

Пользовательские плагины

Создание плагина

Плагины необходимо добавлять в качестве свойства объекта `sceditor.plugins.*`.

Например, для создания плагина с таким названием `myplugin` будет выглядеть так:

[script.js](#)

```
// Created in sceditor.plugins.[name] object
sceditor.plugins.myplugin = function() {
    // Place signal handlers and functions here
};
```

Чтобы затем включить плагин в редакторе, добавьте его `myplugin` в [раздел плагинов](#) при создании редактора.

 **Важно:** При создании пользовательского плагина это необходимо сделать до создания экземпляра редактора, например, до вызова метода `sceditor.create()`.

Структура пользовательского формата должна быть следующей:

Обработка сигналов

Для создания обработчика сигнала достаточно создать функцию с именем сигнала, и она будет автоматически вызываться всякий раз, когда этот сигнал возникает.

[script.js](#)

```
sceditor.plugins.myplugin = function() {
    base.signalKeydownEvent = function(e) {
        // this will automatically be called when 'myplugin'
        // is registered with an editor instance and there
        // is a keydown event
    };
};
```

Все сигналы будут `this` установлены на тот экземпляр редактора, к которому применяется плагин.

init

Вызывается при регистрации плагина в редакторе.

destroy

Вызывается при `destroy()` вызове соответствующей функции в редакторе или при удалении плагина из редактора.

Этот сигнал следует использовать для отмены привязки любых событий DOM и для выполнения очистки, чтобы браузер мог освободить используемую память.

signalReady

Вызывается после создания редактора.



Важно: Эту функцию можно вызвать до полной загрузки страницы.

signalKeydownEvent

Параметры:

- `e` event Объект события keydown

Вызывается всякий раз, когда keydown событие срабатывает в редакторе в режиме WYSIWYG или в режиме исходного кода.

Используйте эту функцию `this.sourceMode()`, чтобы проверить, в каком режиме работает редактор.

signalKeyupEvent

Параметры:

- `e` event Объект события keyup

Вызывается всякий раз, когда keyup событие срабатывает в редакторе в режиме WYSIWYG или в режиме исходного кода.

Используйте эту функцию `this.sourceMode()`, чтобы проверить, в каком режиме работает редактор.

signalFocusEvent

Параметры:

- e event Объект события focus

Вызывается всякий раз, когда focus событие срабатывает в редакторе в режиме WYSIWYG или в режиме исходного кода.

Используйте эту функцию `this.sourceMode()`, чтобы проверить, в каком режиме работает редактор.

signalBlurEvent

Параметры:

- e event Объект события blur

Вызывается всякий раз, когда blur событие срабатывает в редакторе в режиме WYSIWYG или в режиме исходного кода.

Используйте эту функцию `this.sourceMode()`, чтобы проверить, в каком режиме работает редактор.

signalContextmenuEvent

Параметры:

- e event Объект события contextmenu

Вызывается всякий раз, когда contextmenu событие срабатывает в редакторе в режиме WYSIWYG или в режиме исходного кода.

Используйте эту функцию `this.sourceMode()`, чтобы проверить, в каком режиме работает редактор.

signalSelectionchangedEvent

Параметры:

- e event Объект события selectionchanged

Вызывается всякий раз, когда изменяется выделение в WYSIWYG-редакторе.



Важно: Это событие не происходит в режиме источника.

Дополнения и Файлы

[Ссылка на оригинальную статью](#)

From:
<https://wwoss.ru/> - worldwide open-source software

Permanent link:
https://wwoss.ru/doku.php?id=software:development:web:docs:web:wysiwyg:sceditor_doc_custom_plugins

Last update: **2026/01/07 12:40**

