# Events


- Event Type
- Adding Events
  - Using API
  - Adding at the Editor Initialization
  - Using Functions
- Removing Events
  - Using API
- Object Delivered by Functions
- Cancelling Events


---

## Event Type

| Event Name | Version | Description |
|---|---|---|
| initialized | RELEASE 2.1.0 OR ABOVE | This event occurs when the Editor is initialized. |
| beforeUploadImage | RELEASE 2.2.0 OR ABOVE | This event occurs before an image is uploaded. |
| afterUploadImage | RELEASE 2.2.0 OR ABOVE | This event occurs after an image is uploaded. |
| beforeUploadVideo | RELEASE 2.2.0 OR ABOVE | This event occurs before a video is uploaded. |
| afterUploadVideo | RELEASE 2.2.0 OR ABOVE | This event occurs after a video is uploaded. |
| beforeUploadFile | RELEASE 2.2.0 OR ABOVE | This event occurs before a file is uploaded. |
| afterUploadFile | RELEASE 2.2.0 OR ABOVE | This event occurs after a file is uploaded. |
| beforeOpenDocument | RELEASE 2.3.0 OR ABOVE | This event occurs before a document is opened (imported). |
| afterOpenDocument | RELEASE 2.3.0 OR ABOVE | This event occurs after a document is opened (imported). |
| beforeNewDocument | RELEASE 2.4.1 OR ABOVE | This event occurs before creating a new document. |
| afterNewDocument | RELEASE 2.4.1 OR ABOVE | This event occurs after creating a new document. |
| onSelection | RELEASE 2.6.0 OR ABOVE | This event occurs when performing a selection operation such as moving the caret and expanding the selection. |
| beforePaste | RELEASE 2.10.0 OR ABOVE | This event occurs before pasting. |
| afterPaste | RELEASE 2.10.0 OR ABOVE | This event occurs after pasting. |
| afterEdit | RELEASE 2.10.0 OR ABOVE | This event occurs after all editing actions. |

## Adding Events

### Using API

#### editor.setEventListener()

API used to add an event.

```
editor.setEventListener('Event Name', function(e) {
});
```

### Adding at the Editor Initialization

```
var editorId = 'synapEditor';
var editorConfig = {};
var html = '';
var eventListeners = {
    'Event Name': function (e) {
    }
};

new SynapEditor(editorId, editorConfig, html, eventListeners);
```

### Using Functions

```
var editorId = 'synapEditor';
var editorConfig = {};
var html = '';

function SynapEditorEvent Name(e) {
    // The first letter of the event name shall be capitalized
    // ex: initialized => SynapEditorInitialized
    // ex: beforeUploadImage => SynapEditorBeforeUploadImage
}

new SynapEditor(editorId, editorConfig, html);
```

## Removing Events

### Using API

#### editor.removeEventListener()

API used to remove an event..

```
editor.removeEventListener('Event Name');
```

## Object Delivered by Functions

```
{
        editor: {Editor Object}, // Editor
        eventType: 'initialized', // Event Name
        cancelable: false, // Whether the event is cancelable
        returnValue: null, // Return Value
        ....

}
```

- editor: Editor in which the event is occurred.
- eventType: Event name.
- cancelable: Whether the event is cancelable.

- returValue: Return value. As for cancelable events, the events can be cancelled by delivering (cancelable: true) false.

## Cancelling Events

When cancelable is set to true, you may cancel the event by setting returnValue to false.

```
var editorId = 'synapEditor';
var editorConfig = {};
var html = '';
var eventListeners = {
    'beforeUploadImage': function (e) {
                e.returnValue = false; // Upload will be no longer proceed.
    }
};

new SynapEditor(editorId, editorConfig, html, eventListeners);
```