

Muz-Doraemon

基于Uni-App的VUE3组件库——Жидзин (Zidjin) 矢争系列组件库。

注意：文档编撰格式参考 [Element UI](#) 组件文档。

安装

推荐使用 npm 的方式安装。

```
npm install muz-doraemon
```

引入

全局引入，在 pages.json 中写入以下内容：

```
{
  "pages": [],
  "globalStyle": {
    "navigationStyle": "custom" // 启用自定义导航条
  },
  "easycom": {
    "autoscan": true, // 自动查找安装、引用、注册符合“components/组件名称/组件名称.vue”目录结构的组件。
    "custom": {
      "^muz-(.*)": "muz-doraemon/src/muz-$1/muz-$1.vue" // 自定义组件正则匹配规则，如果需要匹配node_modules内的vue文件，需要使用packageName/path/to/vue-file-$1.vue形式的匹配规则
    }
  }
}
```

局部引入，在 vue 页面文件中写入以下内容

```
import {MuzItem, MuzListArea, ...} from 'muz-doraemon';
export default {
  components: {
    MuzItem,
    MuzListArea,
    ...
  },
};
```

语法糖 - 模板字符串

x-template-string.js

一、直接引用值。

基础版——直接引用值。

语法: `{{key}}`

```
// 例子:
import XTemplateString from "./x-template-string.js";
const param = {category_id: 35, category_value: "杭州西湖区"};
const str = "类别ID: {{category_id}}, 类别值: {{category_value}}。";
XTemplateString.format(param, str);
// 结果 ==> "类别ID: 35, 类别值: 杭州西湖区。"
```

二、管道选项值

高级版——管道选项值。即指定列表对象、标签键、值键，找到引用值。

语法: `{{valueKey: key | options[labelKey]}}`

```
// 例子:
import XTemplateString from "./x-template-string.js";
const param = {category_id: 35, category_list: [{value: 27, label: "杭州上城区"},
{value: 35, label: "杭州西湖区"}]};
const str = "类别: {{ value: category_id | category_list[label] }}"
XTemplateString.format(param, str);
// 结果 ==> "类别: 杭州西湖区。"
```

三、强制转换值

高级版——强制转换值。即指定来源键、目标键，生成数组对象。

语法: `{{(Array <targetKey: sourceKey, ...>) valueKey<splitFlag>}}`

```
// 例子:
import XTemplateString from "./x-template-string.js";
const param = {photo_list: [{link: '/images/a.jpg', label: "杭州上城区"}, {link:
'/images/b.jpg', label: "杭州西湖区"}]};
const str = "{{ (Array< url: link, text: label >)photo_list }}"
const str2 = "{{ (Array< :link >)photo_list }}"
XTemplateString.format(param, str);
// 结果 ==> "[{url: '/images/a.jpg', text: "杭州上城区"}, {url: '/images/b.jpg',
text: "杭州西湖区"}]"
XTemplateString.format(param, str2);
// 结果 ==> "['/images/a.jpg', '/images/b.jpg']"
```

Top-Bar 顶部栏

顶部栏

基础用法

```
<template>
  <muz-plus-bar title="标题" :is-search="true" />
</template>

<script>
  import {ZjTable} from 'zj-element';
  export default {
    data: ()=>({
    })
  };
</script>
```

Nav-Bar Attributes

参数	说明	类型	可选值	默认值
title	标题	string	—	标题
rtext	右侧文件	string	—	—
root	是否根页, 无返回	boolean	true	false
is-search	是否显示搜索	boolean	true	false
is-add	是否显示添加	boolean	true	false
is-edit	是否显示编辑	boolean	true	false
is-delete	是否显示删除	boolean	true	false
is-scan	是否显示扫码	boolean	true	false
is-filter	是否显示过滤	boolean	true	false
is-more	是否显示更多	boolean	true	false
is-with	是否显示分享???	boolean	true	false
is-left-search	是否显示右侧搜索	boolean	true	false
is-plus	是否显示加号	boolean	true	false
is-customize	是否左侧自定义	boolean	true	false
customize-title	自定义内容 文字	string	—	—
is-off	是否显示左侧关闭	boolean	true	false
is-dropdown	是否显示下拉	boolean	true	false
dropdownDelete	下拉的删除	boolean	true	false
dropdownEdit	下拉的编辑	boolean	true	false
dropdownWith	下拉的分享???	boolean	true	false

Tab-Bar 标签栏

标签栏

基础用法

```

<template>
  <muz-tab-bar v-if="it.component === 'muz-tab-bar'" :default="it.default"
  :page-param="currentParam" :config="it.config" :request="it.request"
  @tab="onTab" />
</template>

<script>
  export default {
    data: ()=>({

    })
  };
</script>

```

Tab-Bar Attributes

参数	说明	类型	可选值	默认值
class	自定义类名	string	—	—
styles	自定义样式	object	—	—
default	选中高亮显示	[String, Number]	—	—
pageParam	界面传参数	Object		
config	当前激活的导航按钮	Object	—	—
request	请求导航条配置	Object		

Option of Config

参数	说明	类型	可选值	默认值
nodeLabel	节点标签文字	string	—	—
nodeValue	节点值	string	—	valueNode: "node_name",
unshiftList	头部插入标签列表	Array[Object]		unshiftList: [{value: "全部"}], 注意: value与valueNode的值保持一致
appendList	尾部插入标签列表	Array[Object]		appendList: [{value: "其他"}], 注意: value与valueNode的值保持一致

Nav-Bar 导航栏

导航栏

基础用法

```
<template>
  <muz-nav-bar current-nav="home" :nav-list="list"></muz-nav-bar>
</template>

<script>
import {ZjTable} from 'zj-element';
export default {
  data: ()=>({
    list: [
      {
        key: 'home',
        icon: '/static/icons/icon-home.svg',
        activeIcon: '/static/icons/icon-home-color.svg',
        title: '首页',
        url: '/pages/home/home',
      },
      {
        key: 'me',
        icon: '/static/icons/icon-me.svg',
        activeIcon: '/static/icons/icon-me-color.svg',
        title: '我',
        url: '/pages-me/me/me',
      },
    ],
  })
};
</script>
```

Nav-Bar Attributes

参数	说明	类型	可选值	默认值
current-nav	当前激活的导航按钮	string	—	—
current	当前激活的下标按钮	number	—	-1
nav-list	导航条配置	array	{ key: 'me', icon: '/static/icon-me.svg', activeIcon: '/static/icon-me-color.svg', title: '我', url: '/pages-me/me/me', },	—
request	请求导航条配置			
close-url-active	关闭自动检查激活即根据Url地址匹配激活	boolean	true	false
is-tab-mode	启用标签, 暂时弃用	boolean	true	false

Doraemon 哆啦A梦

哆啦A梦的组件百宝箱。

基础用法

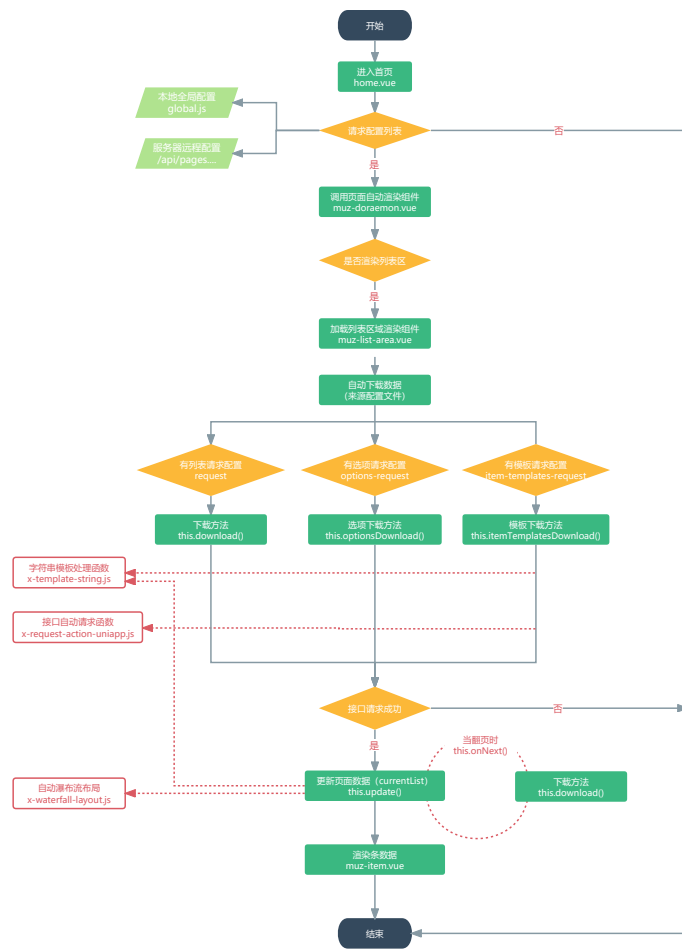
```

<template>
  <muz-doraemon :list="list"></muz-doraemon>
</template>

<script>
  import {MuzDoraemon} from 'muz-doraemon';
  export default {
    data: ()=>({
      list: [
        ...
      ],
    })
  };
</script>

```

UML流程图



Doraemon Attributes

参数	说明	类型	可选值	默认值
root-url	根域名。小程序/APP 需要添加域名	string	—	—
page-param	传入页面参数，用于定制化不同页面过滤结果	object	{page.id,nav,page}	—
list	组件化的列表配置	array	—	—

Option of List

参数	说明	类型	可选值	默认值
show	是否显示	boolean	false	true
component	组件名称。 符合渲染配置化的 组件。	string	muz-background, muz-divider, muz-classify, muz-item, muz-list-area, muz-item-area	—
class	自定义类名	string	—	—
styles	自定义样式。 作用于组件根标 签。	object	例: {fontSize: "14px"},	—
config	组件配置。 详见本文组件。	object	—	—
list 适用: muz- classify	分类列表	array	例: [{ icon: '/icon/news.svg', text: '投标报价', value: "", }],	—
itemTemplate 适用: muz-item muz-list- area, muz-item- area	条数据模版。 详见: Item 条数据	object	{ title: { title: "标题 Один One", hot_tip: "热门", }, bars: [{ type: 'bit', text: '这是文字', }] },	—
itemTemplateSet 适用: muz-list- area	条数据模版集合。			

参数	说明	类型	可选值	默认值
request 适用: muz-list-area, muz-item-area	组件数据请求配置	object	例: { url: '/api/news_list', rootUrl: ' http://xx.cn ', method: 'GET', param: {}, headers: {}, page: 1, limit: 10, showMore: false, nodeStatus: 'err', nodeStatusValue: 0, nodeData: 'data.list', }	—
optionsRequest 适用: muz-list-area, muz-item-area	多组选项的请求配置	array	例: [{ url: '/api/thread_category_list/', method: 'GET', param: {}, headers: {}, nodeStatus: 'err', nodeStatusValue: 0, nodeData: 'data', options_key: 'category_options', }, ...]	

Doraemon Slot

名称	说明
	自定义插槽。 详见: 条数据 .slot

Background 背景

背景组件。

基础用法

```
<template>
  <muz-background :config="config" />
</template>

<script>
```

```
import {ZjTable} from 'zj-element';
export default {
  data: ()=>({
    config: {
      backgroundImage: "/images/bg.png"
    },
  })
};
</script>
```

Background Attributes

参数	说明	类型	可选值	默认值
class	自定义类名	string	—	—
styles	自定义样式	object	—	—
config	组件配置	object	—	—

Option of Config

参数	说明	类型	可选值	默认值
backgroundImage	背景图片	string	—	—

Divider 分隔行

分隔行组件。

基础用法

```
<template>
  <muz-divider :config="it.config" />
</template>

<script>
import {ZjTable} from 'zj-element';
export default {
  data: ()=>({
    config: {
      backgroundImage: "/images/bg.png"
    },
  })
};
</script>
```

Divider Attributes

参数	说明	类型	可选值	默认值
class	自定义类名	string	—	—
styles	自定义样式	object	—	—
config	组件配置	object	—	—

Option of Config

参数	说明	类型	可选值	默认值
title	分隔行标题	string	—	—
url	点击标题的跳转地址 需启用clickable	string	—	—
backgroundImage	背景图片	string	—	—
clickable	启动跳转	boolean	false	true

Classify 分类图标

分类图标组件。

基础用法

```
<template>
  <muz-classify :config="it.config" :list="list" @classify="onClassify" />
</template>

<script>
  export default {
    data: ()=>({
      config: {
        backgroundImage: "/images/bg.png"
      },
    })
  };
</script>
```

Classify Attributes

参数	说明	类型	可选值	默认值
class	自定义类名	string	—	—
styles	自定义样式	object	—	—
text	分类标题	string	—	—
subtitle	副标题	string	—	—
config	组件配置	object	—	—
list	分类列表	array	例: <pre>[{ icon: '/icon/news.svg', text: '投标报价', value: "", }]</pre>	—

Classify Event

事件名称	说明	回调参数
classify	点击分类图标时触发	(item: 目前的点击对象)
emit 未实现!!!	点击分类图标时触发	(emit: 信号名称, item: 目前的点击对象)

Option of Config

参数	说明	类型	可选值	默认值
noBorder	是否无边	boolean	true	false
isCircle	是否圆形图标	boolean	—	—
thumb	图标大小	string	medium / small / mini	medium

Option of List

参数	说明	类型	可选值	默认值
icon	图标地址 如果不传则默认显示文本第一个字符	string	—	—
text	文本	string	—	—
sidenote	旁注	string	—	—
link	跳转链接	string	—	—

Menu 菜单组件

菜单组件

基本用法

```
<template>
  <muz-menu class="menu" :list="list" thumb-size="medium"></muz-menu>
</template>

<script>
  export default {
    data: ()=>({
      list: [
        {
          text: "我的发布",
          rightText: "全部",
          icon: "",
          to: "/pages/timeline/timeline",
          config: {
            noBorder: true,
            isLeftAlign: true,
            thumb: 'min',
          },
        },
        {
          children: [
            {
              text: "报名中",
              icon: '/static/icons/icon-bid-clock.svg',
              to: ""
            },
          ],
        },
        {
          group: '隐私',
          children: [
            {
              "text": "俄友圈权限",
            },
          ],
        },
      ],
    })
  };
</script>
```

Muz-Menu Attributes

参数	说明	类型	可选值	默认值
list	分类列表	array	—	—
thumbSize	自定义图标大小	object	—	—
listType	列表类型	string	card, menu	menu

####

Option of List

参数	说明	类型	可选值	默认值
groupName	分组名称	string	—	—
icon	菜单图标	string	—	—
text	菜单文本	string	—	—
subtitle	副标题	string	—	—
sidenote	右侧旁注	string	—	—
to	跳转路径	string	—	—
thumbSize	图标尺寸	string	large, medium, small, mini, min	medium
noBorder	是否有边框	boolean	true	false
config	配置项 仅适用： 详见：Classify 分类图标 “config”	object	—	—
children	子菜单列表 同 Option of List	array	—	—

Option of List's Config

参数	说明	类型	可选值	默认值
noBorder	是否有边框	boolean	true	false
isLeftAlign	是否左对齐	boolean	true	false
thumb	跳转路径	string	—	—

Item 条数据

该组件为原生结构式Vue组件（struct as data）。

基本用法

```
<template>
  <muz-item :item="item" @item="onClick" />
</template>

<script>
  export default {
    components: {
      MuzItem: () => 'muz-item', // 注：微信小程序不支持此组件懒加载方式
    },
    data: () => ({
      item: {
        title: {
          text: "标题 Один Один",
          hot_tip: "热门",
        },
        bars: [
          { type: 'bit', text: '这是文字', }
        ]
      },
    })
  }
</script>
```

高级用法

限定性：1、组件限于展示，不处理复杂逻辑，但有信号事件。2、

```
<template>
  <muz-item :item="item" @item="onClick" />
</template>

<script>
  export default {
    components: {
      MuzItem: () => 'muz-item', // 注：微信小程序不支持此组件懒加载方式
    },
    data: () => ({
      item: {
        id: 283576, // 自定义参数，仅用于信号触发的数据传值等
        title: {
          text: "标题 Один Один", hot_tip: "热门",
        },
        lead: {
          cover: { url: "http://i.muztak.cn/images/yellow-river.jpg",
hot_tip: "棒", },
          avatar: { url: "http://i.muztak.cn/images/yellow-river.jpg",
},
      },
    })
  }
</script>
```

```

bars: [
  {
    list:[
      { type: "title", text: "今天节日活动拼团的，可以过来看看。Расцветали яблони и груши, Поплыли туманы над рекой.", hot_tip: "热门",},
    ]
  },
  {
    list:[
      [
        { type: "author", icon:
"http://i.muзtak.cn/images/yellow-river.jpg", text: "滕户纱织", emit: "author", },
        { type: "rank", icon:
"http://i.muзtak.cn/images/icon-rank.png", },
      ],
      [
        { type: "price", icon_code: "r", text: 2500,
unit: "卢布", emit: "author", },
      ]
    ]
  },
  {
    list:[
      { type: "summary", text: "今天节日活动拼团的，可以过来看看看看。Расцветали яблони и груши, Поплыли туманы над рекой. Выходила на берег катюша, На высокий берег на крутой.",},
    ]
  },
  {
    list:[
      [
        { type: "tags", list: ['同城', '圣诞节'], },
      ],
      [
        { type: "bit", icon_code: "M", text: 75,
label: "共", unit: "评",},
        { type: "bit", icon:
"http://i.muзtak.cn/images/icon-heart.svg", text: 284, label: "共", unit: "赞",
emit: "like", },
      ]
    ]
  },
  {
    list:[
      { type: "five-star", text: 4.5, },
      { type: "time", icon:
"http://i.muзtak.cn/images/icon-time-color.svg", text: 18357, mode:
"timestamp",},
    ]
  },
  {
    list:[
      [
        [
          { type: "button", text: '电话', color:
'indigo', emit: 'call'},
        ]
      ]
    ]
  }
]

```



```

//薛定谔式
schrodinger: {
  list: [
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      burn: false, //阅后即焚
      auth: false, //授权查看
      text: '公开照片', //是否授权
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      burn: true, //阅后即焚
      burned: false, //是否焚毁
      text: '阅后即焚', //是否授权
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      burn: true, //阅后即焚
      burned: true, //是否焚毁
      text: '已焚毁', //是否授权
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      auth: true, //授权查看
      authed: false, //是否授权
      text: '红包', //是否授权
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      auth: true, //授权查看
      authed: true, //是否授权
      text: '已支付', //是否授权
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      burn: true, //阅后即焚
      auth: true, //授权查看
      burned: false, //是否焚毁
      authed: false, //是否授权
      text: '红包阅后即焚', //是否授权
      icon: 'http://i.muztak.cn/images/icon-house.svg', //

      price: 12, // 价格
      unit: '币', // 单位
    },
    {
      image: "http://i.muztak.cn/images/avatar-live.jpg",
      burn: true, //阅后即焚
      auth: true, //授权查看
      burned: true, //是否焚毁
      authed: true, //是否授权
      text: '已焚毁', //是否授权
    },
  ],
},
//胶卷式
kodak: {
  list: [

```

不写单位也可以放图标

```

        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
        'http://i.muзtak.cn/images/yellow-river.jpg',
    ],
    },
    alltext: {
        text: "    今天“<b>节日活动拼团</b>”的，可以过来看看看。
<p>Расцветали яблони и груши, Поплыли туманы над рекой. </p><p>Выходила на берег
катюша, На высокий берег на крутой. Выходила на берег катюша, На высокий берег на
крутой. Выходила на берег катюша, На высокий берег на крутой. Выходила на берег
катюша, На высокий берег на крутой. </p>"
    },
    bars: [
        {
            list: [
                [
                    { type: "tags", list: ['同城', '圣诞节'], },
                ],
                [
                    { type: "bit", icon_code: "M", text: 75, unit:
"评", },
                    { type: "bit", icon_code: "L", text: 284, unit:
"赞", },
                ]
            ]
        },
        {
            list: [
                [
                    { type: "button", text: '加微信', color: 'green',
emit: 'call'},
                ]
            ]
        },
    ],
    description: "今天节日活动拼团的，可以过来看看看。Сегодня хороший
день.",
    introduce: "今天节日活动拼团的，可以过来看看看。Расцветали яблони и
груши, Поплыли туманы над рекой. Выходила на берег катюша, На высокий берег на
крутой.",
    },
    })
}
</script>

```

Muz-Item Attributes

参数	说明	类型	可选值	默认值
item	条数据配置	object	—	—

Option of Item

注意：文本统一键名text!!!，链接统一键名url!!!

参数	说明	类型	可选值	默认值
title	=== 标题区配置 ===	object	—	—
title.text 原: title.title	大标题	string	—	—
title.hot_tip	标题热点提示 (右侧红条)	string	"热门"	—
title.styles	自定义样式	object	{color: red}	—
description	=== 描述文字 ===	string	—	—
introduce 没有呢	=== 介绍文字 ===	string	—	—
summary	=== 摘要配置 ===	object	—	—
summary.text	摘要文字	string	—	—
description	=== 描述配置 ===	object	—	—
description.text	描述文字	string	—	—
alltext	=== 全文区配置 ===	object	—	—
alltext.text	全部文字	string	—	—
lead	=== 领头区配置 ===	object	—	—
lead.cover.url	封面地址 (矩形)	string	—	—
lead.cover.hot_tip	封面热点提示 (右上三角)	string	"棒"	—
lead.cover.aspect	封面外形	string	square, pinkie	rectangle
lead.avatar.url	作者头像 (中圆)	string	—	—
lead.avatar.aspect	尺寸样式	string	large, medium, small, mini	big
lead.bars....	横条, 见bars	object	—	—
bars	=== 横条区配置 === 详见: Option of Bars	array	—	—
cover	=== 封面配置 ===	object	—	—
cover.url	封面地址	string	—	—
cover.text	封面文字	string	—	—
cover.aspect	封面外形	string	square	rectangle
cover.emit	封面信号	string	—	—
video	=== 视频区配置 ===	object	—	—

参数	说明	类型	可选值	默认值
video.text 原: video.title	视频标题	string	—	—
video.cover	视频封面	string	—	—
video.url	视频地址。支持mp4,avi	string	—	—
video.danmuku	视频弹幕。只是个想法, 还没实现.....	string	[[{text: "红色走一波", timestamp: 74848},]	—
swiper	=== 轮播区配置 ===	object	—	—
swiper.emit	轮播图信号	string	—	—
swiper.autoplay	轮播图自动播放	boolean	true	false
swiper.mode	轮播图图片模式	string	aspectFill	aspectFit, scaleToFill 等14种
swiper.styles	轮播图样式	object	{color: 'red'}	—
swiper.list	轮播图列表 详见: Option of Swiper	array	—	—
isPreview	是否支持预览	boolean	true	false
sudoku	=== 九图区配置 ===	object	—	—
swiper.list	图片地址数组	array	例: ['/images/1.jpg', ...]	—
schrodinger	=== 薛定谔图配置 ===	object	—	—
schrodinger.list	图片列表	array	—	—
kodak	=== 柯达胶卷图配置 ===	object	—	—
kodak.list	图片地址数组	array	例: ['/images/1.jpg', ...]	—
emit	=== 信号配置 ===	object, string	—	—
emit.emit	自定义信号名	string	—	—

参数	说明	类型	可选值	默认值
emit.url 支持: muz-area-list	信号触发后的跳转地址。 注意: 需配合muz-area-list组件使用, 原因是考虑到muz-item组件的闭合性原则, 所以只在更高级的组件中使用“模版字符串”等高级功能。	string	—	—
config.cancelBubble	取消冒泡事件 即取消"item"事件	boolean	false	true
	=== 自定义键配置 ===	string	—	—
.slot	自定义槽名	string	—	—

Option of Bars in Item

参数	说明	类型	可选值	默认值
type	数据类型	string	bit, title, author, rank, summary, description tags, five-star, time, price, button	bit
text	显示文本	string, number	—	—
text	评星星级 适用: five-star	number	0.0~5.0 (step: 0.5)	—
hot_tip	热点提示 适用: title	string	"棒"	—
icon	图标地址 适用: author, rank, price, bit, time	string	—	—
icon_code	图标字体代码 适用: author, rank, price, bit	string	"r"卢布, "M"评论, "L"爱心	—
label	标签 (前辍) 适用: bit	string	—	—
unit	单位 (后辍) 适用: price, bit	string	"卢布"	—
zero	价格为空或零时的占位符 适用: price	string	—	—
emit	信号名, 事件 适用: bit, title, author, 暂不支持: price, button, tags	string	—	—
list	多个标签 适用: tags	string array array	"同城;圣诞节" 或 ['同城', '圣诞节'] 或 [{tag: '同城', color: 'red'}]	—
split	分割符 适用: tags	string	—	","

参数	说明	类型	可选值	默认值
mode	模式。设定text值是时间戳或文本 适用: time	string	timestamp	date
format	格式化。 适用: time	string	yyyy-mm-dd HH:ii:ss.fff	yyyy-mm-dd HH:ii:ss
color	按钮 适用: bit, tags, button	string	indigo, red, orange, green, blue, purple	
list	横条区一行数据。 1个元素左对齐, 2个元素两端对齐, 3+个元素两头分散对齐, 选项配置与Bars相同。	array 或 array<array>		

Option of Swiper's List in Item

参数	说明	类型	可选值	默认值
url	轮播图地址	string	—	—
text 原: title	轮播图文本	string	—	—

Option of Schrodinger's List in Item

参数	说明	类型	可选值	默认值
image	图片地址	string	—	—
burn	是否阅后即焚	boolean	true	false
burned	是否已焚毁	boolean	true	false
auth	是否授权查看	boolean	true	false
authed	是否已授权	boolean	true	false
text	图上文字	string	—	—
icon	价格前的图标	string	—	—
price	图片价格	number	—	—
unit	价格后的单位	string	—	—

Event

事件名称	说明	回调参数
item	点击条数据任一位置触发	item - 条数据配置 (结构配置)
emit	自定义事件触发 Todo仅实现了根信号, 各元素待开发	{emit, config, node} - 信号名, 信号配置, 触发节点

List-Area 列表区渲染

该组件基于muz-item进行列表区域渲染, 自动请求相关数据。

基本用法

```
<template>
  <muz-list-area :config="it.config" :request="request" :options-
request="optionsRequest" :page="request.page" :limit="request.limit" :item-
template="itemTemplate" :item-function="itemFunction" />
</template>

<script>
  export default {
    components: {
      MuzItem: () => 'muz-item', // 注: 微信小程序不支持此组件懒加载方式
    },
    data: () => ({
      config: {

      },
      request: {
        url: '/api/news_list',
        rootUrl: 'http://xx.cn',
        method: 'GET',
        param: {},
        headers: {},
        page: 1,
        limit: 10,
        showMore: false,
        nodeStatus: 'err',
        nodeStatusValue: 0,
        nodeData: 'data.list',
      },
    })
  }
</script>
```

List-Area Attributes

参数	说明	类型	可选值	默认值
page	当前页数 注：哆啦A梦需在request配置	number	—	1
limit	每页限制条数 注：哆啦A梦需在request配置	number	—	10
config	列表区配置	object	—	—
page-param	传入页面参数，用于定制化不同页面过滤结果	object	—	—
item-template	条数据模版。 详见：Item 条数据	object	—	—
item-template-set	条数据模版集合。 item-template的集合，常用配置多展示样式	object	{news: ...}	—
item-function	条数据函数。 遍历每条数据时调用该函数序列化为muz-item配置。	function	function(item) { return {title: xxx} }	—
request	组件数据请求配置 详见：Option of Request	object, function 待加	—	—
options-request	多组选项的请求配置 详见：Option of Options-Request	array	—	—
item-templates-request	条数据模版集请求配置 详见：Option of Item-Templates-Request	array	—	—

Option of Config

参数	说明	类型	可选值	默认值
onClickUrl	点击跳转地址	string	—	—
nodeTemplate 适用: item-template-set, item-templates-request	条数据展示模板的节点键名	string	—	show
debug	开启调试, 控制台输出序列化结果	boolean	true	false
emits	信号触发后的配置集	object<emit, object>	{click: {url: ...}}	—

Option of Request & Options-Request & Item-Templates-Request

参数	说明	类型	可选值	默认值
rootUrl	根域名或IP地址。 建议：结尾不要加"/"。在App、小程序等非代理模式时可用。	string	—	—
url	请求地址。 建议：省略域名时应以"/"开头，与rootUrl拼接完整地址。	string	—	—
method	请求方法	string	GET, POST, PUT, DELETE	GET
param	请求参数	object	—	—
headers	自定义请求头。 注意：1、Web端GET为 <code>Content-Type: ''</code> ，即删除键 小程序端GET为 <code>Content-Type: application/text</code> 所有POST为 <code>Content-Type: application/json</code> 2、Web端为 <code>authorization</code> ，小程序端为 <code>Authorization</code>	object	例：{ authorization: xxx }	—
showMore	是否显示更多	boolean	false	true
page 适用：Request选项	当前页数	number	—	1
limit 适用：Request选项	每页限制条数	number	—	10
options_key 适用：Options-Request选项	指定创建的数据选项节点键名 注：不要与条数据已存在的键名冲突，该值将在遍历序列化时追添至条数据。	options_key	例： category_options	—
templateName 适用：Item-Templates-Request	自定义模块展示名称（节点值）	string	例：news	—
nodeStatus	响应的状态码节点键名	string	—	err

参数	说明	类型	可选值	默认值
nodeStatusValue	响应的状态码成功的键值	string	—	0
nodeData	响应的结果数据节点键名	string	例: data.list	data
nodeTemplate 删除!!! 适用: Item- Templates- Request	条数据展示模板的节点键名 20230325删除, 移到主 config中了	string	—	show
needToken	是否需要令牌 (Token) 仅支持Bearer令牌	boolean	true	false
tokenKey	从缓存中获取Token令牌的 键名	string	—	token

注: node节点键名支持对象操作符"."点、支持数组操作符"[]", 支持多层子树。"

Option of Emits in Config

参数	说明	类型	可选值	默认值
url	跳转地址	string	—	—

Event

事件名称	说明	回调参数
item	点击条数据任一位置触发	item - 当前条数据参数 注: 这点与muz-item不同
emit	条数据的自定义事件触发。 注意: 自定义事件默认会关闭条数据的"item"冒泡事件, 并使条数据的onClickUrl属性失效, 此时需要通过配置条数据的config.url属性来设置新url。如果希望恢复冒泡事件, 设置条数据配置cancelBubble属性为false, 那时onClickUrl和新url都会生效, 当然这种需求十分少见。	({emit, config, node}, param) {信号名, 信号配置, 触发节点}, 当前条数据参数

Item-Area 条数据区渲染

该组件基于muz-item进行条数据区域渲染，自动请求相关数据。

基本用法

```
<template>
  <muz-item-area :config="it.config" :request="concatObject(it.request,
'param', this.pageParam)" :options-request="it.optionsRequest" :item-
template="it.item" />
</template>

<script>
  export default {
    components: {
      MuzItemArea: () => 'muz-item-area', // 注：微信小程序不支持此组件懒加载方
式
    },
    data: () => ({
      // 与muz-list-area相同
      ...
    })
  }
</script>
```

Item-Area Attributes

参数	说明	类型	可选值	默认值
config	条数据配置	object	—	—
page	当前页数 注：哆啦A梦需在request选项	number	—	1
limit	每页限制条数 注：哆啦A梦需在request选项	number	—	10
request	组件数据请求配置 详见：List-Area 列表区渲染	object	—	—
options-request	多组选项的请求配置 详见：List-Area 列表区渲染	array	—	—
item-template 或item	条数据模版。 详见：Item 条数据	object	—	—

Chat-List 聊天列表

聊天记录列表, 待开发

- 基本使用

```
<muz-chat-list :list="list" />

<script>
export default{
  data: () => ({
    list: [
      {user_id:01 ,author/nickname: '想摆烂',avatar:'(http://i.cn/1111.jpg)'
      ,mode: 'mine', type: 'text', content: '每天为这几个臭钱而奔波', Url:'',timestamp:
      1651212271},

      {user_id:02 ,author/nickname: '海琴烟',avatar:'(http://i.cn/1111.jpg)'
      ,mode: 'their', type: 'video', content: '我饿了!', Url:'[]
      (http://i.cn/1111.mp4)',timestamp: 1651212282},

      {user_id:01 ,author/nickname: '想摆烂',avatar:'(http://i.cn/1111.jpg)'
      ,mode: 'their', type: 'image', content: '我饿了!',Url:'[](http://i.cn/1111.jpg)'
      ,timestamp: 1651212290},

      {user_id:02 ,author/nickname: '海琴烟',avatar:'(http://i.cn/1111.jpg)'
      ,mode: 'mine', type: 'sound', content: '饿, 巨饿',Url:'http://i.cn/1111.mp3'
      ,timestamp: 1651215871},

    ],
  })
}
</script>
```

- list属性

参数	说明	类型	默认值	可选值
user_id	用户唯一标识	number	-	-
author/nickname	用户发送的消息	string	-	-
avatar	用户头像	string	-	-
mode	自己/对方	sting	-	mine/their
type	用户输入的类型	string	-	text, video, image, sound

- mode 属性

参数	说明	类型	默认值	可选值
mine	自己	string		-
their	对方	string		-

- type 属性

参数	说明	类型	默认值	可选值
text	纯文本	string	-	-
video	视频	string	-	-
image	图片	string	-	-
sound	语音	string	-	-

- Props 属性

参数	说明	类型	默认值	可选值
list	数据表头	Array[Object, ...]	见下表	见下表
formData	生成key-value对象。需sync	Object	{headers.key:headers.value}	/
onlyRead	表单只读	Boolean	false	true

- headers 属性中的字段数据(Item)参数

键	说明	类型	必选	默认值	可选值
key	键名	String	是	/	例: "sex"
label	标签文字	String	否	/	例: "姓名"
value	值	String Array	否	/	/

```
/**
 * function xTemplateString::format 对模板字符串进行编译，格式化为结果字符串
 * @param {object} item 给定需要格式化的Key-value对象。
 * @param {string} text 要格式化的字符串文本
 * @desc
 * @return {string} 格式化完成后字符串
 * @example
 * 模板语法糖:
```

```

* 一、基础版——直接引用值。语法： {{key}}
* 例子：
*   param={category_id: 35, category_value: "杭州西湖区"};
*   str="类别ID: {{category_id}}, 类别值: {{category_value}}。";
*   XTemplateString.format(param, str);
*   结果 ==> "类别ID: 35, 类别值: 杭州西湖区。"
* 二、高级版——管道选项值。即指定列表对象、标签键、值键，找到引用值。语法：
*   {{valueKey:key|options[labelKey]}}
* 例子：
*   param={category_id: 35, category_list: [{value: 27, label: "杭州上城区"},
*   {value: 35, label: "杭州西湖区"}]};
*   str="类别: {{ value: category_id | category_list[label] }}"
*   XTemplateString.format(param, str);
*   结果 ==> "类别: 杭州西湖区。"
* 三、高级版——强制转换值。即指定来源键、目标键，生成数组对象。语法：
*   {{(Array<targetKey:sourceKey>)valueKey}}
* 例子：
*   param={photo_list: [{link: '/images/a.jpg', label: "杭州上城区"}, {link:
*   '/images/b.jpg', label: "杭州西湖区"}]};
*   str="{{ (Array< url: link, text: label >)photo_list }}"
*   str2="{{ (Array< :link >)photo_list }}"
*   XTemplateString.format(param, str);
*   结果 ==> "[{url: '/images/a.jpg', text: "杭州上城区"}, {url:
*   '/images/b.jpg', text: "杭州西湖区"}]"
*   XTemplateString.format(param, str2);
*   结果 ==> "['/images/a.jpg', '/images/b.jpg']"
*/

```

版本说明

V3.0.1.20230302-alpha

正式发布支持Vue3版本的Muz-Doraemon组件。

参考文献

[uni-list](#) 组件

app端nvue专用组件。在app-nvue下，如果是长列表，使用list组件的性能高于使用view或scroll-view的滚动。原因在于list在不可见部分的渲染资源回收有特殊的优化处理。

原生渲染的资源回收机制，与webview渲染不同。webview不需要数据有规则格式，长页面处于不可见的部分，其渲染资源会自动回收，除非webview使用区域滚动而不是页面滚动。所以vue页面只要不用scroll-view，就不需要关注这个问题。而原生渲染则必须注意。

如果需要跨端，建议使用uni-ui的uni-list组件，它会自动处理webview渲染和原生渲染的情况，在app-nvue下使用list组件，而在其他平台则使用页面滚动。详见<https://ext.dcloud.net.cn/plugin?id=24>

