

getUploadedFiles

2.2.1

(path) (isDeleted) .
[afterUploadFile](#), [afterUploadImage](#), [afterUploadVideo](#) [addUploadPath](#) .

Parameters:

Name	Type	Description
[fileType]	string	. ("image", "video", "file")

Return:

[fileType]

Type	Attribute	Description
Array	<pre>[{ path: "/upload/path/filename.png", isDeleted: false }, { path: "/upload/path/filename2.png", isDeleted: true }, //...]</pre>	path: isDeleted:

[fileType]

Type	Attribute	Description
Object	<pre>{ image: [{ path: "/upload/path/filename.png", isDeleted: false }, { path: "/upload/path/filename2.png", isDeleted: true }, //...], video: [], file: [] }</pre>	path: isDeleted:

Example:

```
var filesData = editor.getUploadedFiles();  
var filesData2 = editor.getUploadedFiles('image');
```

1.

API .

```

var editorId = 'synapEditor';
var editorConfig = {};
var html = '';
var editor = new SynapEditor(editorId, editorConfig, html);

editor.setEventListener('afterUploadImage', function (e) {
    var fileType = e.fileType;
    var uploadPath = e.path;
    e.editor.addUploadPath(fileType, uploadPath); // editor.getUploadedFiles()
});

editor.setEventListener('beforeUploadImage', function (e) {
    var fileType = e.fileType;
    var uploadCount = e.uploadCount; // ( )
    var imageCount = 0;
    e.editor.getUploadedFiles(fileType).forEach(function (info) {
        if (!info.isDeleted) {
            imageCount++; //
        }
    });
    if (imageCount + uploadCount >= 5) { // 5
        e.returnValue = false; // returnValue false
        alert(' 5 ');
    }
});

```

API

- [getUploadedFiles](#)
- [addUploadPath](#)