```
<!--
 * @Author: mrslimslim 2451319596@qq.com
 * @Date: 2023-04-11 23:35:15
 * @LastEditors: mrslimslim 2451319596@qq.com
 * @LastEditTime: 2023-04-13 00:24:45
 * @FilePath: \gpt4-pdf-chatbot-langchain\tdesing.md
 * @Description: 这是默认设置,请设置`customMade`, 打开koroFileHeader查看配置 进行设置:
https://github.com/OBKoro1/koro1FileHeader/wiki/%E9%85%8D%E7%BD%AE
-->
```

以下是tdesign(vue3)的文档

安装
使用 npm 安装
推荐使用 npm 方式进行开发

npm i tdesign-vue-next
通过 浏览器引入 安装
目前可以通过 unpkg.com/tdesign-vue-next 获取到最新版本的资源，在页面上引入 js 和 css 文件即可开始使用。

```
<script src="https://unpkg.com/vue@next"></script>
<link rel="stylesheet" href="https://unpkg.com/tdesign-vue-next/dist/tdesign.min.css" />
<script src="https://unpkg.com/tdesign-vue-next/dist/tdesign.min.js"></script>
...
<script>
        Vue.createApp({}).use(TDesign).mount('#app');
</script>
```
npm package 中提供了多种构建产物，可以阅读 这里 了解不同目录下产物的差别。

使用
基础使用
```
import { createApp } from 'vue';
import TDesign from 'tdesign-vue-next';
import App from './app.vue';

// 引入组件库全局样式资源
import 'tdesign-vue-next/es/style/index.css';

const app = createApp(App);
app.use(TDesign);
npm package 中提供了多种构建产物，可以阅读 这里 了解不同目录下产物的差别。

按需引入使用
对产物大小有要求的场景，可以按需引入组件， 借助 Webpack 或 Rollup 等支持 tree-shaking 特性的构建工具达到按需引入的使用效果。

import { createApp } from 'vue';
import { Button as TButton } from 'tdesign-vue-next';
import App from './app.vue';

// 引入组件库全局样式资源
import 'tdesign-vue-next/es/style/index.css';

const app = createApp(App);
app.use(TButton);
```
通过插件按需引用
除此之外，也可以使用 unplugin-vue-components 和 unplugin-auto-import 来实现自动导入：

npm install -D unplugin-vue-components unplugin-auto-import
然后在 Webpack 或 Vite 对应的配置文件添加上述插件。

Vite
```
import AutoImport from 'unplugin-auto-import/vite';
import Components from 'unplugin-vue-components/vite';
import { TDesignResolver } from 'unplugin-vue-components/resolvers';
export default {
  plugins: [
    // ...
    AutoImport({
      resolvers: [TDesignResolver({
        library: 'vue-next'
      })],
    }),
    Components({
      resolvers: [TDesignResolver({
        library: 'vue-next'
      })],
    }),
  ],
};
Webpack
const AutoImport = require('unplugin-auto-import/webpack');
const Components = require('unplugin-vue-components/webpack');
const { TDesignResolver } = require('unplugin-vue-components/resolvers');
module.exports = {
  // ...
  plugins: [
    AutoImport({
      resolvers: [TDesignResolver({
        library: 'vue-next'
      })],
    }),
    Components({
      resolvers: [TDesignResolver({
        library: 'vue-next'
      })],
    }),
  ],
};
```

TDesignResolver 支持的配置，可以点击此链接。

编辑器提示
安装注册 TDesign 之后，在开发项目时，可以配合插件在VSCode等主流编辑器中达到提示组件名及API的效果。

推荐安装 volar，并在项目的 tsconfig.json 的 includes 属性中增加node_modules/tdesign-vue-next/global.d.ts，即可实现该效果。

全局特性配置

Pagination 分页

```

<template>
  <t-config-provider :global-config="globalConfig">
    <div class="tdesign-demo-item--locale-provider-base">
      <t-pagination v-model="current" :total="36" show-jumper :max-page-btn="5" />
    </div>
  </t-config-provider>
</template>

<script setup>
import { ref } from 'vue';

const current = ref(1);
const globalConfig = {
  pagination: {
    itemsPerPage: '{size} / page',
    jumpTo: 'jump to',
    page: '',
    total: 'Total {total} items',
  },
};
```

```
</script>
```

Link 组件文档

```
[
    {
        "名称": "content",
        "类型": "String / Slot / Function",
        "默认值": "-",
        "说明": "链接内容。TS 类型：string | TNode。通用类型定义",
        "必传": "N"
    },
    {
        "名称": "default",
        "类型": "String / Slot / Function",
        "默认值": "-",
        "说明": "链接内容，同 content。TS 类型：string | TNode。通用类型定义",
        "必传": "N"
    },
    {
        "名称": "disabled",
        "类型": "Boolean",
        "默认值": "-",
        "说明": "禁用链接",
        "必传": "N"
    },
    {
        "名称": "hover",
        "类型": "String",
        "默认值": "underline",
        "说明": "链接悬浮态样式，有 文本颜色变化、添加下划线等 2 种方法。可选项：color/underline",
        "必传": "N"
    },
    {
        "名称": "href",
        "类型": "String",
        "默认值": "-",
        "说明": "跳转链接",
        "必传": "N"
    },
    {
        "名称": "prefixIcon",
        "类型": "Slot / Function",
        "默认值": "-",
        "说明": "前置图标。TS 类型：TNode。通用类型定义",
        "必传": "N"
    },
    {
        "名称": "size",
        "类型": "String",
        "默认值": "medium",
        "说明": "尺寸。可选项：small/medium/large。TS 类型：SizeEnum。通用类型定义",
        "必传": "N"
    },
    {
        "名称": "suffixIcon",
        "类型": "Slot / Function",
        "默认值": "-",
        "说明": "后置图标。TS 类型：TNode。通用类型定义",
        "必传": "N"
    },
    {
        "名称": "target",
        "类型": "String",
        "默认值": "-",
        "说明": "跳转方式，如：当前页面打开、新页面打开等，同 HTML 属性 target 含义相同",
        "必传": "N"
    },
```

~~~
    {
        "名称": "theme",
        "类型": "String",
        "默认值": "default",
        "说明": "组件风格，依次为默认色、品牌色、危险色、警告色、成功色。可选项：
default/primary/danger/warning/success",
        "必传": "N"
    },
    {
        "名称": "underline",
        "类型": "Boolean",
        "默认值": "-",
        "说明": "普通状态是否显示链接下划线",
        "必传": "N"
    },
    {
        "名称": "onClick",
        "类型": "Function",
        "默认值": "",
        "说明": "TS 类型：(e: MouseEvent) => void"
    },
    {
        "名称": "点击事件，禁用状态不会触发点击事件",
        "类型": "N"
    },
    {
        "名称": "Link Events"
    },
    {
        "名称": "名称",
        "类型": "参数",
        "默认值": "描述"
    },
    {
        "名称": "click",
        "类型": "(e: MouseEvent)",
        "默认值": "点击事件，禁用状态不会触发点击事件"
    }
]
```
~~~

Link组件用法示例

~~~
```
<template>
  <t-space>
    <t-link theme="primary">跳转链接</t-link>
  </t-space>
</template>
```
~~~

Button组件文档

~~~
```
[
    {
        "名称": "block",
        "类型": "Boolean",
        "默认值": "false",
        "说明": "是否为块级元素",
        "必传": "N",
        "": ""
    },
    {
        "名称": "content",
        "类型": "String / Slot / Function",
~~~

```
        "默认值": "-",
        "说明": "按钮内容。TS 类型：string",
        "必传": "TNode。通用类型定义",
        "": "N"
    },
    {
        "名称": "default",
        "类型": "String / Slot / Function",
        "默认值": "-",
        "说明": "按钮内容。TS 类型：string",
        "必传": "TNode。通用类型定义",
        "": "N"
    },
    {
        "名称": "disabled",
        "类型": "Boolean",
        "默认值": "false",
        "说明": "禁用状态",
        "必传": "N",
        "": ""
    },
    {
        "名称": "ghost",
        "类型": "Boolean",
        "默认值": "false",
        "说明": "是否为幽灵按钮（镂空按钮）",
        "必传": "N",
        "": ""
    },
    {
        "名称": "href",
        "类型": "String",
        "默认值": "-",
        "说明": "跳转地址。href 存在时，按钮标签默认使用 <a> 渲染；如果指定了 tag 则使用指定的标签渲
染",
        "必传": "N",
        "": ""
    },
    {
        "名称": "icon",
        "类型": "Slot / Function",
        "默认值": "-",
        "说明": "按钮内部图标，可完全自定义。TS 类型：TNode。通用类型定义",
        "必传": "N",
        "": ""
    },
    {
        "名称": "loading",
        "类型": "Boolean",
        "默认值": "false",
        "说明": "是否显示为加载状态",
        "必传": "N",
        "": ""
    },
    {
        "名称": "shape",
        "类型": "String",
        "默认值": "rectangle",
        "说明": "按钮形状，有 4 种：长方形、正方形、圆角长方形、圆形。可选项：
rectangle/square/round/circle",
        "必传": "N",
        "": ""
    },
    {
        "名称": "size",
        "类型": "String",
        "默认值": "medium",
        "说明": "组件尺寸。可选项：small/medium/large。TS 类型：SizeEnum。通用类型定义",
        "必传": "N",
        "": ""
    },
    {
        "名称": "suffix",
        "类型": "Slot / Function",
        "默认值": "-",
        "说明": "右侧内容，可用于定义右侧图标。TS 类型：TNode。通用类型定义",
        "必传": "N",
        "": ""
    },
    {
        "名称": "tag",
        "类型": "String",
        "默认值": "-",
        "说明": "渲染按钮的 HTML 标签，默认使用标签 <button> 渲染，可以自定义为 <a> <div> 等。透传全部 HTML 属性，如：href/target/data-* 等。⚠️ 禁用按钮 <button disabled>无法显示 Popup 浮层信息，可通过修改 tag=div 解决这个问题。可选项：button/a/div",
        "必传": "N",
        "": ""
    },
    {
        "名称": "theme",
        "类型": "String",
        "默认值": "-",
        "说明": "组件风格，依次为默认色、品牌色、危险色、警告色、成功色。可选项：default/primary/danger/warning/success",
        "必传": "N",
        "": ""
    },
    {
        "名称": "type",
        "类型": "String",
        "默认值": "button",
        "说明": "按钮类型。可选项：submit/reset/button",
        "必传": "N",
        "": ""
    },
    {
        "名称": "variant",
        "类型": "String",
        "默认值": "base",
        "说明": "按钮形式，基础、线框、虚线、文字。可选项：base/outline/dashed/text",
        "必传": "N",
        "": ""
    },
    {
        "名称": "onClick",
        "类型": "Function TS 类型：(e: MouseEvent) => void 点击时触发",
        "默认值": "",
        "说明": "",
        "必传": "N",
        "": ""
    }
]
```

Button 组件用法示例

```
<template>
  <t-space size="24px">
    <t-button theme="default" variant="base">填充按钮</t-button>
    <t-button theme="default" variant="outline">描边按钮</t-button>
    <t-button theme="default" variant="dashed">虚框按钮</t-button>
    <t-button theme="default" variant="text">文字按钮</t-button>
  </t-space>
</template>
```

Divder 组件文档

```
{"名称":"align","类型":"String","默认值":"center","说明":"文本位置（仅在水平分割线有效）。可选项：left/right/center","必传":"N"}
{"名称":"content","类型":"String / Slot / Function","默认值":"-","说明":"子元素。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"dashed","类型":"Boolean","默认值":"false","说明":"是否虚线（仅在水平分割线有效）","必传":"N"}
{"名称":"default","类型":"String / Slot / Function","默认值":"-","说明":"子元素，同 content。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"layout","类型":"String","默认值":"horizontal","说明":"分隔线类型有两种：水平和垂直。可选项：horizontal/vertical","必传":"N"}
{"名称":"theme","类型":"String","默认值":"-","说明":"已废弃。请更为使用 layout。分隔线类型有两种：水平和垂直。可选项：horizontal/vertical","必传":"N"}
```

Divder 组件用法示例

```
<template>
  <t-space direction="vertical">
    <p>
      这是第一种类型的内容元素或信息数据，通过分割线进行分割。分割线方便用户清晰明确的读取和预览，减少信息符合，提高用户使用效率。避免大段内容的聚集、信息负荷过多，导致信息获取或内容操作效率降低。
    </p>
    <t-divider />
    <p>
      这是第二种类型的内容元素或信息数据，通过分割线进行分割。分割线方便用户清晰明确的读取和预览，减少信息符合，提高用户使用效率。避免大段内容的聚集、信息负荷过多，导致信息获取或内容操作效率降低。
    </p>
</template>
```

AutoComplete 自动填充组件文档

```
{"名称":"autofocus","类型":"Boolean","默认值":"-","说明":"自动获取焦点","必传":"N"}
{"名称":"clearable","类型":"Boolean","默认值":"-","说明":"是否允许清空","必传":"N"}
{"名称":"default","类型":"String / Slot / Function","默认值":"-","说明":"触发显示联想词下拉框的元素，同 triggerElement。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"disabled","类型":"Boolean","默认值":"-","说明":"是否禁用","必传":"N"}
{"名称":"filter","类型":"Function","默认值":"-","说明":"自定义过滤规则，用于对现有数据进行搜索过滤，判断是否过滤某一项数据。参数 filterWords 表示搜索词，option表示单个选项内容，返回值为 true 保留该选项，返回值为 false 则隐藏该选项。使用该方法时无需设置 filterable。TS 类型：(filterWords: string, option: T) => boolean | Promise<boolean>","必传":"N"}
{"名称":"filterable","类型":"Boolean","默认值":"true","说明":"是否根据输入内容过滤联想词。默认过滤规则不区分大小写，全文本任意位置匹配。如果默认搜索规则不符合业务需求，可以更为使用 filter 自定义过滤规则。部分场景下输入关键词和下拉联想词完全不同，此时可以设置为 false","必传":"N"}
{"名称":"highlightKeyword","类型":"Boolean","默认值":"true","说明":"是否高亮联想词中和输入值的相同部分","必传":"N"}
{"名称":"inputProps","类型":"Object","默认值":"-","说明":"透传 Input 组件全部特性。TS 类型：InputProps，Input API Documents。详细类型定义","必传":"N"}
{"名称":"options","类型":"Array","默认值":"-","说明":"下拉联想词列表。示例一：['联想词一', '联想词二']。示例二：{ label: () => <div>联想词元素</div>, text: '用于搜索的纯联想词' }。TS 类型：Array<T>","必传":"N"}
{"名称":"panelBottomContent","类型":"String / Slot / Function","默认值":"-","说明":"面板内的底部内容。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"panelTopContent","类型":"String / Slot / Function","默认值":"-","说明":"面板内的顶部内容。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"placeholder","类型":"String","默认值":"undefined","说明":"输入框为空时的占位提示。组件本身默认值为 undefined，但全局配置存在默认值，不同语言全局默认值不同","必传":"N"}
{"名称":"popupProps","类型":"Object","默认值":"-","说明":"透传 Popup 组件全部特性。TS 类型：PopupProps，Popup API Documents。详细类型定义","必传":"N"}
{"名称":"readonly","类型":"Boolean","默认值":"-","说明":"是否只读","必传":"N"}
{"名称":"size","类型":"String","默认值":"medium","说明":"组件尺寸。可选项：small/medium/large。TS 类型：SizeEnum。通用类型定义","必传":"N"}
{"名称":"status","类型":"String","默认值":"default","说明":"输入框状态。可选项：default/success/warning/error","必传":"N"}
```

```
{"名称":"textareaProps","类型":"Object","默认值":"-","说明":"透传 Textarea 组件全部特性。TS 类型：TextareaProps，Textarea API Documents。详细类型定义","必传":"N"}
{"名称":"tips","类型":"String / Slot / Function","默认值":"-","说明":"输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"triggerElement","类型":"String / Slot / Function","默认值":"-","说明":"触发显示联想词下拉框的元素，默认为 Input 组件，可以使用 trigger 自定义为 Textarea 组件或其他组件。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"value","类型":"String","默认值":"-","说明":"输入框的值，即当前指定的联想词。支持语法糖 v-model 或 v-model:value","必传":"N"}
{"名称":"defaultValue","类型":"String","默认值":"-","说明":"输入框的值，即当前指定的联想词。非受控属性","必传":"N"}
{"名称":"onBlur","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: FocusEvent; value: string }) => void"}
{"名称":"失去焦点时触发","类型":"N"}
{"名称":"onChange","类型":"Function","默认值":"","说明":"TS 类型：(value: string, context?: { e?: InputEvent | MouseEvent | CompositionEvent | KeyboardEvent }) => void"}
{"名称":"输入框值发生变化时触发","类型":"N"}
{"名称":"onClear","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: MouseEvent }) => void"}
{"名称":"清空按钮点击时触发","类型":"N"}
{"名称":"onCompositionend","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: CompositionEvent; value: string }) => void"}
{"名称":"中文输入结束时触发","类型":"N"}
{"名称":"onCompositionstart","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: CompositionEvent; value: string }) => void"}
{"名称":"中文输入开始时触发","类型":"N"}
{"名称":"onEnter","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: KeyboardEvent; value: string }) => void"}
{"名称":"回车键按下时触发","类型":"N"}
{"名称":"onFocus","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: FocusEvent; value: string }) => void"}
{"名称":"获得焦点时触发","类型":"N"}
{"名称":"onSelect","类型":"Function","默认值":"","说明":"TS 类型：(value: string, context: { e: MouseEvent | KeyboardEvent }) => void"}
{"名称":"选中联想词时触发","类型":"N"}
```

AutoComplete 组件使用文档

```
<template>
  <t-space>
    <!-- 组件内置的过滤规则：不区分大小写，文本任意位置匹配 -->
    <t-auto-complete
      v-model="value1"
      :options="options"
      highlight-keyword
      filterable
      placeholder="组件默认过滤规则（不区分大小写）"
      style="width: 280px"
    />

  </t-space>
</template>

<script setup>
import { ref } from 'vue';

const LIST = ['第一个 AutoComplete 默认联想词', '第二个 AutoComplete 默认联想词', '第三个 AutoComplete 默认联想词'];

const value1 = ref('');
const options = ref([...LIST]);

function filterWords(keyword, option) {
  const regExp = new RegExp(keyword);
  return regExp.test(option.text);
}
</script>
```

```

TreeSelect 组件使用文档

```

TreeSelect Props
{"名称":"autoWidth","类型":"Boolean","默认值":"false","说明":"宽度随内容自适应","必传":"N"}
{"名称":"borderless","类型":"Boolean","默认值":"false","说明":"【开发中】无边框模式","必传":"N"}
{"名称":"clearable","类型":"Boolean","默认值":"false","说明":"是否允许清空","必传":"N"}
{"名称":"collapsedItems","类型":"Slot / Function","默认值":"-","说明":"多选情况下，用于设置折叠项内容，默认为 +N。如果需要悬浮就显示其他内容，可以使用 collapsedItems 自定义。TS 类型：TNode<{ value: DataOption[]; collapsedSelectedItems: DataOption[]; count: number }>。通用类型定义","必传":"N"}
{"名称":"data","类型":"Array","默认值":"[]","说明":"数据。TS 类型：Array<DataOption>","必传":"N"}
{"名称":"disabled","类型":"Boolean","默认值":"-","说明":"是否禁用组件","必传":"N"}
{"名称":"empty","类型":"String / Slot / Function","默认值":"''","说明":"当下拉列表为空时显示的内容。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"filter","类型":"Function","默认值":"-","说明":"过滤方法，用于对现有数据进行搜索过滤，判断是否过滤某一项数据。TS 类型：(filterWords: string, option: DataOption) => boolean","必传":"N"}
{"名称":"filterable","类型":"Boolean","默认值":"false","说明":"是否可搜索","必传":"N"}
{"名称":"inputProps","类型":"Object","默认值":"-","说明":"透传给 输入框 Input 组件的全部属性。TS 类型：InputProps，Input API Documents。详细类型定义","必传":"N"}
{"名称":"inputValue","类型":"String / Number","默认值":"-","说明":"输入框的值。支持语法糖 v-model:inputValue。TS 类型：InputValue，Input API Documents。详细类型定义","必传":"N"}
{"名称":"defaultInputValue","类型":"String / Number","默认值":"-","说明":"输入框的值。非受控属性。TS 类型：InputValue，Input API Documents。详细类型定义","必传":"N"}
{"名称":"loading","类型":"Boolean","默认值":"false","说明":"是否正在加载数据","必传":"N"}
{"名称":"loadingText","类型":"String / Slot / Function","默认值":"-","说明":"远程加载时显示的文字，支持自定义。如加上超链接。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"max","类型":"Number","默认值":"0","说明":"用于控制多选数量，值为 0 则不限制","必传":"N"}
{"名称":"minCollapsedNum","类型":"Number","默认值":"0","说明":"最小折叠数量，用于多选情况下折叠选中项，超出该数值的选中项折叠。值为 0 则表示不折叠","必传":"N"}
{"名称":"multiple","类型":"Boolean","默认值":"false","说明":"是否允许多选","必传":"N"}
{"名称":"placeholder","类型":"String","默认值":"undefined","说明":"占位符","必传":"N"}
{"名称":"popupProps","类型":"Object","默认值":"-","说明":"透传给 popup 组件的全部属性。TS 类型：PopupProps，Popup API Documents。详细类型定义","必传":"N"}
{"名称":"popupVisible","类型":"Boolean","默认值":"-","说明":"是否显示下拉框","必传":"N"}
{"名称":"prefixIcon","类型":"Slot / Function","默认值":"-","说明":"组件前置图标。TS 类型：TNode。通用类型定义","必传":"N"}
{"名称":"readonly","类型":"Boolean","默认值":"false","说明":"只读状态，值为真会隐藏输入框，且无法打开下拉框","必传":"N"}
{"名称":"selectInputProps","类型":"Object","默认值":"-","说明":"【开发中】透传 SelectInput 筛选器输入框组件的全部属性。TS 类型：SelectInputProps，SelectInput API Documents。详细类型定义","必传":"N"}
{"名称":"size","类型":"String","默认值":"medium","说明":"尺寸。可选项：small/medium/large","必传":"N"}
{"名称":"status","类型":"String","默认值":"-","说明":"输入框状态。可选项：default/success/warning/error","必传":"N"}
{"名称":"tagProps","类型":"Object","默认值":"-","说明":"【开发中】透传 Tag 标签组件全部属性。TS 类型：TagProps，Tag API Documents。详细类型定义","必传":"N"}
{"名称":"tips","类型":"String / Slot / Function","默认值":"-","说明":"输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型定义","必传":"N"}
{"名称":"treeProps","类型":"Object","默认值":"-","说明":"透传 Tree 组件的全部属性。TS 类型：TreeProps，Tree API Documents。详细类型定义","必传":"N"}
{"名称":"value","类型":"String / Number / Object / Array","默认值":"-","说明":"选中值。支持语法糖 v-model 或 v-model:value。TS 类型：TreeSelectValue type TreeSelectValue = string | number | object | Array<TreeSelectValue>。详细类型定义","必传":"N"}
{"名称":"defaultValue","类型":"String / Number / Object / Array","默认值":"-","说明":"选中值。非受控属性。TS 类型：TreeSelectValue type TreeSelectValue = string | number | object | Array<TreeSelectValue>。详细类型定义","必传":"N"}
{"名称":"valueDisplay","类型":"Slot / Function","默认值":"-","说明":"自定义选中项呈现方式。TS 类型：string | TNode<{ value: TreeSelectValue; onClose: (index: number, item?: any) => void }>。通用类型定义","必传":"N"}
{"名称":"valueType","类型":"String","默认值":"value","说明":"用于控制选中值的类型。假设数据选项为：[{ label: '姓名', value: 'name' }]，value 表示值仅返回数据选项中的 value， object 表示值返回全部数据。可选项：value/object","必传":"N"}
{"名称":"onBlur","类型":"Function","默认值":"","说明":"TS 类型：(context: { value: TreeSelectValue; e: FocusEvent }) => void"}
{"名称":"输入框失去焦点时触发","类型":"N"}

```
{"名称":"onChange","类型":"Function","默认值":"","说明":"TS 类型：(value: TreeSelectValue, context: { node: TreeNodeModel<DataOption>; trigger: TreeSelectValueChangeTrigger; e?: MouseEvent | KeyboardEvent }) => void"}
{"名称":"节点选中状态变化时触发，context.node 表示当前变化的选项，context. trigger 表示触发变化的来源。详细类型定义。"}
{"名称":"type TreeSelectValueChangeTrigger = 'clear' | 'tag-remove' | 'backspace' | 'check' | 'uncheck'"}
{"名称":"N"}
{"名称":"onClear","类型":"Function","默认值":"","说明":"TS 类型：(context: { e: MouseEvent }) => void"}
{"名称":"点击清除按钮时触发","类型":"N"}
{"名称":"onFocus","类型":"Function","默认值":"","说明":"TS 类型：(context: { value: TreeSelectValue; e: FocusEvent }) => void"}
{"名称":"输入框获得焦点时触发","类型":"N"}
{"名称":"onInputChange","类型":"Function","默认值":"","说明":"TS 类型：(value: InputValue, context?: SelectInputValueChangeContext) => void"}
{"名称":"输入框值发生变化时触发，context.trigger 表示触发输入框值变化的来源：文本输入触发、清除按钮触发、失去焦点等。详细类型定义。"}
{"名称":"import { SelectInputValueChangeContext } from '@SelectInput'"}
{"名称":"N"}
{"名称":"onPopupVisibleChange","类型":"Function","默认值":"","说明":"TS 类型：(visible: boolean, context: PopupVisibleChangeContext) => void"}
{"名称":"下拉框显示或隐藏时触发。详细类型定义。"}
{"名称":"import { PopupVisibleChangeContext } from '@Popup'"}
{"名称":"N"}
{"名称":"onRemove","类型":"Function","默认值":"","说明":"TS 类型：(options: RemoveOptions<DataOption>) => void"}
{"名称":"多选模式下，选中数据被移除时触发。详细类型定义。"}
{"名称":"interface RemoveOptions<T> { value: string | number | object; data: T; e?: MouseEvent }"}
{"名称":"N"}
{"名称":"onSearch","类型":"Function","默认值":"","说明":"TS 类型：(filterWords: string) => void"}
{"名称":"输入值变化时，触发搜索事件。主要用于远程搜索新数据","类型":"N"}
```

```
TreeSelect Events
{"名称":"blur","参数":"(context: { value: TreeSelectValue; e: FocusEvent })","描述":"输入框失去焦点时触发"}
{"名称":"change","参数":"(value: TreeSelectValue, context: { node: TreeNodeModel<DataOption>; trigger: TreeSelectValueChangeTrigger; e?: MouseEvent | KeyboardEvent })","描述":"节点选中状态变化时触发，context.node 表示当前变化的选项，context. trigger 表示触发变化的来源。详细类型定义。"}
{"名称":"type TreeSelectValueChangeTrigger = 'clear' | 'tag-remove' | 'backspace' | 'check' | 'uncheck'"}
{"名称":"clear","参数":"(context: { e: MouseEvent })","描述":"点击清除按钮时触发"}
{"名称":"focus","参数":"(context: { value: TreeSelectValue; e: FocusEvent })","描述":"输入框获得焦点时触发"}
{"名称":"input-change","参数":"(value: InputValue, context?: SelectInputValueChangeContext)","描述":"输入框值发生变化时触发，context.trigger 表示触发输入框值变化的来源：文本输入触发、清除按钮触发、失去焦点等。详细类型定义。"}
{"名称":"import { SelectInputValueChangeContext } from '@SelectInput'"}
{"名称":"popup-visible-change","参数":"(visible: boolean, context: PopupVisibleChangeContext)","描述":"下拉框显示或隐藏时触发。详细类型定义。"}
{"名称":"import { PopupVisibleChangeContext } from '@Popup'"}
{"名称":"remove","参数":"(options: RemoveOptions<DataOption>)","描述":"多选模式下，选中数据被移除时触发。详细类型定义。"}
{"名称":"interface RemoveOptions<T> { value: string | number | object; data: T; e?: MouseEvent }"}
{"名称":"search","参数":"(filterWords: string)","描述":"输入值变化时，触发搜索事件。主要用于远程搜索新数据"}

```

TreeSelect 组件的基本用法

```

<template>
  <t-tree-select
    v-model="value"
    :data="options"
    clearable
    placeholder="请选择"
    :popup-visible="popupVisible"
    @popup-visible-change="onVisibleChange"
  />
</template>
<script setup>
import { ref } from 'vue';

const options = [
  {
    label: '广东省',
    value: 'guangdong',
    children: [
      {
        label: '广州市',
        value: 'guangzhou',
      },
      {
        label: '深圳市',
        value: 'shenzhen',
      },
    ],
  },
  {
    label: '江苏省',
    value: 'jiangsu',
    disabled: true,
    children: [
      {
        label: '南京市',
        value: 'nanjing',
      },
      {
        label: '苏州市',
        value: 'suzhou',
      },
    ],
  },
];

const value = ref('');
const popupVisible = ref(false);

const onVisibleChange = (visible, context) => {
  console.log(visible, context);
  if (context.trigger || context.node?.label !== '广州市') {
    popupVisible.value = visible;
  }
};
</script>
```