

Carga de archivos

Carga archivos haciendo clic o arrastrándolos.

Clic para cargar archivos

Personalice el tipo y texto del botón utilizando la propiedad `slot`. Defina las propiedades `limit` y `on-exceed` para limitar el número máximo de archivos a subir y especifique un método para cuando el límite ha sido excedido.

```
<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-preview="handlePreview"
  :on-remove="handleRemove"
  multiple
  :limit="3"
  :on-exceed="handleExceed"
  :file-list="fileList">
  <el-button size="small" type="primary">Clic para subir archivo</el-button>
  <div slot="tip" class="el-upload__tip">Solo archivos jpg/png con un tamaño menor
de 500kb</div>
</el-upload>
<script>
  export default {
    data() {
      return {
        fileList: [{name: 'food.jpeg', url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'}, {name: 'food2.jpeg', url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'}]
      };
    },
    methods: {
      handleRemove(file, fileList) {
        console.log(file, fileList);
      },
      handlePreview(file) {
        console.log(file);
      },
      handleExceed(files, fileList) {
        this.$message.warning(`El límite es 3, haz seleccionado ${files.length}
archivos esta vez, añade hasta ${files.length + fileList.length}`);
      }
    }
  }
</script>
```

...

Cargar avatar de usuario

Utilice el hook `before-upload` para limitar el formato de archivo y su tamaño.

:::demo

```
<el-upload
  class="avatar-uploader"
  action="https://jsonplaceholder.typicode.com/posts/"
  :show-file-list="false"
  :on-success="handleAvatarSuccess"
  :before-upload="beforeAvatarUpload">
  
  <i v-else class="el-icon-plus avatar-uploader-icon"></i>
</el-upload>

<style>
.avatar-uploader .el-upload {
  border: 1px dashed #d9d9d9;
  border-radius: 6px;
  cursor: pointer;
  position: relative;
  overflow: hidden;
}
.avatar-uploader .el-upload:hover {
  border-color: #409EFF;
}
.avatar-uploader-icon {
  font-size: 28px;
  color: #8c939d;
  width: 178px;
  height: 178px;
  line-height: 178px;
  text-align: center;
}
.avatar {
  width: 178px;
  height: 178px;
  display: block;
}
</style>

<script>
export default {
  data() {
    return {
      imageUrl: ''
    };
  },
  methods: {
    handleAvatarSuccess(res, file) {
      this.imageUrl = URL.createObjectURL(file.raw);
    }
  }
}
```

```

    },
    beforeAvatarUpload(file) {
      const isJPG = file.type === 'image/jpeg';
      const isLt2M = file.size / 1024 / 1024 < 2;

      if (!isJPG) {
        this.$message.error('La imagen debe estar en formato JPG!');
      }
      if (!isLt2M) {
        this.$message.error('La imagen excede los 2MB!');
      }
      return isJPG && isLt2M;
    }
  }
}
</script>

```

...

Pared de fotografías

Utilice la propiedad `list-type` para cambiar el estilo a un listado de archivos.

:::demo

```

<el-upload
  action="https://jsonplaceholder.typicode.com/posts/"
  list-type="picture-card"
  :on-preview="handlePictureCardPreview"
  :on-remove="handleRemove">
  <i class="el-icon-plus"></i>
</el-upload>
<el-dialog :visible.sync="dialogVisible">
  
</el-dialog>
<script>
  export default {
    data() {
      return {
        dialogImageUrl: '',
        dialogVisible: false
      };
    },
    methods: {
      handleRemove(file, fileList) {
        console.log(file, fileList);
      },
      handlePictureCardPreview(file) {
        this.dialogImageUrl = file.url;
        this.dialogVisible = true;
      }
    }
  }

```

```
}  
</script>
```

...

Personalizar el template thumbnail

Use el `scoped-slot` para cambiar el template por defecto del thumbnail.

...demo

```
<el-upload  
  action="#"  
  list-type="picture-card"  
  :auto-upload="false">  
  <i slot="default" class="el-icon-plus"></i>  
  <div slot="file" slot-scope="{file}">  
      
    <span class="el-upload-list__item-actions">  
      <span  
        class="el-upload-list__item-preview"  
        @click="handlePictureCardPreview(file) "  
      >  
        <i class="el-icon-zoom-in"></i>  
      </span>  
      <span  
        v-if="!disabled"  
        class="el-upload-list__item-delete"  
        @click="handleDownload(file) "  
      >  
        <i class="el-icon-download"></i>  
      </span>  
      <span  
        v-if="!disabled"  
        class="el-upload-list__item-delete"  
        @click="handleRemove(file) "  
      >  
        <i class="el-icon-delete"></i>  
      </span>  
    </div>  
  </el-upload>  
  <el-dialog :visible.sync="dialogVisible">  
      
  </el-dialog>  
<script>  
  export default {  
    data() {  
      return {
```

```

        dialogImageUrl: '',
        dialogVisible: false,
        disabled: false
    };
},
methods: {
    handleRemove(file) {
        console.log(file);
    },
    handlePictureCardPreview(file) {
        this.dialogImageUrl = file.url;
        this.dialogVisible = true;
    },
    handleDownload(file) {
        console.log(file);
    }
}
}
}
</script>

```

:::

Lista de archivos con miniatura

:::demo

```

<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-preview="handlePreview"
  :on-remove="handleRemove"
  :file-list="fileList"
  list-type="picture">
  <el-button size="small" type="primary">Clic para subir archivo</el-button>
  <div slot="tip" class="el-upload__tip">Solo archivos jpg/png con un tamaño menor
de 500kb</div>
</el-upload>
<script>
  export default {
    data() {
      return {
        fileList: [{name: 'food.jpeg', url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'}, {name: 'food2.jpeg', url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'}]
      };
    },
    methods: {
      handleRemove(file, fileList) {
        console.log(file, fileList);
      },
    },
  },

```

```

        handlePreview(file) {
            console.log(file);
        }
    }
}
</script>

```

...

Control de lista de archivos

Utilice el *hook* `on-change` para controlar la funcionalidad de la lista de archivos subidos.

...demo

```

<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-change="handleChange"
  :file-list="fileList">
  <el-button size="small" type="primary">Clic para subir archivo</el-button>
  <div slot="tip" class="el-upload__tip">Solo archivos jpg/png con un tamaño menor
de 500kb</div>
</el-upload>
<script>
  export default {
    data() {
      return {
        fileList: [{
          name: 'food.jpeg',
          url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'
        }, {
          name: 'food2.jpeg',
          url:
'https://fuss10.elemecdn.com/3/63/4e7f3a15429bfda99bce42a18cdd1jpeg.jpeg?
imageMogr2/thumbnail/360x360/format/webp/quality/100'
        }]
      };
    },
    methods: {
      handleChange(file, fileList) {
        this.fileList = fileList.slice(-3);
      }
    }
  }
</script>

```

...

Arrastrar para cargar archivo

Puede arrastrar el archivo dentro de un área en específico para cargar el archivo.

:::demo

```
<el-upload
  class="upload-demo"
  drag
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-preview="handlePreview"
  :on-remove="handleRemove"
  :file-list="fileList"
  multiple>
  <i class="el-icon-upload"></i>
  <div class="el-upload__text">Suelta tu archivo aquí o <em>haz clic para
cargar</em></div>
  <div slot="tip" class="el-upload__tip">Solo archivos jpg/png con un tamaño menor
de 500kb</div>
</el-upload>
```

:::

Carga normal

:::demo

```
<el-upload
  class="upload-demo"
  ref="upload"
  action="https://jsonplaceholder.typicode.com/posts/"
  :auto-upload="false">
  <el-button slot="trigger" size="small" type="primary">Selecciona un archivo</el-
button>
  <el-button style="margin-left: 10px;" size="small" type="success"
@click="submitUpload">Cargar al servidor</el-button>
  <div slot="tip" class="el-upload__tip">Solo archivos jpg/png con un tamaño menor
de 500kb</div>
</el-upload>
<script>
  export default {
    methods: {
      submitUpload() {
        this.$refs.upload.submit();
      }
    }
  }
</script>
```

:::

Atributos

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
action	obligatorio, URL de la petición	string	—	—
headers	cabeceras de la petición	object	—	—
multiple	especifica si se permite subir múltiples archivos	boolean	—	—
data	opciones adicionales de la petición	object	—	—
name	nombre clave del archivo	string	—	file
with-credentials	especifica si enviará cookies	boolean	—	false
show-file-list	especifica si se debe mostrar la lista de archivos cargados	boolean	—	true
drag	se especifica si se activará el modo arrastrar y soltar	boolean	—	false
accept	acepta tipos de archivos , puede no funcionar cuando <code>thumbnail-mode</code> esta en <code>true</code>	string	—	—
on-preview	<i>hook</i> lanzado al hacer clic en los archivos subidos	function(file)	—	—
on-remove	<i>hook</i> lanzado cuando los archivos son eliminados	function(file, fileList)	—	—
on-success	<i>hook</i> lanzado cuando los archivos fueron cargados correctamente	function(response, file, fileList)	—	—
on-error	<i>hook</i> lanzado cuando han ocurrido algunos errores	function(err, file, fileList)	—	—
on-progress	<i>hook</i> lanzado cuando se produce algún progreso	function(event, file, fileList)	—	—
on-change	<i>hook</i> lanzado cuando el archivo seleccionado se carga correctamente o falla	function(file, fileList)	—	—
before-upload	<i>hook</i> lanzado antes de que el archivo sea cargado. Si este devuelve <code>true</code> o <code>Promise</code> entonces será rechazado, la carga puede ser cancelada	function(file)	—	—
before-remove	<i>hook</i> lanzado antes de eliminar un archivo. Los parámetros son el archivo y la lista de archivos. Si se	function(file, fileList)	—	—

	devuelve <code>false</code> o se devuelve una <code>Promise</code> y que luego es rechazada, la eliminación será abortada.			
thumbnail-mode	especifica si se mostrará la miniatura	boolean	—	false
file-list	archivos cargados por defecto, por ejemplo, [{name: 'food.jpg', url: 'https://xxx.cdn.com/xxx.jpg'}]	array	—	[]
list-type	tipo de lista de archivos	string	text/picture/picture-card	text
auto-upload	se especifica si se auto cargan archivos	boolean	—	true
http-request	sobrescribe el comportamiento por defecto de xhr, permitiendo implementar tu propia petición de carga de archivos	function	—	—
disabled	especifica si se deshabilita la carga de archivos	boolean	—	false
limit	número máximo de cargas permitidas	number	—	—
on-exceed	<i>hook</i> lanzado cuando el límite ha sido excedido	function(files, fileList)	—	-

Slot

Nombre	Descripción
trigger	contenido que desencadena el diálogo de archivo
tip	contenido de los tips

Métodos

Nombre del método	Descripción	Parámetros
clearFiles	limpia la lista de archivos cargados (este método no esta soportado en el <i>hook before-upload</i>)	—
abort	cancela la petición de carga	(file: fileList's item)
submit	La lista de archivos se sube manualmente	—