

文件选择上传和拖拽上传控件。

何时使用

上传是将信息（网页、文字、图片、视频等）通过网页或者上传工具发布到远程服务器上的过程。

- 当需要上传一个或一些文件时。
- 当需要展现上传的进度时。
- 当需要使用拖拽交互时。

API

参数	说明	类型	默认值	版本
accept	接受上传的文件类型, 详见 input accept Attribute	string	-	
action	上传的地址	string (file) => Promise<string>	-	
beforeUpload	上传文件之前的钩子, 参数为上传的文件, 若返回 <code>false</code> 则停止上传。支持返回一个 Promise 对象, Promise 对象 reject 时则停止上传, resolve 时开始上传 (resolve 传入 <code>File</code> 或 <code>Blob</code> 对象则上传 resolve 传入对象); 也可以返回 <code>Upload.LIST_IGNORE</code> , 此时列表中将不展示此文件。 注意: IE9 不支持该方法	(file, fileList) => boolean Promise<File> Upload.LIST_IGNORE	-	
customRequest	通过覆盖默认的上传行为, 可以自定义自己的上传实现	function	-	
data	上传所需额外参数或返回上传额外参数的方法	object (file) => object Promise<object>	-	
defaultFileList	默认已经上传的文件列表	object[]	-	
directory	支持上传文件夹 (caniuse)	boolean	false	
disabled	是否禁用	boolean	false	
fileList	已经上传的文件列表 (受控), 使用此参数时, 如果遇到 <code>onChange</code> 只调用一次的问题, 请参考 #2423	UploadFile[]	-	

headers	设置上传的请求头部, IE10 以上有效	object	-	
iconRender	自定义显示 icon	(file: UploadFile, listType?: UploadListType) => ReactNode	-	
isImageUrl	自定义缩略图是否使用 <code></code> 标签进行显示	(file: UploadFile) => boolean	(内部实现)	
itemRender	自定义上传列表项	(originNode: ReactElement, file: UploadFile, fileList: object[], actions: { download: function, preview: function, remove: function }) => React.ReactNode	-	4.16.0
listType	上传列表的内建样式, 支持三种基本样式 <code>text</code> , <code>picture</code> 和 <code>picture-card</code>	string	text	
maxCount	限制上传数量。当为 1 时, 始终用最新上传的文件代替当前文件	number	-	4.10.0
method	上传请求的 http method	string	post	
multiple	是否支持多选文件, ie10+ 支持。开启后按住 ctrl 可选择多个文件	boolean	false	
name	发到后台的文件参数名	string	file	
openFileDialogOnClick	点击打开文件对话框	boolean	true	
previewFile	自定义文件预览逻辑	(file: File Blob) => Promise<dataURL: string>	-	
progress	自定义进度条样式	ProgressProps (仅支持 <code>type="line"</code>)	{ strokeWidth: 2, showInfo: false }	4.3.0
showUploadList	是否展示文件列表, 可设为一个对象, 用于单独设定 <code>showPreviewIcon</code> , <code>showRemoveIcon</code> , <code>showDownloadIcon</code> , <code>removeIcon</code> 和 <code>downloadIcon</code>	boolean { showPreviewIcon?: boolean, showRemoveIcon?: boolean, showDownloadIcon?: boolean, previewIcon?: ReactNode (file:	true	function: 4.7.0

		UploadFile) => ReactNode, removeIcon?: ReactNode (file: UploadFile) => ReactNode, downloadIcon?: ReactNode (file: UploadFile) => ReactNode }		
withCredentials	上传请求时是否携带 cookie	boolean	false	
onChange	上传文件改变时的状态, 详见 onChange	function	-	
onDrop	当文件被拖入上传区域时执行的回调功能	(event: React.DragEvent) => void	-	4.16.0
onDownload	点击下载文件时的回调, 如果没有指定, 则默认跳转到文件 url 对应的标签页	function(file): void	(跳转新标签页)	
onPreview	点击文件链接或预览图标时的回调	function(file)	-	
onRemove	点击移除文件时的回调, 返回值为 false 时不移除。支持返回一个 Promise 对象, Promise 对象 resolve(false) 或 reject 时不移除	function(file): boolean Promise	-	

UploadFile

继承自 File，附带额外属性用于渲染。

参数	说明	类型	默认值
name	文件名	string	-
percent	上传进度	number	-
status	上传状态，不同状态展示颜色也会有所不同	error success done uploading removed	-
thumbUrl	缩略图地址	string	-
uid	唯一标识符，不设置时会自动生成	string	-
url	下载地址	string	-

onChange

上传中、完成、失败都会调用这个函数。

文件状态改变的回调，返回为：

```
{
  file: { /* ... */ },
  fileList: [ /* ... */ ],
  event: { /* ... */ },
}
```

1. `file` 当前操作的文件对象。

```
{
  uid: 'uid',          // 文件唯一标识，建议设置为负数，防止和内部产生的 id 冲突
  name: 'xx.png'       // 文件名
  status: 'done',     // 状态有: uploading done error removed, 被 beforeUpload 拦截
                    // 的文件没有 status 属性
  response: '{"status": "success"}', // 服务端响应内容
  linkProps: '{"download": "image"}', // 下载链接额外的 HTML 属性
}
```

2. `fileList` 当前的文件列表。
3. `event` 上传中的服务端响应内容，包含了上传进度等信息，高级浏览器支持。

FAQ

服务端如何实现？

- 服务端上传接口实现可以参考 [jQuery-File-Upload](#)。
- 如果要做本地 mock 可以参考这个 [express 的例子](#)。

如何显示下载链接？

请使用 `fileList` 属性设置数组项的 `url` 属性进行展示控制。

`customRequest` 怎么使用？

请参考 <https://github.com/react-component/upload#customrequest>。

为何 `fileList` 受控时，上传不在列表中的文件不会触发 `onChange` 后续的 `status` 更新事件？

`onChange` 事件仅会作用于在列表中的文件，因而 `fileList` 不存在对应文件时后续事件会被忽略。请注意，在 4.13.0 版本之前受控状态存在 bug 导致不在列表中的文件也会触发。

`onChange` 为什么有时候返回 `File` 有时候返回 `{ originFileObj: File }`？

历史原因，在 `beforeUpload` 返回 `false` 时，会返回 `File` 对象。在下个大版本我们会统一返回 `{ originFileObj: File }` 对象。当前版本已经兼容所有场景下 `info.file.originFileObj` 获取原 `File` 写

法。你可以提前切换。