

## Cargando

Se muestra la animación mientras se cargan los datos.

### Cargando dentro de un contenedor

Muestra una animación en un contenedor (como en una tabla) mientras se cargan los datos.

El demo Element provee dos maneras para invocar el componente de Cargando: por directiva y por servicio. Para la directiva personalizada `v-loading`, solo necesitas enlazarlo a un valor `Boolean`. Por defecto, la máscara de carga se agregará al elemento donde se usa la directiva. Al agregar el modificador `body`, la máscara se agrega al elemento body.

```
<template>
  <el-table
    v-loading="loading"
    :data="tableData"
    style="width: 100%">
    <el-table-column
      prop="date"
      label="Fecha"
      width="180">
    </el-table-column>
    <el-table-column
      prop="name"
      label="Nombre"
      width="180">
    </el-table-column>
    <el-table-column
      prop="address"
      label="Dirección">
    </el-table-column>
  </el-table>
</template>

<style>
  body {
    margin: 0;
  }
</style>

<script>
  export default {
    data() {
      return {
        tableData: [{
          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'
      }],
      loading: true
    };
  }
};
</script>

```

⋮

## Personalización

Puede personalizar el texto de carga, spinner de carga y color de fondo.

⋮:demo Agregue el atributo `element-loading-text` al elemento en el que `v-loading` está vinculado, y su valor se mostrará debajo del spinner. Del mismo modo, `element-loading-spinner` y `element-loading-background` son para personalizar el nombre de la clase del spinner y el color de fondo.

```

<template>
  <el-table
    v-loading="loading"
    element-loading-text="Loading..."
    element-loading-spinner="el-icon-loading"
    element-loading-background="rgba(0, 0, 0, 0.8)"
    :data="tableData"
    style="width: 100%">
    <el-table-column
      prop="date"
      label="Fecha"
      width="180">
    </el-table-column>
    <el-table-column
      prop="name"
      label="Nombre"
      width="180">
    </el-table-column>
    <el-table-column
      prop="address"
      label="Dirección">
    </el-table-column>
  </el-table>
</template>

<script>
  export default {
    data() {
      return {
        tableData: [{

```

```

        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'
      }
    ],
    loading: true
  };
}
};
</script>

```

...

## Cargando a pantalla completa

Muestra una animación de pantalla completa mientras se cargan los datos

:::demo Cuando se utiliza como una directiva, la carga a pantalla completa requiere el modificador `fullscreen`, y este puede ser agregado al `body`. En este caso, si desea deshabilitar el desplazamiento en `body`, puede agregar otro modificador `lock`. Cuando se utiliza como un servicio, el componente puede ser mostrado a pantalla completa por defecto.

```

<template>
  <el-button
    type="primary"
    @click="openFullScreen1"
    v-loading.fullscreen.lock="fullscreenLoading">
    Como directiva
  </el-button>
  <el-button
    type="primary"
    @click="openFullScreen2">
    Como servicio
  </el-button>
</template>

<script>
export default {
  data() {
    return {
      fullscreenLoading: false
    }
  },
  methods: {
    openFullScreen1() {

```

```

    this.fullscreenLoading = true;
    setTimeout(() => {
      this.fullscreenLoading = false;
    }, 2000);
  },
  openFullScreen2() {
    const loading = this.$loading({
      lock: true,
      text: 'Loading',
      spinner: 'el-icon-loading',
      background: 'rgba(0, 0, 0, 0.7)'
    });
    setTimeout(() => {
      loading.close();
    }, 2000);
  }
}
}
</script>

```

⋮

## Servicio

Puede invocar el componente con un servicio. Importe el servicio:

```
import { Loading } from 'element-ui';
```

Invocar:

```
Loading.service(options);
```

El parámetro `options` es la configuración del componente, y estos detalles pueden ser encontrados en la siguiente table. `LoadingService` devuelve una instancia del componente, y puede cerrarlo invocando el método `close`:

```
let loadingInstance = Loading.service(options);
loadingInstance.close();
```

Tenga en cuenta que, en este caso, el componente a pantalla completa es una instancia única. Si un nuevo componente de pantalla completa es invocado antes de que se cierre la existente, se devolverá la instancia existente en lugar de crear la otra instancia:

```
let loadingInstance1 = Loading.service({ fullscreen: true });
let loadingInstance2 = Loading.service({ fullscreen: true });
console.log(loadingInstance1 === loadingInstance2); // true
```

Llamar al método `close` en cualquiera de estas puede cerrarlo.

Si Element es importado completamente, un método global `$loading` puede ser registrado a `Vue.prototype`. Puede invocarlo como esto: `this.$loading(options)` , y también devuelve una instancia del componente.

Opciones

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
target	el nodo del DOM que el componente debe cubrir. Acepta un objeto DOM o una cadena. Si está es una cadena, este será pasado a <code>document.querySelector</code> para obtener el correspondiente nodo del DOM	object/string	—	document.body
body	igual que el modificador <code>body</code> de <code>v-loading</code>	boolean	—	false
fullscreen	igual que el modificador <code>fullscreen</code> de <code>v-loading</code>	boolean	—	true
lock	igual que el modificador <code>lock</code> de <code>v-loading</code>	boolean	—	false
text	texto de cargando que se muestra debajo del spinner	string	—	—
spinner	nombre de clase del spinner personalizado	string	—	—
background	color de fondo de la máscara	string	—	—
customClass	nombre de clase personalizada para el componente	string	—	—