

Popover

Basic usage

Similar to Tooltip, Popover is also built with **Vue-popper**. So for some duplicated attributes, please refer to the documentation of Tooltip.

:::demo The **trigger** attribute is used to define how popover is triggered: **hover**, **click**, **focus** or **manual**. As for the triggering element, you can write it in two different ways: use the **slot="reference"** named slot, or use the **v-popover** directive and set it to Popover's **ref**.

```
<template>
  <el-popover
    placement="top-start"
    title="Title"
    width="200"
    trigger="hover"
    content="this is content, this is content, this is content">
    <el-button slot="reference">Hover to activate</el-button>
  </el-popover>

  <el-popover
    placement="bottom"
    title="Title"
    width="200"
    trigger="click"
    content="this is content, this is content, this is content">
    <el-button slot="reference">Click to activate</el-button>
  </el-popover>

  <el-popover
    ref="popover"
    placement="right"
    title="Title"
    width="200"
    trigger="focus"
    content="this is content, this is content, this is content">
  </el-popover>
  <el-button v-popover:popover>Focus to activate</el-button>

  <el-popover
    placement="bottom"
    title="Title"
    width="200"
    trigger="manual"
    content="this is content, this is content, this is content"
```

```

      v-model="visible">
      <el-button slot="reference" @click="visible = !visible">Manual to activate</el-button>
    </el-popover>
  </template>

<script>
  export default {
    data() {
      return {
        visible: false
      };
    }
  };
</script>
:::

```

Nested information

Other components can be nested in popover. Following is an example of nested table.

:::demo replace the content attribute with a default slot.

```

<el-popover
  placement="right"
  width="400"
  trigger="click">
  <el-table :data="gridData">
    <el-table-column width="150" property="date" label="date"></el-table-column>
    <el-table-column width="100" property="name" label="name"></el-table-column>
    <el-table-column width="300" property="address" label="address"></el-table-column>
  </el-table>
  <el-button slot="reference">Click to activate</el-button>
</el-popover>

<script>
  export default {
    data() {
      return {
        gridData: [{
          date: '2016-05-02',
          name: 'Jack',
          address: 'New York City'
        }, {
          date: '2016-05-04',
          name: 'Jack',

```

```

        address: 'New York City'
      }, {
        date: '2016-05-01',
        name: 'Jack',
        address: 'New York City'
      }, {
        date: '2016-05-03',
        name: 'Jack',
        address: 'New York City'
      }
    ]
  };
}
};
</script>
:::

```

Nested operation

Of course, you can nest other operations. It's more light-weight than using a dialog.

```

<el-popover
  placement="top"
  width="160"
  v-model="visible">
  <p>Are you sure to delete this?</p>
  <div style="text-align: right; margin: 0">
    <el-button size="mini" type="text" @click="visible = false">cancel</el-button>
    <el-button type="primary" size="mini" @click="visible = false">confirm</el-button>
  </div>
  <el-button slot="reference">Delete</el-button>
</el-popover>

<script>
  export default {
    data() {
      return {
        visible: false,
      };
    }
  }
</script>

```

Attributes

Attribute	Description	Type	Accepted Values	Default
trigger	how the popover is triggered	string	click/focus/hover/manual	
title	popover title	string	—	—
content	popover content, can be replaced with a default <code>slot</code>	string	—	—
width	popover width	string, number	—	Min width 150px
placement	popover placement	string	top/top-start/top-end/bottom/bottom-start/bottom-end/left/left-start/left-end/right/right-start/right-end	bottom
disabled	whether Popover is disabled	boolean	—	false
value / v-model	whether popover is visible	Boolean	—	false
offset	popover offset	number	—	0
transition	popover transition animation	string	—	el-fade-in-linear
visible-arrow	whether a tooltip arrow is displayed or not. For more info, please refer to Vue-popper	boolean	—	true
popper-options	parameters for popper.js	object	please refer to popper.js	{ boundariesElement: 'body', gpuAcceleration: false }
popper-class	custom class name for popover	string	—	—
open-delay	delay before appearing when <code>trigger</code> is hover, in milliseconds	number	—	—
close-delay	delay before disappearing when <code>trigger</code> is hover, in milliseconds	number	—	200
tabindex	tabindex of Popover	number	—	0

Slot

Name	Description
—	text content of popover
reference	HTML element that triggers popover

Events

Event Name	Description	Parameters
show	triggers when popover shows	—
after-enter	triggers when the entering transition ends	—
hide	triggers when popover hides	—
after-leave	triggers when the leaving transition ends	—