

Upload

Upload files by clicking or drag-and-drop

Click to upload files

demo Customize upload button type and text using `slot`. Set `limit` and `on-exceed` to limit the maximum number of uploads allowed and specify method when the limit is exceeded. Plus, you can abort removing a file in the `before-remove` hook.

```
<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">Click to upload</el-button>
  <div slot="tip" class="el-upload__tip">jpg/png files with a size less than
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(`The limit is 3, you selected ${files.length} files
this time, add up to ${files.length + fileList.length} totally`);
      },
      beforeRemove(file, fileList) {
        return this.$confirm(`Cancel the transfert of ${ file.name } ?`);
      }
    }
  }
</script>
```

...

User avatar upload

Use `before-upload` hook to limit the upload file format and size.

...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.$message.error('Avatar picture must be JPG format!');
    }
    if (!isLt2M) {
      this.$message.error('Avatar picture size can not exceed 2MB!');
    }
    return isJPG && isLt2M;
  }
}
}
</script>

```

⋮

Photo Wall

Use `list-type` to change the fileList style.

⋮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>

```

...

FileList with thumbnail

:::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">Click to upload</el-button>
  <div slot="tip" class="el-upload__tip">jpg/png files with a size less than
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>

```

...

File list control

Use `on-change` hook function to control upload file list

:::demo

```

<el-upload
  class="upload-demo"
  action="https://jsonplaceholder.typicode.com/posts/"
  :on-change="handleChange"
  :file-list="fileList">
  <el-button size="small" type="primary">Click to upload</el-button>
  <div slot="tip" class="el-upload__tip">jpg/png files with a size less than
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>

```

...

Drag to upload

You can drag your file to a certain area to upload it.

...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">Drop file here or <em>click to upload</em></div>
  <div class="el-upload__tip" slot="tip">jpg/png files with a size less than
500kb</div>
</el-upload>
```

...

Manual upload

...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">select file</el-button>
  <el-button style="margin-left: 10px;" size="small" type="success"
@click="submitUpload">upload to server</el-button>
  <div class="el-upload__tip" slot="tip">jpg/png files with a size less than
500kb</div>
</el-upload>
<script>
  export default {
    methods: {
      submitUpload() {
        this.$refs.upload.submit();
      }
    }
  }
</script>
```

...

Attributes

Attribute	Description	Type	Accepted Values	Default
action	required, request URL	string	—	—
headers	request headers	object	—	—
multiple	whether uploading multiple files is permitted	boolean	—	—
data	additions options of request	object	—	—
name	key name for uploaded file	string	—	file
with-credentials	whether cookies are sent	boolean	—	false
show-file-list	whether to show the uploaded file list	boolean	—	true
drag	whether to activate drag and drop mode	boolean	—	false
accept	accepted file types , will not work when <code>thumbnail-mode</code> is <code>true</code>	string	—	—
on-preview	hook function when clicking the uploaded files	function(file)	—	—
on-remove	hook function when files are removed	function(file, fileList)	—	—
on-success	hook function when uploaded successfully	function(response, file, fileList)	—	—
on-error	hook function when some errors occurs	function(err, file, fileList)	—	—
on-progress	hook function when some progress occurs	function(event, file, fileList)	—	—
on-change	hook function when select file or upload file success or upload file fail	function(file, fileList)	—	—
before-upload	hook function before uploading with the file to be uploaded as its parameter. If <code>false</code> is returned or a <code>Promise</code> is returned and then is rejected, uploading will be aborted	function(file)	—	—
before-remove	hook function before removing a file with the file and file list as its parameters. If <code>false</code> is returned or a <code>Promise</code> is returned and then is rejected, removing will be aborted.	function(file, fileList)	—	—
thumbnail-	whether thumbnail is displayed	boolean	—	false

mode				
file-list	default uploaded files, e.g. [{name: 'food.jpg', url: 'https://xxx.cdn.com/xxx.jpg'}]	array	—	[]
list-type	type of fileList	string	text/picture/picture-card	text
auto-upload	whether to auto upload file	boolean	—	true
http-request	override default xhr behavior, allowing you to implement your own upload-file's request	function	—	—
disabled	whether to disable upload	boolean	—	false
limit	maximum number of uploads allowed	number	—	—
on-exceed	hook function when limit is exceeded	function(files, fileList)	—	-

Slot

Name	Description
trigger	content which triggers file dialog
tip	content of tips

Methods

Methods Name	Description	Parameters
clearFiles	clear the uploaded file list (this method is not supported in the <code>before-upload</code> hook)	—
abort	cancel upload request	(file: fileList's item)
submit	upload the file list manually	—