# 多功能信息展示

## 概述

本组件是一个高度可配置的数据展示和交互工具，它支持丰富的数据类型和格式，包括文本、图片和链接等。它的设计目标是提供一个灵活、高效和用户友好的数据展示和交互界面。

组件的主要特性包括：

1. **懒加载**：组件支持懒加载数据，这意味着只有当数据需要显示在屏幕上时，才会从服务器加载。这种方式可以显著提高应用的性能，特别是在处理大量数据时。
2. **无延迟展示**：通过使用Vue的异步更新队列，组件能够在数据加载完成后立即更新视图，从而实现无延迟的数据展示。
3. **高度可配置**：组件提供了多种配置选项，包括数据的显示方式（如每行显示的项目数量）、项目的样式（如是否显示边框、标题和更多链接）等。这使得组件可以适应各种不同的使用场景。
4. **丰富的交互功能**：组件支持点击事件，可以根据用户的操作触发不同的行为。例如，用户可以点击一个项目的更多链接，来查看该项目的详细信息。
5. **优雅的样式**：组件使用了现代的CSS技术，如Flexbox布局和CSS变量，来实现优雅和响应式的设计。此外，组件还提供了一些预定义的样式，如悬停效果和自定义滚动条，以提高用户体验。

该Vue组件是一个强大而灵活的工具，可以帮助开发者快速创建出高效、美观和用户友好的数据展示和交互界面。

## 开发环境

* Node.js：由于项目使用了Vue CLI和ESLint等工具，需要安装Node.js环境。具体版本没有在项目文件中指定，但建议使用最新的稳定版本。
* Vue CLI：项目使用了Vue CLI来进行项目的构建和开发，需要全局安装Vue CLI，建议安装的Vue CLI版本为~5.0.0。
* Faker.js：这是一个生成大量假数据的库，版本为8.0.2。
* Babel：项目使用了Babel来进行代码的转译，需要安装Babel，建议安装的Babel版本为^7.12.16。
* ESLint：项目使用了ESLint来进行代码的检查，需要安装ESLint，建议安装的ESLint版本为^7.32.0。
* Vue.js：这是一个用于构建用户界面的渐进式框架，版本为2.6.14。
* D3.js：这是一个用于数据驱动文档的JavaScript库，版本为7.8.5。
* Core-js：这是JavaScript的模块化标准库，版本为3.8.3。
* Vue Router：这是Vue.js的官方路由管理器，版本为3.0.2。
* Babel：这是一个JavaScript编译器，用于将ES6代码转换为向后兼容的JavaScript版本，版本为7.12.16。
* ESLint：这是一个开源的JavaScript代码检查工具，用于发现代码错误和统一代码风格，版本为7.32.0。

- 浏览器：

- Google Chrome

- Mozilla Firefox

- Safari

- Microsoft Edge

## 属性

| **属性名** | **说明** | **类型** | **必填** | **默认值** | **可选值** |
| --- | --- | --- | --- | --- | --- |
| valueData | 显示的数据 | Object | 是 | — | — |
| details | 展示内容配置 | Array | 是 | — | — |
| pageSize | 每页展示条数 | Number | 否 | 50 | — |
| currentPage | 当前页码 | Number | 否 | 1 | — |
| position | 对齐方式 | Stirng | 否 | center | left/center/right |
| showBorder | 是否显示边框 | Boolean | 否 | true | true/false |
| itemsPerRow | 卡片展示布局列数 | Number | 否 | 1 | — |
| columns | 卡片信息展示布局列数 | Number | 否 | 1 | — |
| isShowTitle | 是否显示内容标题 | Boolean | 否 | false | true/false |
| isShowMore | 是否显示更多按钮 | Boolean | 否 | false | true/false |
| moreHtml | 更多按钮内容 | String | 否 | 更多+ | — |

## 插槽

| **插槽名** | **说明** |
| --- | --- |
| - | 自定义默认内容 |
| detail | 每一个卡面内的展示内容 |

## 代码样例

### 默认

|  |
| --- |
| <div style="width: 50%;margin: 0 auto">  <lx-dropdown-scrolling  :details="showTitle"  :valueData="show"  :pageSize="50"  position="center"  :itemsPerRow="1"  :columns="1"  :isShowTitle="false"  :showBorder="true"  :isShowMore="true"  moreHtml="更多+"  @linkClick="linkClick"  >  </lx-dropdown-scrolling>  </div>   import lxDropdownScrolling from '@/components/dropdownScrolling.vue'  export default {  name: 'App',  data() {  return {  show:[],  showTitle:[],  };  },  components: {  lxDropdownScrolling,  },  methods:{  },  created() {  this.showTitle = [  {key:'name',value:'名称',lableWidth:120,labelSize:30, type: 'link',clickType:'select'},  {key:'avatar',value:'头像',lableWidth:120,labelSize:15, type: 'image'},  {key:'email',value:'电子邮件',lableWidth:120,labelSize:15, type: 'link',clickType:'update'},  {key:'protocol',value:'协议',lableWidth:120,labelSize:15, type: 'text'},  {key:'url',value:'URL地址',lableWidth:120,labelSize:15, type: 'text'},  {key:'ip',value:'IP地址',lableWidth:120,labelSize:15, type: 'text'},  {key:'ipv6',value:'协议6',lableWidth:120,labelSize:15, type: 'text'},  {key:'port',value:'端口',lableWidth:120,labelSize:15, type: 'text'},  ]  this.show = getData(1000,true)  } } </script>  <style>  </style> |

### 使用插槽

|  |
| --- |
| <lx-dropdown-scrolling  :details="showTitle"  :valueData="show"  :pageSize="50"  position="center"  :itemsPerRow="1"  :columns="1"  :isShowTitle="false"  :showBorder="true"  :isShowMore="true"  moreHtml="更多+"  @linkClick="linkClick"  >  <template v-slot:detail="slotProps">  <div>{{ slotProps.detail.name }}</div>  <div><img :src="slotProps.detail.avatar" /></div>  <div>{{ slotProps.detail.email }}</div>  <div>{{ slotProps.detail.protocol }}</div>  </template>  </lx-dropdown-scrolling> |

## 图例



