Cascader

If the options have a clear hierarchical structure, Cascader can be used to view and select them.

Basic usage

There are two ways to expand child option items.

:::demo Assigning the `options` attribute to an array of options renders a Cascader. The `props.expandTrigger` attribute defines how child options are expanded.

```
<div class="block">
  <span class="demonstration">Child options expand when clicked (default)</span>
  <el-cascader
    v-model="value"
    :options="options"
    @change="handleChange"></el-cascader>
</div>
<div class="block">
  <span class="demonstration">Child options expand when hovered</span>
  <el-cascader
    v-model="value"
    :options="options"
    :props="{ expandTrigger: 'hover' }"
    @change="handleChange"></el-cascader>
</div>

<script>
  export default {
    data() {
      return {
        value: [],
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
```

```
      value: 'navigation',
      label: 'Navigation',
      children: [{
        value: 'side nav',
        label: 'Side Navigation'
      }, {
        value: 'top nav',
        label: 'Top Navigation'
      }]
    }]
  }, {
    value: 'component',
    label: 'Component',
    children: [{
      value: 'basic',
      label: 'Basic',
      children: [{
        value: 'layout',
        label: 'Layout'
      }, {
        value: 'color',
        label: 'Color'
      }, {
        value: 'typography',
        label: 'Typography'
      }, {
        value: 'icon',
        label: 'Icon'
      }, {
        value: 'button',
        label: 'Button'
      }]
    }, {
      value: 'form',
      label: 'Form',
      children: [{
        value: 'radio',
        label: 'Radio'
      }, {
        value: 'checkbox',
        label: 'Checkbox'
      }, {
        value: 'input',
        label: 'Input'
      }, {
        value: 'input-number',
        label: 'InputNumber'
      }, {
        value: 'select',
        label: 'Select'
      }, {
        value: 'cascader',
        label: 'Cascader'
      }, {
        value: 'switch',
        label: 'Switch'
      }, {
        value: 'slider',
        label: 'Slider'
      }, {
        value: 'time-picker',
        label: 'TimePicker'
      }, {
        value: 'date-picker',
        label: 'DatePicker'
      }, {
        value: 'datetime-picker',
        label: 'DateTimePicker'
      }, {
        value: 'upload',
        label: 'Upload'
      }, {
        value: 'rate',
        label: 'Rate'
      }, {
        value: 'form',
        label: 'Form'
      }]
    }, {
      value: 'data',
      label: 'Data',
      children: [{
        value: 'table',
        label: 'Table'
      }, {
        value: 'tag',
        label: 'Tag'
      }, {
        value: 'progress',
        label: 'Progress'
      }, {
        value: 'tree',
        label: 'Tree'
      }, {
        value: 'pagination',
        label: 'Pagination'
      }, {
        value: 'badge',
        label: 'Badge'
      }]
    }, {
      value: 'notice',
      label: 'Notice',
      children: [{
        value: 'alert',
        label: 'Alert'
      }, {
        value: 'loading',
        label: 'Loading'
      }, {
        value: 'message',
        label: 'Message'
      }, {
        value: 'message-box',
        label: 'MessageBox'
      }, {
        value: 'notification',
        label: 'Notification'
      }]
    }, {
      value: 'navigation',
      label: 'Navigation',
      children: [{
        value: 'menu',
        label: 'NavMenu'
      }, {
        value: 'tabs',
        label: 'Tabs'
      }, {
        value: 'breadcrumb',
        label: 'Breadcrumb'
      }, {
        value: 'dropdown',
        label: 'Dropdown'
      }, {
        value: 'steps',
        label: 'Steps'
      }]
    }, {
      value: 'others',
      label: 'Others',
      children: [{
        value: 'dialog',
        label: 'Dialog'
      }, {
        value: 'tooltip',
        label: 'Tooltip'
      }, {
        value: 'popover',
        label: 'Popover'
      }, {
        value: 'card',
        label: 'Card'
      }, {
        value: 'carousel',
        label: 'Carousel'
```

```
            }, {
              value: 'collapse',
              label: 'Collapse'
            }]
          }]
        }, {
          value: 'resource',
          label: 'Resource',
          children: [{
            value: 'axure',
            label: 'Axure Components'
          }, {
            value: 'sketch',
            label: 'Sketch Templates'
          }, {
            value: 'docs',
            label: 'Design Documentation'
          }]
        }]
      };
    },
    methods: {
      handleChange(value) {
        console.log(value);
      }
    }
  };
</script>
```

:::

Disabled option

Disable an option by setting a `disabled` field in the option object.

:::demo In this example, the first item in `options` array has a `disabled: true` field, so it is disabled. By default, Cascader checks the `disabled` field in each option object; if you are using another field name to indicate whether an option is disabled, you can assign it in the `props.disabled` attribute (see the API table below for details). And of course, field name `value`, `label` and `children` can also be customized in the same way.

```
<el-cascader :options="options"></el-cascader>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          disabled: true,
          children: [{
```

```
        value: 'disciplines',
        label: 'Disciplines',
        children: [{
          value: 'consistency',
          label: 'Consistency'
        }, {
          value: 'feedback',
          label: 'Feedback'
        }, {
          value: 'efficiency',
          label: 'Efficiency'
        }, {
          value: 'controllability',
          label: 'Controllability'
        }]
      }, {
        value: 'navigation',
        label: 'Navigation',
        children: [{
          value: 'side nav',
          label: 'Side Navigation'
        }, {
          value: 'top nav',
          label: 'Top Navigation'
        }]
      }]
    }, {
      value: 'component',
      label: 'Component',
      children: [{
        value: 'basic',
        label: 'Basic',
        children: [{
          value: 'layout',
          label: 'Layout'
        }, {
          value: 'color',
          label: 'Color'
        }, {
          value: 'typography',
          label: 'Typography'
        }, {
          value: 'icon',
          label: 'Icon'
        }, {
          value: 'button',
          label: 'Button'
        }]
      }, {
        value: 'form',
        label: 'Form',
        children: [{
          value: 'radio',
          label: 'Radio'
        }, {
          value: 'checkbox',
          label: 'Checkbox'
        }, {
          value: 'input',
          label: 'Input'
        }, {
          value: 'input-number',
          label: 'InputNumber'
        }, {
          value: 'select',
          label: 'Select'
        }, {
          value: 'cascader',
          label: 'Cascader'
        }, {
          value: 'switch',
          label: 'Switch'
        }, {
          value: 'slider',
          label: 'Slider'
        }, {
          value: 'time-picker',
          label: 'TimePicker'
        }, {
          value: 'date-picker',
          label: 'DatePicker'
        }, {
          value: 'datetime-picker',
          label: 'DateTimePicker'
        }, {
          value: 'upload',
          label: 'Upload'
        }, {
          value: 'rate',
          label: 'Rate'
        }, {
          value: 'form',
          label: 'Form'
        }]
      }, {
        value: 'data',
        label: 'Data',
        children: [{
          value: 'table',
          label: 'Table'
        }, {
          value: 'tag',
          label: 'Tag'
        }, {
```

```
      value: 'progress',
      label: 'Progress'
    }, {
      value: 'tree',
      label: 'Tree'
    }, {
      value: 'pagination',
      label: 'Pagination'
    }, {
      value: 'badge',
      label: 'Badge'
    }]
  }, {
    value: 'notice',
    label: 'Notice',
    children: [{
      value: 'alert',
      label: 'Alert'
    }, {
      value: 'loading',
      label: 'Loading'
    }, {
      value: 'message',
      label: 'Message'
    }, {
      value: 'message-box',
      label: 'MessageBox'
    }, {
      value: 'notification',
      label: 'Notification'
    }]
  }, {
    value: 'navigation',
    label: 'Navigation',
    children: [{
      value: 'menu',
      label: 'NavMenu'
    }, {
      value: 'tabs',
      label: 'Tabs'
    }, {
      value: 'breadcrumb',
      label: 'Breadcrumb'
    }, {
      value: 'dropdown',
      label: 'Dropdown'
    }, {
      value: 'steps',
      label: 'Steps'
    }]
  }, {
    value: 'others',
              label: 'Others',
              children: [{
                value: 'dialog',
                label: 'Dialog'
              }, {
                value: 'tooltip',
                label: 'Tooltip'
              }, {
                value: 'popover',
                label: 'Popover'
              }, {
                value: 'card',
                label: 'Card'
              }, {
                value: 'carousel',
                label: 'Carousel'
              }, {
                value: 'collapse',
                label: 'Collapse'
              }]
            }]
          }, {
            value: 'resource',
            label: 'Resource',
            children: [{
              value: 'axure',
              label: 'Axure Components'
            }, {
              value: 'sketch',
              label: 'Sketch Templates'
            }, {
              value: 'docs',
              label: 'Design Documentation'
            }]
          }]
        };
      }
    };
</script>
```

:::

Clearable

Set `clearable` attribute for `el-cascader` and a clear icon will appear when selected and hovered

:::demo

```
<el-cascader :options="options" clearable></el-cascader>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
            value: 'navigation',
            label: 'Navigation',
            children: [{
              value: 'side nav',
              label: 'Side Navigation'
            }, {
              value: 'top nav',
              label: 'Top Navigation'
            }]
          }]
        }, {
          value: 'component',
          label: 'Component',
          children: [{
            value: 'basic',
            label: 'Basic',
            children: [{
              value: 'layout',
              label: 'Layout'
            }, {
              value: 'color',
              label: 'Color'
            }, {
              value: 'typography',
              label: 'Typography'
            }, {
              value: 'icon',
              label: 'Icon'
            }, {
              value: 'button',
      label: 'Button'
    }]
  }, {
    value: 'form',
    label: 'Form',
    children: [{
      value: 'radio',
      label: 'Radio'
    }, {
      value: 'checkbox',
      label: 'Checkbox'
    }, {
      value: 'input',
      label: 'Input'
    }, {
      value: 'input-number',
      label: 'InputNumber'
    }, {
      value: 'select',
      label: 'Select'
    }, {
      value: 'cascader',
      label: 'Cascader'
    }, {
      value: 'switch',
      label: 'Switch'
    }, {
      value: 'slider',
      label: 'Slider'
    }, {
      value: 'time-picker',
      label: 'TimePicker'
    }, {
      value: 'date-picker',
      label: 'DatePicker'
    }, {
      value: 'datetime-picker',
      label: 'DateTimePicker'
    }, {
      value: 'upload',
      label: 'Upload'
    }, {
      value: 'rate',
      label: 'Rate'
    }, {
      value: 'form',
      label: 'Form'
    }]
  }, {
    value: 'data',
    label: 'Data',
    children: [{
        value: 'table',
        label: 'Table'
      }, {
        value: 'tag',
        label: 'Tag'
      }, {
        value: 'progress',
        label: 'Progress'
      }, {
        value: 'tree',
        label: 'Tree'
      }, {
        value: 'pagination',
        label: 'Pagination'
      }, {
        value: 'badge',
        label: 'Badge'
      }]
    }, {
      value: 'notice',
      label: 'Notice',
      children: [{
        value: 'alert',
        label: 'Alert'
      }, {
        value: 'loading',
        label: 'Loading'
      }, {
        value: 'message',
        label: 'Message'
      }, {
        value: 'message-box',
        label: 'MessageBox'
      }, {
        value: 'notification',
        label: 'Notification'
      }]
    }, {
      value: 'navigation',
      label: 'Navigation',
      children: [{
        value: 'menu',
        label: 'NavMenu'
      }, {
        value: 'tabs',
        label: 'Tabs'
      }, {
        value: 'breadcrumb',
        label: 'Breadcrumb'
      }, {
        value: 'dropdown',
        label: 'Dropdown'
```

```
          }, {
            value: 'steps',
            label: 'Steps'
          }]
        }, {
          value: 'others',
          label: 'Others',
          children: [{
            value: 'dialog',
            label: 'Dialog'
          }, {
            value: 'tooltip',
            label: 'Tooltip'
          }, {
            value: 'popover',
            label: 'Popover'
          }, {
            value: 'card',
            label: 'Card'
          }, {
            value: 'carousel',
            label: 'Carousel'
          }, {
            value: 'collapse',
            label: 'Collapse'
          }]
        }]
      }, {
        value: 'resource',
        label: 'Resource',
        children: [{
          value: 'axure',
          label: 'Axure Components'
        }, {
          value: 'sketch',
          label: 'Sketch Templates'
        }, {
          value: 'docs',
          label: 'Design Documentation'
        }]
      }]
    }
  }
}
</script>
```

:::

Display only the last level

The input can display only the last level instead of all levels.

:::demo The `show-all-levels` attribute defines if all levels are displayed. If it is `false`, only the last level is displayed.

```
<el-cascader :options="options" :show-all-levels="false"></el-cascader>
<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
            value: 'navigation',
            label: 'Navigation',
            children: [{
              value: 'side nav',
              label: 'Side Navigation'
            }, {
              value: 'top nav',
              label: 'Top Navigation'
            }]
          }]
        }, {
          value: 'component',
          label: 'Component',
          children: [{
            value: 'basic',
            label: 'Basic',
            children: [{
              value: 'layout',
              label: 'Layout'
            }, {
              value: 'color',
              label: 'Color'
            }, {
```

```
        value: 'typography',
        label: 'Typography'
      }, {
        value: 'icon',
        label: 'Icon'
      }, {
        value: 'button',
        label: 'Button'
      }]
    }, {
      value: 'form',
      label: 'Form',
      children: [{
        value: 'radio',
        label: 'Radio'
      }, {
        value: 'checkbox',
        label: 'Checkbox'
      }, {
        value: 'input',
        label: 'Input'
      }, {
        value: 'input-number',
        label: 'InputNumber'
      }, {
        value: 'select',
        label: 'Select'
      }, {
        value: 'cascader',
        label: 'Cascader'
      }, {
        value: 'switch',
        label: 'Switch'
      }, {
        value: 'slider',
        label: 'Slider'
      }, {
        value: 'time-picker',
        label: 'TimePicker'
      }, {
        value: 'date-picker',
        label: 'DatePicker'
      }, {
        value: 'datetime-picker',
        label: 'DateTimePicker'
      }, {
        value: 'upload',
        label: 'Upload'
      }, {
        value: 'rate',
        label: 'Rate'
      }, {
        value: 'form',
        label: 'Form'
      }]
    }, {
      value: 'data',
      label: 'Data',
      children: [{
        value: 'table',
        label: 'Table'
      }, {
        value: 'tag',
        label: 'Tag'
      }, {
        value: 'progress',
        label: 'Progress'
      }, {
        value: 'tree',
        label: 'Tree'
      }, {
        value: 'pagination',
        label: 'Pagination'
      }, {
        value: 'badge',
        label: 'Badge'
      }]
    }, {
      value: 'notice',
      label: 'Notice',
      children: [{
        value: 'alert',
        label: 'Alert'
      }, {
        value: 'loading',
        label: 'Loading'
      }, {
        value: 'message',
        label: 'Message'
      }, {
        value: 'message-box',
        label: 'MessageBox'
      }, {
        value: 'notification',
        label: 'Notification'
      }]
    }, {
      value: 'navigation',
      label: 'Navigation',
      children: [{
        value: 'menu',
        label: 'NavMenu'
      }, {
        value: 'tabs',
            label: 'Tabs'
          }, {
            value: 'breadcrumb',
            label: 'Breadcrumb'
          }, {
            value: 'dropdown',
            label: 'Dropdown'
          }, {
            value: 'steps',
            label: 'Steps'
          }]
        }, {
          value: 'others',
          label: 'Others',
          children: [{
            value: 'dialog',
            label: 'Dialog'
          }, {
            value: 'tooltip',
            label: 'Tooltip'
          }, {
            value: 'popover',
            label: 'Popover'
          }, {
            value: 'card',
            label: 'Card'
          }, {
            value: 'carousel',
            label: 'Carousel'
          }, {
            value: 'collapse',
            label: 'Collapse'
          }]
        }]
      }, {
        value: 'resource',
        label: 'Resource',
        children: [{
          value: 'axure',
          label: 'Axure Components'
        }, {
          value: 'sketch',
          label: 'Sketch Templates'
        }, {
          value: 'docs',
          label: 'Design Documentation'
        }]
      }]
    };
  }
};
</script>
```

:::

Multiple Selection

Set `props.multiple = true` to use multiple selection.

:::demo When using multiple selection, all selected tags will display by default, You can set `collapse-tags = true` to fold selected tags.

```
<div class="block">
  <span class="demonstration">Display all tags (default)</span>
  <el-cascader
    :options="options"
    :props="props"
    clearable></el-cascader>
</div>
<div class="block">
  <span class="demonstration">Collapse tags</span>
  <el-cascader
    :options="options"
    :props="props"
    collapse-tags
    clearable></el-cascader>
</div>

<script>
  export default {
    data() {
      return {
        props: { multiple: true },
        options: [{
          value: 1,
          label: 'Asia',
          children: [{
            value: 2,
            label: 'China',
            children: [
              { value: 3, label: 'Beijing' },
              { value: 4, label: 'Shanghai' },
              { value: 5, label: 'Hangzhou' }
            ]
          }, {
            value: 6,
            label: 'Japan',
            children: [
              { value: 7, label: 'Tokyo' },
              { value: 8, label: 'Osaka' },
              { value: 9, label: 'Kyoto' }
            ]
          }, {
            value: 10,
              label: 'Korea',
              children: [
                { value: 11, label: 'Seoul' },
                { value: 12, label: 'Busan' },
                { value: 13, label: 'Taegu' }
              ]
            }]
          }, {
            value: 14,
            label: 'Europe',
            children: [{
              value: 15,
              label: 'France',
              children: [
                { value: 16, label: 'Paris' },
                { value: 17, label: 'Marseille' },
                { value: 18, label: 'Lyon' }
              ]
            }, {
              value: 19,
              label: 'UK',
              children: [
                { value: 20, label: 'London' },
                { value: 21, label: 'Birmingham' },
                { value: 22, label: 'Manchester' }
              ]
            }]
          }, {
            value: 23,
            label: 'North America',
            children: [{
              value: 24,
              label: 'US',
              children: [
                { value: 25, label: 'New York' },
                { value: 26, label: 'Los Angeles' },
                { value: 27, label: 'Washington' }
              ]
            }, {
              value: 28,
              label: 'Canada',
              children: [
                { value: 29, label: 'Toronto' },
                { value: 30, label: 'Montreal' },
                { value: 31, label: 'Ottawa' }
              ]
            }]
          }]
        };
      }
    };
  </script>
```

:::

Select any level of options

In single selection, only the leaf nodes can be checked, and in multiple selection, check parent nodes will lead to leaf nodes be checked eventually. When enable this feature, it can make parent and child nodes unlinked and you can select any level of options.

:::demo Set `props.checkStrictly = true` to make checked state of a node not affects its parent nodes and child nodes, and then you can select any level of options.

```
<div class="block">
  <span class="demonstration">Select any level of options (Single selection)</span>
  <el-cascader
    :options="options"
    :props="{ checkStrictly: true }"
    clearable></el-cascader>
</div>
<div class="block">
  <span class="demonstration">Select any level of options (Multiple selection)
</span>
  <el-cascader
    :options="options"
    :props="{ multiple: true, checkStrictly: true }"
    clearable></el-cascader>
</div>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
```

```
        value: 'navigation',
        label: 'Navigation',
        children: [{
          value: 'side nav',
          label: 'Side Navigation'
        }, {
          value: 'top nav',
          label: 'Top Navigation'
        }]
      }]
    }, {
      value: 'component',
      label: 'Component',
      children: [{
        value: 'basic',
        label: 'Basic',
        children: [{
          value: 'layout',
          label: 'Layout'
        }, {
          value: 'color',
          label: 'Color'
        }, {
          value: 'typography',
          label: 'Typography'
        }, {
          value: 'icon',
          label: 'Icon'
        }, {
          value: 'button',
          label: 'Button'
        }]
      }, {
        value: 'form',
        label: 'Form',
        children: [{
          value: 'radio',
          label: 'Radio'
        }, {
          value: 'checkbox',
          label: 'Checkbox'
        }, {
          value: 'input',
          label: 'Input'
        }, {
          value: 'input-number',
          label: 'InputNumber'
        }, {
          value: 'select',
          label: 'Select'
        }, {
          value: 'cascader',
        label: 'Cascader'
      }, {
        value: 'switch',
        label: 'Switch'
      }, {
        value: 'slider',
        label: 'Slider'
      }, {
        value: 'time-picker',
        label: 'TimePicker'
      }, {
        value: 'date-picker',
        label: 'DatePicker'
      }, {
        value: 'datetime-picker',
        label: 'DateTimePicker'
      }, {
        value: 'upload',
        label: 'Upload'
      }, {
        value: 'rate',
        label: 'Rate'
      }, {
        value: 'form',
        label: 'Form'
      }]
    }, {
      value: 'data',
      label: 'Data',
      children: [{
        value: 'table',
        label: 'Table'
      }, {
        value: 'tag',
        label: 'Tag'
      }, {
        value: 'progress',
        label: 'Progress'
      }, {
        value: 'tree',
        label: 'Tree'
      }, {
        value: 'pagination',
        label: 'Pagination'
      }, {
        value: 'badge',
        label: 'Badge'
      }]
    }, {
      value: 'notice',
      label: 'Notice',
      children: [{
```

```
      value: 'alert',
      label: 'Alert'
    }, {
      value: 'loading',
      label: 'Loading'
    }, {
      value: 'message',
      label: 'Message'
    }, {
      value: 'message-box',
      label: 'MessageBox'
    }, {
      value: 'notification',
      label: 'Notification'
    }]
  }, {
    value: 'navigation',
    label: 'Navigation',
    children: [{
      value: 'menu',
      label: 'NavMenu'
    }, {
      value: 'tabs',
      label: 'Tabs'
    }, {
      value: 'breadcrumb',
      label: 'Breadcrumb'
    }, {
      value: 'dropdown',
      label: 'Dropdown'
    }, {
      value: 'steps',
      label: 'Steps'
    }]
  }, {
    value: 'others',
    label: 'Others',
    children: [{
      value: 'dialog',
      label: 'Dialog'
    }, {
      value: 'tooltip',
      label: 'Tooltip'
    }, {
      value: 'popover',
      label: 'Popover'
    }, {
      value: 'card',
      label: 'Card'
    }, {
      value: 'carousel',
      label: 'Carousel'
```

```
            }, {
              value: 'collapse',
              label: 'Collapse'
            }]
          }]
        }, {
          value: 'resource',
          label: 'Resource',
          children: [{
            value: 'axure',
            label: 'Axure Components'
          }, {
            value: 'sketch',
            label: 'Sketch Templates'
          }, {
            value: 'docs',
            label: 'Design Documentation'
          }]
        }]
      };
    }
  };
</script>
```

:::

Dynamic loading

Dynamic load its child nodes when checked a node.

:::demo Set `lazy = true` to use dynamic loading, and you have to specify how to load the data source by `lazyload`. There are two parameters of `lazyload`,the first parameter `node` is the node currently clicked, and the `resolve` is a callback that indicate loading is finished which must invoke. To display the status of node more accurately, you can add a `leaf` field (can be modified by `props.leaf`) to indicate whether it is a leaf node. Otherwise, it will be inferred by if has any child nodes.

```
<el-cascader :props="props"></el-cascader>

<script>
  let id = 0;

  export default {
    data() {
      return {
        props: {
          lazy: true,
          lazyLoad (node, resolve) {
            const { level } = node;
            setTimeout(() => {
              const nodes = Array.from({ length: level + 1 })
                .map(item => ({
                  value: ++id,
                  label: `Option - ${id}`,
                  leaf: level >= 2
                }));
              // Invoke `resolve` callback to return the child nodes data and
indicate the loading is finished.
              resolve(nodes);
            }, 1000);
          }
        }
      };
    }
  };
</script>
```

:::

Filterable

Search and select options with a keyword.

:::demo Adding `filterable` to `el-cascader` enables filtering. Cascader will match nodes whose label or parent's label (according to `show-all-levels`) includes input keyword. Of course, you can customize search logic by `filter-method` which accepts a function, the first parameter is `node`, the second is `keyword`, and need return a boolean value indicating whether it hits.

```
<div class="block">
  <span class="demonstration">Filterable (Single selection)</span>
  <el-cascader
    placeholder="Try searchingL Guide"
    :options="options"
    filterable></el-cascader>
</div>
<div class="block">
  <span class="demonstration">Filterable (Multiple selection)</span>
  <el-cascader
    placeholder="Try searchingL Guide"
    :options="options"
    :props="{ multiple: true }"
    filterable></el-cascader>
</div>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
            value: 'navigation',
            label: 'Navigation',
            children: [{
              value: 'side nav',
              label: 'Side Navigation'
            }, {
              value: 'top nav',
              label: 'Top Navigation'
            }]
          }]
        }, {
          value: 'component',
          label: 'Component',
          children: [{
            value: 'basic',
            label: 'Basic',
            children: [{
              value: 'layout',
              label: 'Layout'
            }, {
              value: 'color',
              label: 'Color'
            }, {
              value: 'typography',
              label: 'Typography'
            }, {
              value: 'icon',
              label: 'Icon'
            }, {
              value: 'button',
              label: 'Button'
            }]
          }, {
            value: 'form',
            label: 'Form',
            children: [{
              value: 'radio',
              label: 'Radio'
          }, {
            value: 'checkbox',
            label: 'Checkbox'
          }, {
            value: 'input',
            label: 'Input'
          }, {
            value: 'input-number',
            label: 'InputNumber'
          }, {
            value: 'select',
            label: 'Select'
          }, {
            value: 'cascader',
            label: 'Cascader'
          }, {
            value: 'switch',
            label: 'Switch'
          }, {
            value: 'slider',
            label: 'Slider'
          }, {
            value: 'time-picker',
            label: 'TimePicker'
          }, {
            value: 'date-picker',
            label: 'DatePicker'
          }, {
            value: 'datetime-picker',
            label: 'DateTimePicker'
          }, {
            value: 'upload',
            label: 'Upload'
          }, {
            value: 'rate',
            label: 'Rate'
          }, {
            value: 'form',
            label: 'Form'
          }]
        }, {
          value: 'data',
          label: 'Data',
          children: [{
            value: 'table',
            label: 'Table'
          }, {
            value: 'tag',
            label: 'Tag'
          }, {
            value: 'progress',
            label: 'Progress'
        }, {
          value: 'tree',
          label: 'Tree'
        }, {
          value: 'pagination',
          label: 'Pagination'
        }, {
          value: 'badge',
          label: 'Badge'
        }]
      }, {
        value: 'notice',
        label: 'Notice',
        children: [{
          value: 'alert',
          label: 'Alert'
        }, {
          value: 'loading',
          label: 'Loading'
        }, {
          value: 'message',
          label: 'Message'
        }, {
          value: 'message-box',
          label: 'MessageBox'
        }, {
          value: 'notification',
          label: 'Notification'
        }]
      }, {
        value: 'navigation',
        label: 'Navigation',
        children: [{
          value: 'menu',
          label: 'NavMenu'
        }, {
          value: 'tabs',
          label: 'Tabs'
        }, {
          value: 'breadcrumb',
          label: 'Breadcrumb'
        }, {
          value: 'dropdown',
          label: 'Dropdown'
        }, {
          value: 'steps',
          label: 'Steps'
        }]
      }, {
        value: 'others',
        label: 'Others',
        children: [{
```

```
                value: 'dialog',
                label: 'Dialog'
              }, {
                value: 'tooltip',
                label: 'Tooltip'
              }, {
                value: 'popover',
                label: 'Popover'
              }, {
                value: 'card',
                label: 'Card'
              }, {
                value: 'carousel',
                label: 'Carousel'
              }, {
                value: 'collapse',
                label: 'Collapse'
              }]
            }]
          }, {
            value: 'resource',
            label: 'Resource',
            children: [{
              value: 'axure',
              label: 'Axure Components'
            }, {
              value: 'sketch',
              label: 'Sketch Templates'
            }, {
              value: 'docs',
              label: 'Design Documentation'
            }]
          }]
        };
      }
    };
</script>
```

:::

Custom option content

You can customize the content of cascader node.

:::demo You can customize the content of cascader node by `scoped slot`. You'll have access to `node` and `data` in the scope, standing for the Node object and node data of the current node respectively。

```
<el-cascader :options="options">
  <template slot-scope="{ node, data }">
    <span>{{ data.label }}</span>
    <span v-if="!node.isLeaf"> ({{ data.children.length }}) </span>
  </template>
```

```
</el-cascader>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
            value: 'navigation',
            label: 'Navigation',
            children: [{
              value: 'side nav',
              label: 'Side Navigation'
            }, {
              value: 'top nav',
              label: 'Top Navigation'
            }]
          }]
        }, {
          value: 'component',
          label: 'Component',
          children: [{
            value: 'basic',
            label: 'Basic',
            children: [{
              value: 'layout',
              label: 'Layout'
            }, {
              value: 'color',
              label: 'Color'
            }, {
              value: 'typography',
              label: 'Typography'
            }, {
```

```
        value: 'icon',
        label: 'Icon'
      }, {
        value: 'button',
        label: 'Button'
      }]
    }, {
      value: 'form',
      label: 'Form',
      children: [{
        value: 'radio',
        label: 'Radio'
      }, {
        value: 'checkbox',
        label: 'Checkbox'
      }, {
        value: 'input',
        label: 'Input'
      }, {
        value: 'input-number',
        label: 'InputNumber'
      }, {
        value: 'select',
        label: 'Select'
      }, {
        value: 'cascader',
        label: 'Cascader'
      }, {
        value: 'switch',
        label: 'Switch'
      }, {
        value: 'slider',
        label: 'Slider'
      }, {
        value: 'time-picker',
        label: 'TimePicker'
      }, {
        value: 'date-picker',
        label: 'DatePicker'
      }, {
        value: 'datetime-picker',
        label: 'DateTimePicker'
      }, {
        value: 'upload',
        label: 'Upload'
      }, {
        value: 'rate',
        label: 'Rate'
      }, {
        value: 'form',
        label: 'Form'
      }]
```

```
        }, {
          value: 'data',
          label: 'Data',
          children: [{
            value: 'table',
            label: 'Table'
          }, {
            value: 'tag',
            label: 'Tag'
          }, {
            value: 'progress',
            label: 'Progress'
          }, {
            value: 'tree',
            label: 'Tree'
          }, {
            value: 'pagination',
            label: 'Pagination'
          }, {
            value: 'badge',
            label: 'Badge'
          }]
        }, {
          value: 'notice',
          label: 'Notice',
          children: [{
            value: 'alert',
            label: 'Alert'
          }, {
            value: 'loading',
            label: 'Loading'
          }, {
            value: 'message',
            label: 'Message'
          }, {
            value: 'message-box',
            label: 'MessageBox'
          }, {
            value: 'notification',
            label: 'Notification'
          }]
        }, {
          value: 'navigation',
          label: 'Navigation',
          children: [{
            value: 'menu',
            label: 'NavMenu'
          }, {
            value: 'tabs',
            label: 'Tabs'
          }, {
            value: 'breadcrumb',
                label: 'Breadcrumb'
              }, {
                value: 'dropdown',
                label: 'Dropdown'
              }, {
                value: 'steps',
                label: 'Steps'
              }]
            }, {
              value: 'others',
              label: 'Others',
              children: [{
                value: 'dialog',
                label: 'Dialog'
              }, {
                value: 'tooltip',
                label: 'Tooltip'
              }, {
                value: 'popover',
                label: 'Popover'
              }, {
                value: 'card',
                label: 'Card'
              }, {
                value: 'carousel',
                label: 'Carousel'
              }, {
                value: 'collapse',
                label: 'Collapse'
              }]
            }]
          }, {
            value: 'resource',
            label: 'Resource',
            children: [{
              value: 'axure',
              label: 'Axure Components'
            }, {
              value: 'sketch',
              label: 'Sketch Templates'
            }, {
              value: 'docs',
              label: 'Design Documentation'
            }]
          }]
        }
      }
    }
</script>
```

:::

Cascader panel

`CascaderPanel` is the core component of `Cascader` which has various of features such as single selection, multiple selection, dynamic loading and so on.

:::demo Just like `el-cascader`, you can set alternative options by `options`, and enable other features by `props`, see the API form below for details.

```
<el-cascader-panel :options="options"></el-cascader-panel>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'guide',
          label: 'Guide',
          children: [{
            value: 'disciplines',
            label: 'Disciplines',
            children: [{
              value: 'consistency',
              label: 'Consistency'
            }, {
              value: 'feedback',
              label: 'Feedback'
            }, {
              value: 'efficiency',
              label: 'Efficiency'
            }, {
              value: 'controllability',
              label: 'Controllability'
            }]
          }, {
            value: 'navigation',
            label: 'Navigation',
            children: [{
              value: 'side nav',
              label: 'Side Navigation'
            }, {
              value: 'top nav',
              label: 'Top Navigation'
            }]
          }]
        }, {
          value: 'component',
          label: 'Component',
          children: [{
            value: 'basic',
            label: 'Basic',
            children: [{
              value: 'layout',
          label: 'Layout'
        }, {
          value: 'color',
          label: 'Color'
        }, {
          value: 'typography',
          label: 'Typography'
        }, {
          value: 'icon',
          label: 'Icon'
        }, {
          value: 'button',
          label: 'Button'
        }]
      }, {
        value: 'form',
        label: 'Form',
        children: [{
          value: 'radio',
          label: 'Radio'
        }, {
          value: 'checkbox',
          label: 'Checkbox'
        }, {
          value: 'input',
          label: 'Input'
        }, {
          value: 'input-number',
          label: 'InputNumber'
        }, {
          value: 'select',
          label: 'Select'
        }, {
          value: 'cascader',
          label: 'Cascader'
        }, {
          value: 'switch',
          label: 'Switch'
        }, {
          value: 'slider',
          label: 'Slider'
        }, {
          value: 'time-picker',
          label: 'TimePicker'
        }, {
          value: 'date-picker',
          label: 'DatePicker'
        }, {
          value: 'datetime-picker',
          label: 'DateTimePicker'
        }, {
          value: 'upload',
        label: 'Upload'
      }, {
        value: 'rate',
        label: 'Rate'
      }, {
        value: 'form',
        label: 'Form'
      }]
    }, {
      value: 'data',
      label: 'Data',
      children: [{
        value: 'table',
        label: 'Table'
      }, {
        value: 'tag',
        label: 'Tag'
      }, {
        value: 'progress',
        label: 'Progress'
      }, {
        value: 'tree',
        label: 'Tree'
      }, {
        value: 'pagination',
        label: 'Pagination'
      }, {
        value: 'badge',
        label: 'Badge'
      }]
    }, {
      value: 'notice',
      label: 'Notice',
      children: [{
        value: 'alert',
        label: 'Alert'
      }, {
        value: 'loading',
        label: 'Loading'
      }, {
        value: 'message',
        label: 'Message'
      }, {
        value: 'message-box',
        label: 'MessageBox'
      }, {
        value: 'notification',
        label: 'Notification'
      }]
    }, {
      value: 'navigation',
      label: 'Navigation',
          children: [{
            value: 'menu',
            label: 'NavMenu'
          }, {
            value: 'tabs',
            label: 'Tabs'
          }, {
            value: 'breadcrumb',
            label: 'Breadcrumb'
          }, {
            value: 'dropdown',
            label: 'Dropdown'
          }, {
            value: 'steps',
            label: 'Steps'
          }]
        }, {
          value: 'others',
          label: 'Others',
          children: [{
            value: 'dialog',
            label: 'Dialog'
          }, {
            value: 'tooltip',
            label: 'Tooltip'
          }, {
            value: 'popover',
            label: 'Popover'
          }, {
            value: 'card',
            label: 'Card'
          }, {
            value: 'carousel',
            label: 'Carousel'
          }, {
            value: 'collapse',
            label: 'Collapse'
          }]
        }]
      }, {
        value: 'resource',
        label: 'Resource',
        children: [{
          value: 'axure',
          label: 'Axure Components'
        }, {
          value: 'sketch',
          label: 'Sketch Templates'
        }, {
          value: 'docs',
          label: 'Design Documentation'
        }]
        }]
      };
    }
  };
</script>
```

:::

Cascader Attributes

Attribute	Description	Type	Accepted Values	Default
value / v-model	binding value	-	—	—
options	data of the options，the key of `value` and `label` can be customize by `Props`.	array	—	—
props	configuration options, see the following table.	object	—	—
size	size of input	string	medium / small / mini	—
placeholder	placeholder of input	string	—	Select
disabled	whether Cascader is disabled	boolean	—	false
clearable	whether selected value can be cleared	boolean	—	false
show-all-levels	whether to display all levels of the selected value in the input	boolean	—	true
collapse-tags	whether to collapse tags in multiple selection mode	boolean	-	false
separator	option label separator	string	—	' / '
filterable	whether the options can be searched	boolean	—	—
filter-method	customize search logic, the first parameter is `node`, the second is `keyword`, and need return a boolean value indicating whether it hits.	function(node, keyword)	-	-
debounce	debounce delay when typing filter keyword, in milliseconds	number	—	300
before-filter	hook function before filtering with the value to be filtered as its parameter. If `false` is returned or a `Promise` is returned and then is rejected, filtering will be aborted	function(value)	—	—
popper-class	custom class name for Cascader's dropdown	string	—	—

Cascader Events

Event Name	Description	Parameters
change	triggers when the binding value changes	value
expand-change	triggers when expand option changes	an array of the expanding node's parent nodes
blur	triggers when Cascader blurs	(event: Event)
focus	triggers when Cascader focuses	(event: Event)
visible-change	triggers when the dropdown appears/disappears	true when it appears, and false otherwise
remove-tag	triggers when remove tag in multiple selection mode	the value of the tag which is removed

Cascader Methods

Method Name	Description	Parameters
getCheckedNodes	get an array of currently selected node	(leafOnly) whether only return the leaf checked nodes, default is `false`

Cascader Slots

Slot Name	Description
-	the custom content of cascader node, the parameter is { node, data }, which are current Node object and node data respectively.
empty	content when there is no matched options.

CascaderPanel Attributes

Attribute	Description	Type	Accepted Values	Default
value / v-model	binding value	-	—	—
options	data of the options，the key of `value` and `label` can be customize by `Props`.	array	—	—
props	configuration options, see the following table.	object	—	—

CascaderPanel Events

Event Name	Description	Parameters
change	triggers when the binding value changes	value

expand-change	triggers when expand option changes	an array of the expanding node's parent nodes

CascaderPanel Methods

Method Name	Description	Parameters
getCheckedNodes	get an array of currently selected node	(leafOnly) whether only return the leaf checked nodes, default is `false`
clearCheckedNodes	clear checked nodes	-

CascaderPanel Slots

Slot Name	Description
-	the custom content of cascader node, the parameter is { node, data }, which are current Node object and node data respectively.

Props

Attribute	Description	Type	Accepted Values	Default
expandTrigger	trigger mode of expanding options	string	click / hover	'click'
multiple	whether multiple selection is enabled	boolean	-	false
checkStrictly	whether checked state of a node not affects its parent and child nodes	boolean	-	false
emitPath	when checked nodes change, whether to emit an array of node's path, if false, only emit the value of node.	boolean	-	true
lazy	whether to dynamic load child nodes, use with `lazyload` attribute	boolean	-	false
lazyLoad	method for loading child nodes data, only works when `lazy` is true	function(node, resolve)	-	-
value	specify which key of node object is used as the node's value	string	—	'value'
label	specify which key of node object is used as the node's label	string	—	'label'
children	specify which key of node object is used as the node's children	string	—	'children'
disabled	specify which key of node object is used as the node's disabled	string	—	'disabled'

leaf	specify which key of node object is used as the node's leaf field	string	—	'leaf'