

Tabs

Divise des collections de données de types différents, mais reliées entre elles par un contexte ou un type global.

Usage

Onglets basiques.

:::demo Tabs fournit un outil de sélection des onglets. Par défaut, le premier onglet est sélectionné et actif mais vous pouvez activer n'importe lequel en réglant l'attribut `value` .

```
<template>
  <el-tabs v-model="activeName" @tab-click="handleClick">
    <el-tab-pane label="User" name="first">Utilisateur</el-tab-pane>
    <el-tab-pane label="Config" name="second">Config</el-tab-pane>
    <el-tab-pane label="Role" name="third">Rôle</el-tab-pane>
    <el-tab-pane label="Task" name="fourth">Tâche</el-tab-pane>
  </el-tabs>
</template>
<script>
  export default {
    data() {
      return {
        activeName: 'first'
      };
    },
    methods: {
      handleClick(tab, event) {
        console.log(tab, event);
      }
    }
  };
</script>
```

...

Style carte

Les onglets peuvent être stylisés comme des cartes.

:::demo Mettez `type` à `card` pour avoir des onglets avec un style de carte.

```
<template>
  <el-tabs type="card" @tab-click="handleClick">
    <el-tab-pane label="User">Utilisateur</el-tab-pane>
    <el-tab-pane label="Config">Config</el-tab-pane>
    <el-tab-pane label="Role">Rôle</el-tab-pane>
    <el-tab-pane label="Task">Tâche</el-tab-pane>
  </el-tabs>
</template>
<script>
  export default {
```

```

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

```

⋮

Style carte avec bordure

Onglets avec style de carte et bordure.

⋮demo Mettez type à border-card .

```

<el-tabs type="border-card">
  <el-tab-pane label="User">Utilisateur</el-tab-pane>
  <el-tab-pane label="Config">Config</el-tab-pane>
  <el-tab-pane label="Role">Rôle</el-tab-pane>
  <el-tab-pane label="Task">Tâche</el-tab-pane>
</el-tabs>

```

⋮

Position des onglets

Vous pouvez utiliser tab-position pour régler la position des onglets.

⋮demo Il y a quatre positions différentes: tabPosition="left|right|top|bottom"

```

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

  <el-tabs :tab-position="tabPosition" style="height: 200px;">
    <el-tab-pane label="User">Utilisateur</el-tab-pane>
    <el-tab-pane label="Config">Config</el-tab-pane>
    <el-tab-pane label="Role">Rôle</el-tab-pane>
    <el-tab-pane label="Task">Tâche</el-tab-pane>
  </el-tabs>
</template>
<script>

```

```

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

```

...

Onglet personnalisé

Vous pouvez utiliser un slot pour customiser le label d'un onglet.

:::demo

```

<el-tabs type="border-card">
  <el-tab-pane>
    <span slot="label"><i class="el-icon-date"></i> Route</span>
    Route
  </el-tab-pane>
  <el-tab-pane label="Config">Config</el-tab-pane>
  <el-tab-pane label="Role">Rôle</el-tab-pane>
  <el-tab-pane label="Task">Tâche</el-tab-pane>
</el-tabs>

```

...

Ajouter et supprimer des onglets

Seuls les onglets de type carte supportent l'ajout et la suppression.

:::demo

```

<el-tabs v-model="editableTabsValue" type="card" editable @edit="handleTabsEdit">
  <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: 'Onglet 1',

```

```

        name: '1',
        content: 'Contenu de 1\'onglet 1'
      }, {
        title: 'Onglet 2',
        name: '2',
        content: 'Contenu de 1\'onglet 2'
      }
    ],
    tabIndex: 2
  }
},
methods: {
  handleTabsEdit(targetName, action) {
    if (action === 'add') {
      let newTabName = ++this.tabIndex + '';
      this.editableTabs.push({
        title: 'Nouvel onglet',
        name: newTabName,
        content: 'Contenu du nouvel onglet'
      });
      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>

```

...

Personnaliser le bouton d'ajout d'onglet

:::demo

```

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

```

```

    @click="addTab(editableTabsValue)"
  >
    Ajouter un onglet
  </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: 'Onglet 1',
          name: '1',
          content: 'Contenu de l\'onglet 1'
        }, {
          title: 'Onglet 2',
          name: '2',
          content: 'Contenu de l\'onglet 2'
        }],
        tabIndex: 2
      }
    },
    methods: {
      addTab(targetName) {
        let newTabName = ++this.tabIndex + '';
        this.editableTabs.push({
          title: 'Nouvel onglet',
          name: newTabName,
          content: 'Contenu du nouvel onglet'
        });
        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>

```

...

Attributs de Tabs

Attribut	Description	Type	Valeurs acceptées	Défaut
value / v-model	La valeur liée, nom de l'onglet sélectionné.	string	—	Nom du premier onglet.
type	Type de l'onglet.	string	card/border-card	—
closable	Si des onglets peuvent être supprimés.	boolean	—	false
addable	Si des onglets peuvent être ajoutés.	boolean	—	false
editable	Si des onglets peuvent être ajoutés et supprimés.	boolean	—	false
tab-position	Position des onglets.	string	top/right/bottom/left	top
stretch	Si la largeur des onglets s'adapte au conteneur.	boolean	-	false
before-leave	Fonction de hook avant de changer d'onglet. Si <code>false</code> est retourné ou qu'une <code>Promise</code> retournée et rejetée, le changement sera annulé.	Function(activeName, oldActiveName)	—	—

Évènements de Tabs

Nom	Description	Paramètres
tab-click	Se déclenche quand on clique sur un onglet.	clicked tab
tab-remove	Se déclenche quand on clique sur le bouton de suppression des onglets.	name of the removed tab
tab-add	Se déclenche quand on clique sur le bouton d'ajout des onglets.	—

edit	Se déclenche quand on clique sur les boutons d'ajout ou de suppression des onglets.	(targetName, action)
------	---	----------------------

Attributs de Tab-pane

Attribut	Description	Type	Valeurs acceptées	Défaut
label	Titre de l'onglet.	string	—	—
disabled	Si l'onglet est désactivé.	boolean	—	false
name	Identifiant correspondant au nom des onglets, utilisé par Tabs pour savoir quel est l'onglet actif.	string	—	Numéro de l'onglet dans l'ordre d'apparition, e.g. le premier est '1'.
closable	Si l'onglet est supprimable.	boolean	—	false
lazy	Si le contenu de l'onglet bénéficie du lazy-loading.	boolean	—	false