

- [Cascader 组件接口文档](#)
  - [CascaderProps 接口](#)
  - [CascaderOption 接口](#)
  - [Events](#)
  - [示例用法](#)
  - [组件使用示例](#)

# Cascader 组件接口文档

## CascaderProps 接口

描述了 Cascader 组件的属性配置。

参数名	描述	类型	默认值
setProps	配置项，用于各种功能的设置	Props	-
options	数据，每个选项的配置信息	CascaderOption[]	-
clearable	是否显示清空按钮	boolean	false
modelValue	v-model绑定输入框的值	string	-
separator	分隔符	string	-
inputPlaceholder	输入框的占位符	string	请选择
filterable	是否开启搜索功能	boolean	false
panelHeight	自定义面板高度, 需要带单位,min-height: 204px	string	'204px'
lazy	是否动态加载子节点，需与	boolean	false

参数名	描述	类型	默认值
lazyLoad 方法 结合使用			
lazyLoad	加载动态数据的方法，仅在 lazy 为 true 时有效	Function: (cureent:CascaderOption) =>Promise<CascaderOption[]>	-
lazyCallBack	懒加载回调函数，仅在 lazyLoad存在且有效时有效	Fuction: (cureent:CascaderOption, result:CascaderOption[]) =>void	-

## CascaderOption 接口

描述了 Cascader 组件选项的配置。

参数名	描述	类型	默认值
value	选项的值	string  number	-
label	选项显示的文本	string	-
children	子选项数组，用于多级联动	CascaderOption[]	-
disabled	是否禁用该选项	boolean	false
leaf	是否是叶子节点	boolean	false
level	层级	number	-
selected	是否选中	boolean	false
checked	是否勾选	boolean	false

## Events

Cascader 组件触发以下事件，可以通过 v-on 或 @ 监听：

事件名	描述	返回值	参数
finally	在输入框值最终确认后触发	{ newVal }	-
change	在输入框值发生变化时触发	{ newVal }	-

事件名	描述	返回值	参数
close	在面板关闭时触发	-	-

## 示例用法

```
import {CascaderOption } from './main'

const options:CascaderOption[] = [
  {
    value: '1',
    label: 'Option 1',
    children: [
      { value: '1-1', label: 'Option 1-1' },
      { value: '1-2', label: 'Option 1-2' }
    ]
  },
  {
    value: '2',
    label: 'Option 2',
    children: [
      { value: '2-1', label: 'Option 2-1' },
      { value: '2-2', label: 'Option 2-2' }
    ]
  }
]

// 示例使用组件的事件监听
const change = ({ newVal }) => {
  console.log('change:', newVal)
}
const finallyx = ({ newVal }) => {
  console.log('finally:', newVal)
}

const close = () => {
  console.log('close')
}
const lazyLoad = async ({ value, level }: CascaderOption):
Promise<CascaderOption[]> => {
  return new Promise<CascaderOption[]>(async (resolve, reject) => {
    if (value && level) {
      const res = (await getData(value, level + 1)).data
      if (res.code == -1) {
        reject(res)
      }
      if (level == 4) {
        res.data.forEach((item) => {
          item.leaf = true
        })
      }
      let children = res.data
    }
  })
}
```

```
        resolve(children)
      }
    })
  }

  const lazyCallBack = (current: CascaderOption, result: CascaderOption[]) => {
    current.children = result
  }
```

## 组件使用示例

```
<Cascader
  :options="options"
  separator="-"
  :clearable="true"
  v-model="inputValue"
  panel-height="204px"
  @change="change"
  @finally="finallyx"
  @close="close"
  :lazy="true"
  :lazy-load="lazyLoad"
  :lazy-call-back="lazyCallBack"
/>
```