

Tabs 标签页

分隔内容上有关联但属于不同类别的数据集合。

基础用法

基础的、简洁的标签页。

:::demo Tabs 组件提供了选项卡功能，默认选中第一个标签页，你也可以通过 `value` 属性来指定当前选中的标签页。

```
<template>
  <el-tabs v-model="activeName" @tab-click="handleClick">
    <el-tab-pane label="用户管理" name="first">用户管理</el-tab-pane>
    <el-tab-pane label="配置管理" name="second">配置管理</el-tab-pane>
    <el-tab-pane label="角色管理" name="third">角色管理</el-tab-pane>
    <el-tab-pane label="定时任务补偿" name="fourth">定时任务补偿</el-tab-pane>
  </el-tabs>
</template>
<script>
  export default {
    data() {
      return {
        activeName: 'second'
      };
    },
    methods: {
      handleClick(tab, event) {
        console.log(tab, event);
      }
    }
  };
</script>
```

...

选项卡样式

选项卡样式的标签页。

:::demo 只需要设置 `type` 属性为 `card` 就可以使选项卡改变为标签风格。

```
<template>
  <el-tabs v-model="activeName" type="card" @tab-click="handleClick">
    <el-tab-pane label="用户管理" name="first">用户管理</el-tab-pane>
    <el-tab-pane label="配置管理" name="second">配置管理</el-tab-pane>
    <el-tab-pane label="角色管理" name="third">角色管理</el-tab-pane>
    <el-tab-pane label="定时任务补偿" name="fourth">定时任务补偿</el-tab-pane>
  </el-tabs>
</template>
<script>
  export default {
```

```

    data() {
      return {
        activeName: 'first'
      };
    },
    methods: {
      handleClick(tab, event) {
        console.log(tab, event);
      }
    }
  };
</script>

```

...

卡片化

卡片化的标签页。

::demo 将 type 设置为 border-card 。

```

<el-tabs type="border-card">
  <el-tab-pane label="用户管理">用户管理</el-tab-pane>
  <el-tab-pane label="配置管理">配置管理</el-tab-pane>
  <el-tab-pane label="角色管理">角色管理</el-tab-pane>
  <el-tab-pane label="定时任务补偿">定时任务补偿</el-tab-pane>
</el-tabs>

```

...

位置

可以通过 tab-position 设置标签的位置

::demo 标签一共有四个方向的设置 tabPosition="left|right|top|bottom"

```

<template>
  <el-radio-group v-model="tabPosition" style="margin-bottom: 30px;">
    <el-radio-button label="top">top</el-radio-button>
    <el-radio-button label="right">right</el-radio-button>
    <el-radio-button label="bottom">bottom</el-radio-button>
    <el-radio-button label="left">left</el-radio-button>
  </el-radio-group>

  <el-tabs :tab-position="tabPosition" style="height: 200px;">
    <el-tab-pane label="用户管理">用户管理</el-tab-pane>
    <el-tab-pane label="配置管理">配置管理</el-tab-pane>
    <el-tab-pane label="角色管理">角色管理</el-tab-pane>
    <el-tab-pane label="定时任务补偿">定时任务补偿</el-tab-pane>
  </el-tabs>
</template>
<script>

```

```

export default {
  data() {
    return {
      tabPosition: 'left'
    };
  }
};
</script>

```

...

自定义标签页

可以通过具名 `slot` 来实现自定义标签页的内容

:::demo

```

<el-tabs type="border-card">
  <el-tab-pane>
    <span slot="label"><i class="el-icon-date"></i> 我的行程</span>
    我的行程
  </el-tab-pane>
  <el-tab-pane label="消息中心">消息中心</el-tab-pane>
  <el-tab-pane label="角色管理">角色管理</el-tab-pane>
  <el-tab-pane label="定时任务补偿">定时任务补偿</el-tab-pane>
</el-tabs>

```

...

动态增减标签页

增减标签页按钮只能在选项卡样式的标签页下使用

:::demo

```

<el-tabs v-model="editableTabsValue" type="card" editable @edit="handleTabsEdit">
  <el-tab-pane
    :key="item.name"
    v-for="(item, index) in editableTabs"
    :label="item.title"
    :name="item.name"
  >
    {{item.content}}
  </el-tab-pane>
</el-tabs>
<script>
export default {
  data() {
    return {
      editableTabsValue: '2',
      editableTabs: [{
        title: 'Tab 1',

```

```

        name: '1',
        content: 'Tab 1 content'
      }, {
        title: 'Tab 2',
        name: '2',
        content: 'Tab 2 content'
      }
    ],
    tabIndex: 2
  }
},
methods: {
  handleTabsEdit(targetName, action) {
    if (action === 'add') {
      let newTabName = ++this.tabIndex + '';
      this.editableTabs.push({
        title: 'New Tab',
        name: newTabName,
        content: 'New Tab content'
      });
      this.editableTabsValue = newTabName;
    }
    if (action === 'remove') {
      let tabs = this.editableTabs;
      let activeName = this.editableTabsValue;
      if (activeName === targetName) {
        tabs.forEach((tab, index) => {
          if (tab.name === targetName) {
            let nextTab = tabs[index + 1] || tabs[index - 1];
            if (nextTab) {
              activeName = nextTab.name;
            }
          }
        });
      }
      this.editableTabsValue = activeName;
      this.editableTabs = tabs.filter(tab => tab.name !== targetName);
    }
  }
}
}
}
</script>

```

...

自定义增加标签页触发器

:::demo

```

<div style="margin-bottom: 20px;">
  <el-button
    size="small"

```

```

    @click="addTab(editableTabsValue)"
  >
    add tab
  </el-button>
</div>
<el-tabs v-model="editableTabsValue" type="card" closable @tab-remove="removeTab">
  <el-tab-pane
    v-for="(item, index) in editableTabs"
    :key="item.name"
    :label="item.title"
    :name="item.name"
  >
    {{item.content}}
  </el-tab-pane>
</el-tabs>
<script>
  export default {
    data() {
      return {
        editableTabsValue: '2',
        editableTabs: [{
          title: 'Tab 1',
          name: '1',
          content: 'Tab 1 content'
        }, {
          title: 'Tab 2',
          name: '2',
          content: 'Tab 2 content'
        }],
        tabIndex: 2
      }
    },
    methods: {
      addTab(targetName) {
        let newTabName = ++this.tabIndex + '';
        this.editableTabs.push({
          title: 'New Tab',
          name: newTabName,
          content: 'New Tab content'
        });
        this.editableTabsValue = newTabName;
      },
      removeTab(targetName) {
        let tabs = this.editableTabs;
        let activeName = this.editableTabsValue;
        if (activeName === targetName) {
          tabs.forEach((tab, index) => {
            if (tab.name === targetName) {
              let nextTab = tabs[index + 1] || tabs[index - 1];
              if (nextTab) {
                activeName = nextTab.name;
              }
            }
          });
        }
      }
    }
  }

```

```
        }
      });
    }

    this.editableTabsValue = activeName;
    this.editableTabs = tabs.filter(tab => tab.name !== targetName);
  }
}
}
}
</script>
```

...

Tabs Attributes

参数	说明	类型	可选值	默认值
value / v-model	绑定值，选中选项卡的 name	string	—	第一个选项卡的 name
type	风格类型	string	card/border-card	—
closable	标签是否可关闭	boolean	—	false
addable	标签是否可增加	boolean	—	false
editable	标签是否同时可增加和关闭	boolean	—	false
tab-position	选项卡所在位置	string	top/right/bottom/left	top
stretch	标签的宽度是否自撑开	boolean	-	false
before-leave	切换标签之前的钩子，若返回 false 或者返回 Promise 且被 reject，则阻止切换。	Function(activeName, oldActiveName)	—	—

Tabs Events

事件名称	说明	回调参数
tab-click	tab 被选中时触发	被选中的标签 tab 实例
tab-remove	点击 tab 移除按钮后触发	被删除的标签的 name
tab-add	点击 tabs 的新增按钮后触发	—
edit	点击 tabs 的新增按钮或 tab 被关闭后触发	(targetName, action)

Tab-pane Attributes

--	--	--	--	--

参数	说明	类型	可选值	默认值
label	选项卡标题	string	—	—
disabled	是否禁用	boolean	—	false
name	与选项卡绑定值 value 对应的标识符，表示选项卡别名	string	—	该选项卡在选项卡列表中的顺序值，如第一个选项卡则为'1'
closable	标签是否可关闭	boolean	—	false
lazy	标签是否延迟渲染	boolean	—	false