

Dialog

Informar a usuarios preservando el estado de la página actual.

Uso Básico

Dialog abre una caja de diálogo, y es bastante personalizable.

:::demo Establezca el atributo `visible` con un booleano, y el Dialog se muestra cuando es `true`. El diálogo tiene dos partes: `body` y `footer`, este último requiere un slot llamado `footer`. El atributo `title` es opcional (vacío por defecto) y sirve para definir un título. Por último, este ejemplo muestra cómo se utiliza `before-close`.

```
<el-button type="text" @click="dialogVisible = true">click to open the Dialog</el-button>

<el-dialog
  title="Tips"
  :visible.sync="dialogVisible"
  width="30%"
  :before-close="handleClose">
  <span>This is a message</span>
  <span slot="footer" class="dialog-footer">
    <el-button @click="dialogVisible = false">Cancel</el-button>
    <el-button type="primary" @click="dialogVisible = false">Confirm</el-button>
  </span>
</el-dialog>

<script>
export default {
  data() {
    return {
      dialogVisible: false
    };
  },
  methods: {
    handleClose(done) {
      this.$confirm('Are you sure to close this dialog?')
        .then(_ => {
          done();
        })
        .catch(_ => {});
    }
  }
};
</script>

:::
```

`before-close` sólo funciona cuando el usuario hace clic en el icono de cerrar o en el fondo. Si tiene botones que cierran el cuadro de diálogo en el slot llamado `footer`, puede agregar lo que haría `before-close` en el manejador de eventos de los botones.

Personalizaciones

El contenido del Diálogo puede ser cualquier cosa, incluso una tabla o un formulario. Este ejemplo muestra cómo usar Element Table y Form con Dialog

```
<!-- Table -->
<el-button type="text" @click="dialogTableVisible = true">open a Table nested Dialog</el-button>

<el-dialog title="Shipping address" :visible.sync="dialogTableVisible">
  <el-table :data="gridData">
    <el-table-column property="date" label="Date" width="150"></el-table-column>
    <el-table-column property="name" label="Name" width="200"></el-table-column>
    <el-table-column property="address" label="Address"></el-table-column>
  </el-table>
</el-dialog>

<!-- Form -->
<el-button type="text" @click="dialogFormVisible = true">open a Form nested Dialog</el-button>

<el-dialog title="Shipping address" :visible.sync="dialogFormVisible">
  <el-form :model="form">
    <el-form-item label="Promotion name" :label-width="formLabelWidth">
      <el-input v-model="form.name" autocomplete="off"></el-input>
    </el-form-item>
    <el-form-item label="Zones" :label-width="formLabelWidth">
      <el-select v-model="form.region" placeholder="Please select a zone">
        <el-option label="Zone No.1" value="shanghai"></el-option>
        <el-option label="Zone No.2" value="beijing"></el-option>
      </el-select>
    </el-form-item>
  </el-form>
  <span slot="footer" class="dialog-footer">
    <el-button @click="dialogFormVisible = false">Cancel</el-button>
    <el-button type="primary" @click="dialogFormVisible = false">Confirm</el-button>
  </span>
</el-dialog>

<script>
export default {
  data() {
    return {
```

```

gridData: [{
  date: '2016-05-02',
  name: 'John Smith',
  address: 'No.1518, Jinshajiang Road, Putuo District'
}, {
  date: '2016-05-04',
  name: 'John Smith',
  address: 'No.1518, Jinshajiang Road, Putuo District'
}, {
  date: '2016-05-01',
  name: 'John Smith',
  address: 'No.1518, Jinshajiang Road, Putuo District'
}, {
  date: '2016-05-03',
  name: 'John Smith',
  address: 'No.1518, Jinshajiang Road, Putuo District'
}],
dialogTableVisible: false,
dialogFormVisible: false,
form: {
  name: '',
  region: '',
  date1: '',
  date2: '',
  delivery: false,
  type: [],
  resource: '',
  desc: ''
},
formLabelWidth: '120px'
};
}
};
</script>

```

Diálogo anidado

Si un diálogo está anidado en otro diálogo, se requiere `append-to-body`.

:::demo Normalmente no recomendamos el uso de Dialog anidado. Si necesita que se muestren múltiples diálogos en la página, puede simplemente aplanarlos para que sean hermanos entre sí. Si debe anidar un Diálogo dentro de otro Diálogo, establezca `append-to-body` del Diálogo anidado como `true`, y lo añadirá al cuerpo en lugar de su nodo padre, para que ambos Diálogos puedan ser correctamente renderizados.

```
<template>
```

```

<el-button type="text" @click="outerVisible = true">open the outer Dialog</el-button>

<el-dialog title="Outer Dialog" :visible.sync="outerVisible">
  <el-dialog
    width="30%"
    title="Inner Dialog"
    :visible.sync="innerVisible"
    append-to-body>
  </el-dialog>
  <div slot="footer" class="dialog-footer">
    <el-button @click="outerVisible = false">Cancel</el-button>
    <el-button type="primary" @click="innerVisible = true">open the inner Dialog</el-button>
  </div>
</el-dialog>
</template>

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

```

Contenido centrado

El contenido de Diálogo se puede centrar.

:::demo Ajuste **center** en **true** para centrar el encabezado y el pie de página del cuadro de diálogo horizontalmente. **center** sólo afecta al encabezado y pie de página de Dialog. El cuerpo de Dialog puede ser cualquier cosa, así que a veces no se ve bien cuando está centrado. Necesitas escribir algún CSS si deseas centrar el cuerpo también.

```

<el-button type="text" @click="centerDialogVisible = true">Click to open the Dialog</el-button>

<el-dialog
  title="Warning"
  :visible.sync="centerDialogVisible"
  width="30%"
  center>
  <span>It should be noted that the content will not be aligned in center by default</span>
</el-dialog>

```

```

    <span slot="footer" class="dialog-footer">
      <el-button @click="centerDialogVisible = false">Cancel</el-button>
      <el-button type="primary" @click="centerDialogVisible = false">Confirm</el-button>
    </span>
  </el-dialog>

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

```

El contenido de Dialog se renderiza en modo lazy, lo que significa que la ranura por defecto no se renderiza en el DOM hasta que se abre por primera vez. Por lo tanto, si necesita realizar una manipulación DOM o acceder a un componente mediante ref, hágalo en el callback del evento **open**.

Si la variable ligada a **visible** se gestiona en el Vuex store, el **.sync** no puede funcionar correctamente. En este caso, elimine el modificador **.sync**, escuche los eventos de **open** y **close** Dialog, y confirme las mutaciones Vuex para actualizar el valor de esa variable en los manejadores de eventos.

Atributo

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
visible	visibilidad del Diálogo, apoya el modificador .sync	boolean	—	false
title	título de Diálogo. También se puede pasar con un slot con nombre (ver la tabla siguiente)	string	—	—
width	anchura de Diálogo	string	—	50%
fullscreen	si el diálogo ocupa pantalla completa	boolean	—	false
top	valor de margin-top del Diálogo CSS	string	—	15vh

Atributo	Descripción	Tipo	Valores acepta- dos	Por de- fecto
modal	si se muestra una máscara	boolean	—	true
modal-append-to-body	si adjuntar modal al elemento de cuerpo. Si es falso, el modal se agregará al elemento principal de Diálogo	boolean	—	true
append-to-body	Si adjuntar el cuadro de diálogo al cuerpo	boolean	—	false
lock-scroll	Si el scroll del cuerpo está desactivado mientras se muestra el cuadro de diálogo	boolean	—	true
custom-class	nombres de clase personalizada para el Diálogo	string	—	—
close-on-click-modal	si el Diálogo puede ser cerrado haciendo clic en la máscara	boolean	—	true
close-on-press-escape	si el Diálogo puede ser cerrado presionando ESC	boolean	—	true
show-close	si mostrar un botón de cerrar	boolean	—	true
before-close	una devolución de llamada antes de que se cierre el cuadro de diálogo, y evitar cerrar el cuadro de diálogo	función(done) done se usa para cerrar el diálogo	—	—
center	si alinear el encabezado y el pie de página en el centro	boolean	—	false
destroy-on-close	Destruir elementos en Dialog cuando se cierra	boolean	—	false

Slots

Nombre	Descripción
—	contenido de Diálogo
title	contenido del título de Diálogo
footer	contenido del pie de página de Diálogo

Eventos

Nombre de Evento	Descripción	Parámetros
open	se activa cuando se abre el cuadro de Diálogo	—
opened	se activa cuando la animación de apertura del Dialog termina.	—
close	se dispara cuando el Diálogo se cierra	—
closed	se activa cuando finaliza la animación de cierre del Diálogo	—