

1.树组件的用法

1.1.示例

```
<v-tree url="/item/category/list"
  :isEdit="true"
  @handleAdd="handleAdd"
  @handleEdit="handleEdit"
  @handleDelete="handleDelete"
  @handleClick="handleClick"
/>
```

效果：



1.2.属性列表

属性名称	说明	数据类型	默认值
url	用来加载数据的地址，即延迟加载	String	-
isEdit	是否开启树的编辑功能	boolean	false
treeData	整颗树数据，这样就不用远程加载了	Array	-

这里推荐使用url进行延迟加载，每当点击父节点时，就会发起请求，根据父节点id查询子节点信息。

当有treeData属性时，就不会触发url加载

远程请求返回的结果格式：

```
[
  {
    "id": 74,
    "name": "手机",
    "parentId": 0,
    "isParent": true,
    "sort": 2
  },
  {
    "id": 75,
    "name": "家用电器",
    "parentId": 0,
    "isParent": true,
    "sort": 3
  }
]
```

1.3.事件：

事件名称	说明	回调参数
handleAdd	新增节点时触发，isEdit为true时有效	新增节点node对象，包含属性：name、parentId和sort
handleEdit	当某个节点被编辑后触发，isEdit为true时有效	被编辑节点的id和name
handleDelete	当删除节点时触发，isEdit为true时有效	被删除节点的id
handleClick	点击某节点时触发	被点击节点的node对象,包含全部信息

1.4.完整node的信息

回调函数中返回完整的node节点会包含以下数据：

```
{
  "id": 76, // 节点id
  "name": "手机", // 节点名称
  "parentId": 75, // 父节点id
  "isParent": false, // 是否是父节点
  "sort": 1, // 顺序
  "path": ["手机", "手机通讯", "手机"] // 所有父节点的名称数组
}
```

2.级联选择组件

2.1.示例：

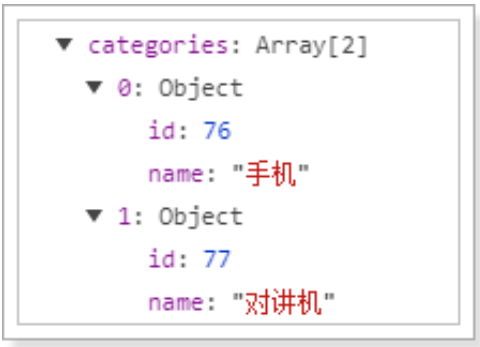
```
<v-cascader
  url="/item/category/list"
  multiple required
  v-model="brand.categories"
  label="请选择商品分类"/>
```

2.2.效果：



2.3.结果值

v-model 绑定的数据结果：



2.4.属性列表：

属性名称	说明	数据类型	默认值
url	延迟加载数据的地址 [{id,name},{}]	String	无
itemText	每个选项中用来显示的字段名称	String	name
itemValue	每个选项中用来作为值的字段名称	String	id
children	子选项数组在父选项中的字段名称	String	children
multiple	是否允许多选	boolean	false
showAllLevels	是否将级联的每级选项都作为结果展示。当multiple值为true时，这个值无效，即只会显示最后一级选项的结果	boolean	false
required	是否是必填项，如果是，会在文本提示后加*	boolean	false
rules	自定义校验规则	Array	无
value	选择框的结果，可以通过v-model进行双向绑定	Array	无
label	提示用户的文字说明	String	无

2.5.说明：

无论是单选还是多选，value的结果格式始终是一个数组。单选时数组的长度始终为1。

数组中的对象结构为：

```
{
  {itemValue}:'', // 属性名取决于itemValue的值，默认是id
  {itemText}:''// 属性名取决于itemText的值，默认是name
}
```

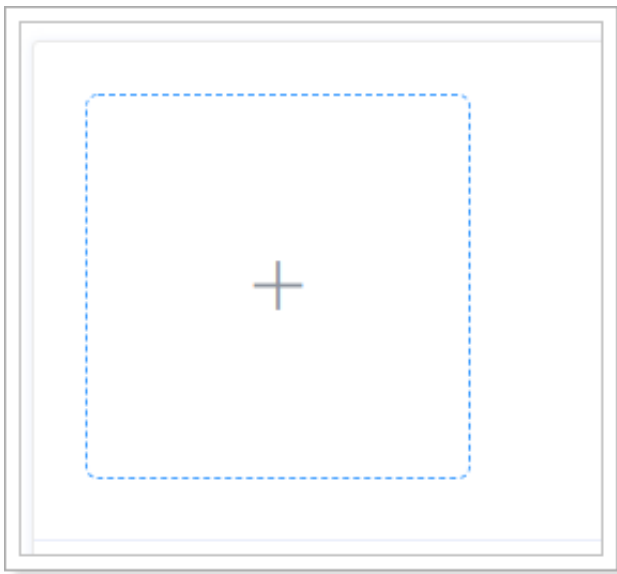
3.文件上传组件

3.1.示例：

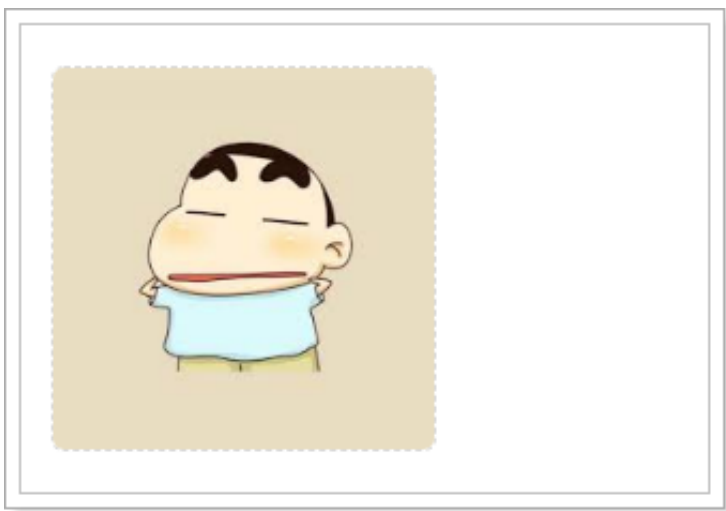
3.1.1.单图片上传：

```
<v-upload  
  v-model="brand.image"  
  url="/item/upload"  
  :multiple="false"  
  :pic-width="150"  
  :pic-height="150"  
>
```

上传前：



上传后：



3.1.2.多图片上传

示例：

```
<v-upload
  v-model="brand.image"
  url="/item/upload"
/>
```

效果：



3.2.属性列表：

属性名	说明	数据类型	默认值
url	上传文件的目标路径	String	无
value	上传成功的返回结果	单图片上传是String。多图片上传是String数组	无
multiple	是否运行多图片上传	Boolean	true
picWidth	单图片上传后的预览宽度	Number	150
picHeight	单图片上传后的预览高度	Number	150

3.3.说明

可以通过v-model双向绑定，来获取图片上传的结果：

- 单图片上传时，value值是一个图片地址
- 多图片上传时，value值是一个图片地址数组
- 文件上传的参数名是：file
- 文件上传的返回值，就是图片的url路径

4.自定义富文本编辑器

4.1.示例：

```
<v-editor v-model="goods.spuDetail.description" upload-url="/upload/image"
fileName="file"/>
```

4.2.效果：



4.3.属性说明

属性列表：

属性名	说明	数据类型	默认值
value	编辑器的输出结果，可以用v-model双向绑定	String	无
upload-url	上传按钮对应的图片上传地址，以项目全局的url配置为前缀	String	无
file-name	上传文件的参数名	String	file
maxUploadSize	上传文件的大小限制，单位byte	Number	500kb

备注：

默认支持的图片类型：jpg/png/jpeg/gif