

Upload

Permet d'uploader des fichiers en cliquant ou en les déplaçant sur le composant.

Cliquer pour envoyer

:::demo Personnalisez le bouton d'envoi avec `slot` . Utilisez `limit` et `on-exceed` pour limiter le nombre maximal de fichiers et déterminer le comportement quand ce nombre est dépassé. De plus, vous pouvez annuler la suppression d'un fichier avec `before-remove` .

```
<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-preview="handlePreview"
  :on-remove="handleRemove"
  :before-remove="beforeRemove"
  multiple
  :limit="3"
  :on-exceed="handleExceed"
  :file-list="fileList">
  <el-button size="small" type="primary">Cliquer pour envoyer</el-button>
  <div slot="tip" class="el-upload__tip">Fichiers jpg/png avec une taille inférieure
à 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(`La limite est 3, vous avez choisi ${files.length}
fichiers, soit ${files.length + fileList.length} au total.`);
      },
      beforeRemove(file, fileList) {
        return this.$confirm(`Supprimer le transfert de ${ file.name } ?`);
      }
    }
  }
</script>
```

...

Envoi d'avatar utilisateur

Utilisez `before-upload` pour restreindre le format et la taille du fichier à envoyer.

...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.$message.error('L\'avatar doit être en JPG !');
    }
    if (!isLt2M) {
      this.$message.error('L\'avatar ne peut pas excéder 2Mb !');
    }
    return isJPG && isLt2M;
  }
}
}
</script>

```

:::

Mur de photos

Utilisez `list-type` pour changer le style de la liste de fichiers.

:::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>

```

⋮

Custom file thumbnail

Use `scoped-slot` to change default thumbnail template.

⋮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>
    </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>

```

...

Liste de fichiers avec miniatures

:::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">Cliquer pour envoyer</el-button>
  <div slot="tip" class="el-upload__tip">Fichiers jpg/png avec une taille inférieure
à 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>

```

...

Contrôle de la liste de fichiers

Utilisez `on-change` pour contrôler le comportement de la liste de fichiers.

:::demo

```

<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-change="handleChange"
  :file-list="fileList">
  <el-button size="small" type="primary">Cliquer pour envoyer</el-button>
  <div slot="tip" class="el-upload__tip">Fichiers jpg/png avec une taille inférieure
à 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>

```

...

Envoi en déposant

Vous pouvez déposer les fichiers par drag'n drop sur l'espace dédié.

...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">Déposer les fichiers ici ou<em>cliquez pour
envoyer</em></div>
  <div class="el-upload__tip" slot="tip">Fichiers jpg/png avec une taille inférieure
à 500kb</div>
</el-upload>
```

...

Envoi manuel

...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">Choisir un fichier</el-
button>
  <el-button style="margin-left: 10px;" size="small" type="success"
@click="submitUpload">Envoyer au serveur</el-button>
  <div class="el-upload__tip" slot="tip">Fichiers jpg/png avec une taille inférieure
à 500kb</div>
</el-upload>
<script>
  export default {
    methods: {
      submitUpload() {
        this.$refs.upload.submit();
      }
    }
  }
</script>
```

...

Attributs

Attribut	Description	Type	Valeurs acceptées	Défaut
action	Requis, l'url de requête.	string	—	—
headers	Les headers de la requête.	object	—	—
multiple	Si envoyer de multiples fichiers est autorisé.	boolean	—	—
data	Options additionnelles de la requête.	object	—	—
name	Identifiant du fichier.	string	—	file
with-credentials	Si les cookies sont aussi envoyés.	boolean	—	false
show-file-list	Si la liste des fichiers est affichée.	boolean	—	true
drag	Si le mode drag'n drop est activé.	boolean	—	false
accept	Types de fichiers acceptés, ne marche pas si <code>thumbnail-mode</code> est <code>true</code> .	string	—	—
on-preview	Fonction pour quand le fichier est cliqué.	function(file)	—	—
on-remove	Fonction pour quand des fichiers sont supprimés.	function(file, fileList)	—	—
on-success	Fonction pour quand l'upload a réussi.	function(response, file, fileList)	—	—
on-error	Fonction pour quand l'upload renvoi une erreur.	function(err, file, fileList)	—	—
on-progress	Fonction pour quand l'upload est en cours.	function(event, file, fileList)	—	—
on-change	Fonction pour quand le statut d'un fichier change (sélection, upload, erreur...).	function(file, fileList)	—	—
before-upload	Fonction pour avant l'upload. Si <code>false</code> est retourné ou une <code>Promise</code> qui est ensuite rejetée, l'envoi sera annulé.	function(file)	—	—
before-remove	Fonction pour avant la suppression de fichiers. Si <code>false</code> est retourné ou une <code>Promise</code> qui est ensuite rejetée, la suppression sera ensuite annulée.	function(file, fileList)	—	—

thumbnail-mode	Si les miniatures sont affichées.	boolean	—	false
file-list	Les fichiers envoyés par défaut, e.g. [{name: 'food.jpg', url: 'https://xxx.cdn.com/xxx.jpg'}]	array	—	[]
list-type	Le type de liste de fichiers.	string	text/picture/picture-card	text
auto-upload	Si l'envoi est automatique.	boolean	—	true
http-request	Écrase le xhr par défaut, afin que vous puissiez implémenter votre propre fonction d'envoi.	function	—	—
disabled	Si le composant est désactivé.	boolean	—	false
limit	Nombre maximum d'envoi autorisés.	number	—	—
on-exceed	Fonction pour quand la limite d'envoi est dépassée.	function(files, fileList)	—	-

Slot

Nom	Description
trigger	Le contenu qui déclenche le Dialog du fichier.
tip	Le contenu des tips.

Méthodes

Méthode	Description	Paramètres
clearFiles	Efface la liste (non supporté dans <code>before-upload</code>).	—
abort	Annule l'envoi en cours.	(file: fileList's item)
submit	Envoi la liste de fichiers manuellement.	—