据表单的v-modal的path路径设定相关信息,如下



效果：





##### 可以被重写的方法及示例

**表单的提交、草稿、各个action操作都可以被重写，因为这些按钮都在header组件里面，所以这些方法属于header组件的，重写的时候先找到header组件：**

重写action操作

this.$refs.header.onAction = function(action, param, closeWindow = true){};

重写提交

this.$refs.header.onSubmit=function(){}

重写保存草稿

this.$refs.header.onDraft=function(){}

重写获取action

this.$refs.header.getActionList=function(){

//这里需要返回一个promise，返回的数据格式需要保持和平台的action接口的数据格式一致

}

**// datagrid实现后端分页的时候也需要重写datagrid组件对应的页码发生改变和页数发生改变的事件**

// 分页pageSize 发生变化时

this.$refs.relProcessList.handleSizeChange = (val) => {

this.$refs.relProcessList.pageSize = val;

this.$refs.relProcessList.pageStart = 0;

this.$refs.relProcessList.pageEnd = val;

var paramData = window.getScreenData();

var pagination = {

pageIndex: 0,

pageSize: val

};

window.getProcessByCondition({

data: paramData,

pagination: pagination

});

};

// 页码发生变化时

this.$refs.relProcessList.handleCurrentChange = (val) => {

this.$refs.relProcessList.currentPage = val;

var paramData = window.getScreenData();

var pagination = {

pageIndex: val - 1,

pageSize: this.$refs.relProcessList.pageSize

};

// 这是请求数据的方法，可以自己定义，这只是个示例，不是固有的方法

window.getProcessByCondition({

data: paramData,

pagination: pagination

});

};

**//替换url的方法**

替换获取流转记录的url

this.setReplaceUrl('processLog', 要替换的url)

替换获取actionList的ur

this.setReplaceUrl(‘actionList’, 要替换的url)