* [Integration conors](https://api.onlyoffice.com/editors/plugins)
  + [Alfresco integration](https://api.onlyoffice.com/editors/alfresco)
  + [Confluence integration](https://api.onlyoffice.com/editors/confluence)
  + [SharePoint integration](https://api.onlyoffice.com/editors/sharepoint)
* [How it works](https://api.onlyoffice.com/editors/howitworks)
  + [Opening file](https://api.onlyoffice.com/editors/open)
  + [Saving file](https://api.onlyoffice.com/editors/save)
  + [Co-editing](https://api.onlyoffice.com/editors/coedit)
  + [Converting and downloading file](https://api.onlyoffice.com/editors/conversion)
  + [Document history](https://api.onlyoffice.com/editors/history)
  + [Security](https://api.onlyoffice.com/editors/security)

**DOCUMENTATION**

* [Advanced parameters](https://api.onlyoffice.com/editors/advanced)
* [Config](https://api.onlyoffice.com/editors/config/)
  + [Document](https://api.onlyoffice.com/editors/config/document)
    - [Info](https://api.onlyoffice.com/editors/config/document/info)
    - [Permissions](https://api.onlyoffice.com/editors/config/document/permissions)
  + [Editor](https://api.onlyoffice.com/editors/config/editor)
    - [Embedded](https://api.onlyoffice.com/editors/config/editor/embedded)
    - [Plugins](https://api.onlyoffice.com/editors/config/editor/plugins)
  + [Events](https://api.onlyoffice.com/editors/config/events)
* [Methods](https://api.onlyoffice.com/editors/methods)
* [Callback handler](https://api.onlyoffice.com/editors/callback)
* [Command service](https://api.onlyoffice.com/editors/command)
* [Conversion API](https://api.onlyoffice.com/editors/conversionapi)
* [Signature](https://api.onlyoffice.com/editors/signature/)
  + [browser](https://api.onlyoffice.com/editors/signature/browser)
  + [request](https://api.onlyoffice.com/editors/signature/request)
* [Changelog](https://api.onlyoffice.com/editors/changelog)
* [Troubleshooting](https://api.onlyoffice.com/editors/troubleshooting)

Basic concepts

ONLYOFFICE Document Server API is used to let the developers integrate the ONLYOFFICE Document/Spreadsheet/Presentation Editors into their own web sites and setup and manage the editors.

The API JavaScript file can normally be found in the following editors folder:

**https://documentserver/web-apps/apps/api/documents/api.js**

Where the **documentserver** is the name of the server with the ONLYOFFICE Document Server installed.

For **Document Server** versions prior to 4.0 the path to the API JavaScript file will look like this:   
**https://documentserver/OfficeWeb/apps/api/documents/api.js**

The target HTML file where the editors are to be embedded need to have a placeholder *div* tag, where all the information about the editor parameters will be passed:

<**div** id="placeholder"></**div**>

<**script** type="text/javascript" src="https://documentserver/web-apps/apps/api/documents/api.js"></**script**>

The page code containing the changeable parameters looks the following way:

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", config);

Where *config* is an object:

config = {

"document": {

"fileType": "docx",

"key": "Khirz6zTPdfd7",

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

},

"documentType": "text",

"editorConfig": {

"callbackUrl": "https://example.com/url-to-callback.ashx"

}

};

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

From now the *docEditor* object can be used to call the **document editor** [Methods](https://api.onlyoffice.com/editors/methods).

The example above includes all the parameters necessary for Document Server correct startup. There are additional non-obligatory parameters though which can be changed to achieve different goals with your document (change access rights for the document, display different information about the document, etc.) See the [Advanced parameters](https://api.onlyoffice.com/editors/advanced) section to find out what these parameters are and how you can change them.

# Try now

Click the buttons with the format names below to try the main functionality of ONLYOFFICE Document Server. You can open various file types for editing, viewing, co-editing, review or see how the rebranding works. Use the **</>** buttons to view the corresponding example source code.

|  | **Documents** | **Spreadsheets** | **Presentations** |
| --- | --- | --- | --- |
| **Edit** | [**DOCX**](https://api.onlyoffice.com/editors/editor?method=docxEditor)</>  [**DOC**](https://api.onlyoffice.com/editors/editor?method=docEditor)</>  [**ODT**](https://api.onlyoffice.com/editors/editor?method=odtEditor)</>  [**TXT**](https://api.onlyoffice.com/editors/editor?method=txtEditor)</> | [**XLSX**](https://api.onlyoffice.com/editors/editor?method=xlsxEditor)</>  [**XLS**](https://api.onlyoffice.com/editors/editor?method=xlsEditor)</>  [**ODS**](https://api.onlyoffice.com/editors/editor?method=odsEditor)</>  [**CSV**](https://api.onlyoffice.com/editors/editor?method=csvEditor)</> | [**PPTX**](https://api.onlyoffice.com/editors/editor?method=pptxEditor)</>  [**PPT**](https://api.onlyoffice.com/editors/editor?method=pptEditor)</>  [**ODP**](https://api.onlyoffice.com/editors/editor?method=odpEditor)</> |
| **Co-Edit** | DOCX</> | XLSX</> | PPTX</> |
| **Strict Co-Edit** | DOCX</> | XLSX</> | PPTX</> |
| **Review** | [**DOCX**](https://api.onlyoffice.com/editors/editor?method=docxReview)</> |  |  |
| **Comment** | [**DOCX**](https://api.onlyoffice.com/editors/editor?method=docxComment)</> |  |  |
| **View** | [**DOCX**](https://api.onlyoffice.com/editors/editor?method=docxViewer)</>  [**PDF**](https://api.onlyoffice.com/editors/editor?method=pdfViewer)</> | [**XLSX**](https://api.onlyoffice.com/editors/editor?method=xlsxViewer)</> | [**PPTX**](https://api.onlyoffice.com/editors/editor?method=pptxViewer)</> |
| **Rebranding** | [**DOCX**](https://api.onlyoffice.com/editors/editor?method=rebrandingDocxEditor)</> | [**XLSX**](https://api.onlyoffice.com/editors/editor?method=rebrandingXlsxEditor)</> | [**PPTX**](https://api.onlyoffice.com/editors/editor?method=rebrandingPptxEditor)</> |

How it works

The work with document files in ONLYOFFICE Document Server is quite a simple process but it requires some understanding of what is going on when you click your document link to [open](https://api.onlyoffice.com/editors/open) it in the browser, or [save](https://api.onlyoffice.com/editors/save) the document or [upload](https://api.onlyoffice.com/editors/conversion) it from your computer to the document server.

The user-document interaction is done both at the client side and at the server side.

Below the main notions used throughout the current documentation are explained.

The client side includes:

* **Document manager** - the list of the documents displayed in the user browser where the user can select the necessary document and perform some actions with it (depending on the provided rights, the user can open the document to view it or edit, share the document with other users).
* **Document editor** - the document viewing and editing interface with all the most known document editing features available, used as a medium between the user and the **document editing service**.

The server side includes:

* **Document storage service** - the server service which stores all the documents available to the users with the appropriate access rights. It provides the document IDs and links to these documents to the **document manager** which the user sees in the browser.
* **Document editing service** - the server service which allows to perform the document viewing and editing (in case the user has the appropriate rights to do that). The **document editor** interface is used to access all the **document editing service** features.
* **Document command service** - the server service which allows to perfom additional commands with **document editing service**.
* **Document conversion service** - the server service which allows to convert the document file into the appropriate Office Open XML format (*docx* for text documents, *xlsx* for spreadsheets and *pptx* for presentations) for their editing or downloading.

Please note, that ONLYOFFICE Document Server includes the **document editor**, **document editing service**, **document command service** and **document conversion service**. The **document manager** and **document storage service** are either included to Community Server or must be implemented by the software integrators who use ONLYOFFICE Document Server on their own server.

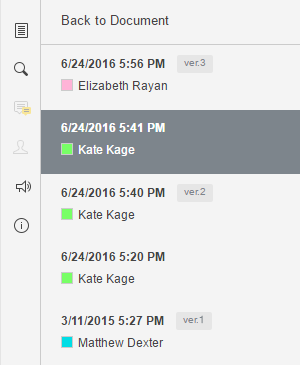
## Document history

If the document has been previously edited using Document Editor, you can view the document history.

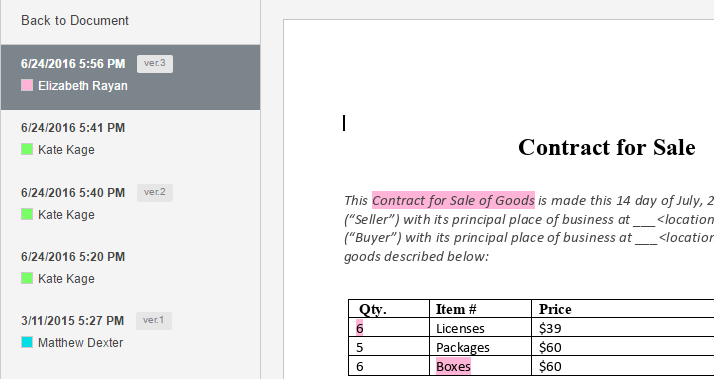
The history is currently available for text document files only.

After editing in Document Editor the information about the changes during the editing session is sent together with the changed document:

* [history](#history) - this information allows to display the time and the author for each document version when you view the document history in the side panel. Must be sent as a property changes of the object sent as the argument to the [refreshHistory](https://api.onlyoffice.com/editors/methods" \l "refreshHistory) method. This method must be called after the [onRequestHistory](https://api.onlyoffice.com/editors/config/events" \l "onRequestHistory) events.



* [changesurl](#changesurl) - the link to the file with the document editing data used to show the changes corresponding to the specific document version. The file must be saved and its address must be sent as changesUrl parameter using the [setHistoryData](https://api.onlyoffice.com/editors/methods" \l "setHistoryData) method. This method must be called after the [onRequestHistoryData](https://api.onlyoffice.com/editors/config/events" \l "onRequestHistoryData) events.



This way the following events must be added to the config file to enable the document changes history display:

* [onRequestHistory](#onRequestHistory) with [refreshHistory](#refreshHistory) which contains document history for each document version if the history parameter has been present for each version.
* [onRequestHistoryData](#onRequestHistoryData) with [setHistoryData](#setHistoryData) which contains the link to the file of the corresponding version, the link to the previous version and the link to the file received using the *changesUrl*.

# Advanced parameters

The parameters, which can be changed for ONLYOFFICE Document Server, can be subdivided into the following main sections:

[**config**](https://api.onlyoffice.com/editors/config/) - allows to change the platform type used, document display size (width and height) and type of the document opened;

* [**document**](https://api.onlyoffice.com/editors/config/document) - contains all the parameters pertaining to the document (title, url, file type, etc.);
  + [**info**](https://api.onlyoffice.com/editors/config/document/info) - contains additional parameters for the document (document author, folder where the document is stored, creation date, sharing settings);
  + [**permissions**](https://api.onlyoffice.com/editors/config/document/permissions) - defines whether the document can be edited and downloaded or not;
* [**editorConfig**](https://api.onlyoffice.com/editors/config/editor) - defines parameters pertaining to the editor interface: opening mode (viewer or editor), interface language, additional buttons, etc.);
  + [**customization**](https://api.onlyoffice.com/editors/config/editor/customization) - allows to customize the editor interface so that it looked like your other products (if there are any) and change the presence or absence of the additional buttons, links, change logos and editor owner details;
  + [**embedded**](https://api.onlyoffice.com/editors/config/editor/embedded) - is used for the embedded document type only and allows to change the behavior of the buttons used to control the embedded mode;
  + [**plugins**](https://api.onlyoffice.com/editors/config/editor/plugins) - is used to connect the necessary [plugins](https://api.onlyoffice.com/plugin/basic) to your Document Server, so that they become visible to all document editor users;
* [**events**](https://api.onlyoffice.com/editors/config/events) - is the list of special events called when some action is applied to the document (when it is loaded, modified, etc.).

The complete *config* with all the additional parameters looks the following way:

config = {

"document": {

"fileType": "docx",

"info": {

"author": "John Smith",

"created": "2010-07-07 3:46 PM",

"folder": "Example Files",

"sharingSettings": [

{

"permissions": "Full Access",

"user": "John Smith"

},

{

"permissions": "Read Only",

"user": "Kate Cage"

},

...

]

},

"key": "Khirz6zTPdfd7",

"permissions": {

"comment": **true**,

"download": **true**,

"edit": **true**,

"print": **true**,

"review": **true**

},

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

},

"documentType": "text",

"editorConfig": {

"callbackUrl": "https://example.com/url-to-callback.ashx",

"createUrl": "https://example.com/url-to-create-document/",

"customization": {

"autosave": **true**,

"chat": **true**,

"commentAuthorOnly": **false**,

"compactToolbar": **false**,

"customer": {

"address": "My City, 123a-45",

"info": "Some additional information",

"logo": "https://example.com/logo-big.png",

"mail": "john@example.com",

"name": "John Smith and Co.",

"www": "example.com"

},

"feedback": {

"url": "https://example.com",

"visible": **true**

},

"forcesave": **false**,

"goback": {

"text": "Go to Documents",

"url": "https://example.com"

},

"logo": {

"image": "https://example.com/logo.png",

"imageEmbedded": "https://example.com/logo\_em.png",

"url": "https://example.com"

},

"showReviewChanges": **false**,

"zoom": 100

},

"embedded": {

"embedUrl": "https://example.com/embedded?doc=exampledocument1.docx",

"fullscreenUrl": "https://example.com/embedded?doc=exampledocument1.docx#fullscreen",

"saveUrl": "https://example.com/download?doc=exampledocument1.docx",

"shareUrl": "https://example.com/view?doc=exampledocument1.docx",

"toolbarDocked": "top"

},

"lang": "en-US",

"mode": "edit",

"plugins": {

"autostart": [

"asc.{0616AE85-5DBE-4B6B-A0A9-455C4F1503AD}",

"asc.{FFE1F462-1EA2-4391-990D-4CC84940B754}",

...

],

"pluginsData": [

"https://example.com/plugin1/config.json",

"https://example.com/plugin2/config.json",

...

]

},

"recent": [

{

"folder": "Example Files",

"title": "exampledocument1.docx",

"url": "https://example.com/exampledocument1.docx"

},

{

"folder": "Example Files",

"title": "exampledocument2.docx",

"url": "https://example.com/exampledocument2.docx"

},

...

],

"user": {

"id": "78e1e841",

"name": "John Smith"

}

},

"events": {

"onAppReady": onAppReady,

"onCollaborativeChanges": onCollaborativeChanges,

"onDocumentReady": onDocumentReady,

"onDocumentStateChange": onDocumentStateChange,

"onDownloadAs": onDownloadAs,

"onError": onError,

"onRequestEditRights": onRequestEditRights,

"onRequestHistory": onRequestHistory,

"onRequestHistoryClose": onRequestHistoryClose,

"onRequestHistoryData": onRequestHistoryData

},

"height": "100%",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.t-IDcSemACt8x4iTMCda8Yhe3iZaWbvV5XKSTbuAn0M",

"type": "desktop",

"width": "100%"

};

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

# Config

Description

The config base section allows to change the platform type used, document display size (width and height) and type of the document opened.

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| documentType | Defines the document type to be opened: open a text document (.doc, .docm, .docx, .dot, .dotm, .dotx, .epub, .fodt, .htm, .html, .mht, .odt, .pdf, .rtf, .txt, .djvu, .xps) for viewing or editing - **text**, open a spreadsheet (.csv, .fods, .ods, .xls, .xlsm, .xlsx, .xlt, .xltm, .xltx) for viewing or editing - **spreadsheet**, open a presentation (.fodp, .odp, .pot, .potm, .potx, .pps, .ppsm, .ppsx, .ppt, .pptm, .pptx) for viewing or editing - **presentation**. | string | "spreadsheet" |
| height | Defines the document height (**100%** by default) in the browser window. | string | "100%" |
| token | Defines the encrypted signature added to the **Document Server** config in the form of a [token](#config). | string | "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.t-IDcSemACt8x4iTMCda8Yhe3iZaWbvV5XKSTbuAn0M" |
| type | Defines the platform type used to access the document. Can be: optimized to access the document from a desktop or laptop computer - **desktop**, optimized to access the document from a tablet or a smartphone - **mobile**, specifically formed to be easily embedded into a web page - **embedded**. The default value is **"desktop"**. | string | "desktop" |
| width | Defines the document width (**100%** by default) in the browser window. | string | "100%" |

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"documentType": "text",

"height": "100%",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.t-IDcSemACt8x4iTMCda8Yhe3iZaWbvV5XKSTbuAn0M",

"type": "desktop",

"width": "100%",

...

});

## Document

Description

The document section allows to change all the parameters pertaining to the document (title, url, file type, etc.).

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| fileType\* | Defines the type of the file for the source viewed or edited document. | string | "docx" |
| key\* | Defines the unique document identifier used for document recognition by the service. In case the known key is sent the document will be taken from the cache. Every time the document is edited and saved, the key must be generated anew. The document url can be used as the **key** but without the special characters and the length is limited to 20 symbols. | string | "Khirz6zTPdfd7" |
| title\* | Defines the desired file name for the viewed or edited document which will also be used as file name when the document is downloaded. | string | "Example Document Title.docx" |
| url\* | Defines the URL where the source viewed or edited document is stored. | string | "https://example.com/url-to-example-document.docx" |

\* - required field

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"document": {

"fileType": "docx",

"key": "Khirz6zTPdfd7",

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx",

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

The key characters can be used: **0-9**, **a-z**, **A-Z**, **-.\_=**. The maximal key length is **20** characters.

### Document Info

Description

The document info section allows to change additional parameters for the document (document author, folder where the document is stored, creation date, sharing settings).

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| author | Defines the name of the document author/creator. | string | "John Smith" |
| created | Defines the document creation date. | string | "2010-07-07 3:46 PM" |
| folder | Defines the folder where the document is stored (can be empty in case the document is stored in the root folder). | string | "Example Files" |
| sharingSettings | Defines the settings which will allow to share the document with other users:   * **permissions** - the access rights for the user with the name above. Can be **Full Access**, **Read Only** or **Deny Access**  **type**: string  **example**: "Full Access" * **user** - the name of the user the document will be shared with  **type**: string  **example**: "John Smith". | array of object |  |
|  | | | |

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"document": {

"info": {

"author": "John Smith",

"created": "2010-07-07 3:46 PM",

"folder": "Example Files",

"sharingSettings": [

{

"permissions": "Full Access",

"user": "John Smith"

},

{

"permissions": "Read Only",

"user": "Kate Cage"

},

...

]

},

...

},

...

});

### Document Permissions

Description

The document permission section allows to change the permission for the document to be edited and downloaded or not.

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| comment | Defines if the document can be commented or not. In case the commenting permission is set to **"true"** the document **side bar** will contain the **Comment**menu option; the document commenting will only be available for the document editor if the [mode](#mode) parameter is set to **edit**. The default value is **true**. | boolean | true |
|  | | | |

In case edit is set to **"true"** and comment is also set to **"true"**, the user will be able to edit the document and comment. In case edit is set to **"true"** and comment is set to **"false"**, the user will be able to edit only, the corresponding commenting functionality will be available for viewing only, the adding and editing of comments will be unavailable. In case edit is set to **"false"** and comment is set to **"true"**, the document will be available for commenting only.

|  |  |  |  |
| --- | --- | --- | --- |
| download | Defines if the document can be downloaded or only viewed or edited online. In case the downloading permission is set to **"false"** the **Download as...** menu option will be absent from the **File** menu. The default value is **true**. | boolean | true |
| edit | Defines if the document can be edited or only viewed. In case the editing permission is set to **"true"** the **File**menu will contain the **Edit Document** menu option; please note that if the editing permission is set to **"false"** the document will be opened in viewer and you will **not** be able to switch it to the editor even if the [mode](#mode) parameter is set to **edit**. The default value is **true**. | boolean | true |
| print | Defines if the document can be printed or not. In case the printing permission is set to **"false"** the **Print** menu option will be absent from the **File** menu. The default value is **true**. | boolean | true |
|  | | | |
| review | Defines if the document can be reviewed or not. In case the reviewing permission is set to **"true"** the document **status bar** will contain the **Review** menu option; the document review will only be available for the document editor if the [mode](#mode) parameter is set to **edit**. The default value is **true**. | boolean | true |
|  | | | |

In case edit is set to **"true"** and review is also set to **"true"**, the user will be able to edit the document, accept/reject the changes made and switch to the review mode him-/herself. In case edit is set to **"true"** and review is set to **"false"**, the user will be able to edit only. In case edit is set to **"false"** and review is set to **"true"**, the document will be available in review mode only.

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"document": {

"permissions": {

"comment": **true**,

"download": **true**,

"edit": **true**,

"print": **true**,

"review": **true**

},

...

},

});

## Editor config

Description

The editorConfig section allows to change the parameters pertaining to the editor interface: opening mode (viewer or editor), interface language, additional buttons, etc.).

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| callbackUrl\* | Specifies URL to the **document storage service** (which [must be implemented](https://api.onlyoffice.com/editors/callback) by the software integrators who use ONLYOFFICE Document Server on their own server). | string | "https://example.com/url-to-callback.ashx" |
| createUrl | Defines the URL of the document where it will be created and available after creation. If not specified, there will be no creation button. | string | "https://example.com/url-to-create-document/" |
|  | | | |
| lang | Defines the editor interface language (if some other languages other than English are present). Is set using the two letter (**de**, **ru**, **it**, etc.) or four letter (**en-US**, **fr-FR**, etc.) language codes. The default value is **"en-US"**. | string | "en-US" |
| mode | Defines the editor opening mode. Can be either **view** to open the document for viewing, or **edit** to open the document in the editing mode allowing to apply changes to the document data. The default value is **"edit"**. | string | "edit" |
| recent | Defines the presence or absence of the documents in the **Open Recent...** menu option where the following document parameters can be set:   * **folder** - the folder where the document is stored (can be empty in case the document is stored in the root folder),  **type**: string,  **example**: "Example Files"; * **title** - the document title that will be displayed in the **Open Recent...** menu option,  **type**: string,  **example**: "exampledocument1.docx"; * **url** - the URL to the document where it is stored,  **type**: string,  **example**: "https://example.com/exampledocument1.docx". | array of object |  |
|  | | | |
| user | Defines the user currently viewing or editing the document:   * **firstname** - the first name of the user. Deprecated since version 4.2, please use name instead,  **type**: string,  **example**: "John"; * **id** - the identification of the user,  **type**: string,  **example**: "78e1e841"; * **lastname** - the last name of the user. Deprecated since version 4.2, please use name instead,  **type**: string,  **example**: "Smith"; * **name** - the full name of the user. Used since version 4.2,  **type**: string,  **example**: "John Smith". | object |  |

\**- required field*

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"editorConfig": {

"callbackUrl": "https://example.com/url-to-callback.ashx",

"createUrl": "https://example.com/url-to-create-document/",

"lang": "en-US",

"mode": "edit",

"recent": [

{

"folder": "Example Files",

"title": "exampledocument1.docx",

"url": "https://example.com/exampledocument1.docx"

},

{

"folder": "Example Files",

"title": "exampledocument2.docx",

"url": "https://example.com/exampledocument2.docx"

},

...

],

"user": {

"id": "78e1e841",

"name": "John Smith"

}

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

### Customization

Description

The customization section allows to customize the editor interface so that it looked like your other products (if there are any) and change the presence or absence of the additional buttons, links, change logos and editor owner details.

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| autosave | Defines if the **Autosave** menu option is enabled or disabled. If set to **false**, only *Strict* co-editing mode can be selected, as *Fast* does not work without autosave. Please note that in case you change this option in menu it will be saved to your browser localStorage. The default value is **true**. | boolean | true |
| chat | Defines if the **Chat** menu button is displayed or hidden; please note that in case you hide the **Chat** button, the corresponding chat functionality will also be disabled. The default value is **true**. | boolean | true |
| commentAuthorOnly | Defines if the user can edit only his comments. The default value is **false**. | boolean | true |
| comments | Defines if the **Comments** menu button is displayed or hidden; please note that in case you hide the **Comments** button, the corresponding commenting functionality will be available for viewing only, the adding and editing of comments will be unavailable. Deprecated since version 4.4, please use [document.permissions.comment](https://api.onlyoffice.com/editors/config/document/permissions" \l "comment) instead. | boolean | true |
| compactToolbar | Defines if the top toolbar type displayed is full (**false**) or compact **true**. The default value is **false**. | boolean | false |
| customer\* | Contains the information for the editor **About** section. The object has the following parameters:   * **address** - postal address of the above company or person,  **type**: string,  **example**: "My City, 123a-45"; * **info** - some information about the above company or person which will be displayed at the **About** page and visible to all editor users,  **type**: string,  **example**: "Some additional information"; * **logo** - the path to the image logo which will be displayed at the **About**page (there are no special recommendations for this file, but it would be better if it were in .png format with transparent background). The image must have the following size: 432x70,  **type**: string,  **example**: "https://example.com/logo-big.png". * **mail** - email address of the above company or person,  **type**: string,  **example**: "john@example.com". * **name** - the name of the company or person who gives access to the editors or the editor authors,  **type**: string,  **example**: "John Smith and Co."; * **www** - home website address of the above company or person,  **type**: string,  **example**: "example.com". | object |  |
| feedback | Defines settings for the **Feedback & Support** menu button. Can be either boolean (simply displays or hides the **Feedback & Support** menu button) or object. In case of object type the following parameters are available:   * **url** - the website address which will be opened when clicking the **Feedback & Support** menu button,  **type**: string,  **example**: "https://example.com"; * **visible** - show or hide the **Feedback & Support** menu button,  **type**: boolean,  **example**: true.   The default value is **false**. | boolean or object | true |
| forcesave | Adds the request for the forced file saving to the [callback handler](#forcesavetype) when saving the document within the **document editing service** (e.g. clicking the **Save** button, etc.). The default value is **false**. | boolean | false |
| goback | Defines settings for the **Go to Documents** menu button and upper right corner link. The object has the following parameters:   * **text** - the text which will be displayed for the **Go to Documents** menu button and upper right corner link (i.e. instead of *Go to Documents*),  **type**: string,  **example**: "Go to Documents"; * **url** - the website address which will be opened when clicking the **Go to Documents** menu button,  **type**: string,  **example**: "https://example.com". | boolean or object | true |
| logo\* | Changes the image file at the top left corner of the Editor header. The recommended image height is 20 pixels. The object has the following parameters:   * **image** - path to the image file used to show in common work mode (i.e. in view and edit modes for all editors). The image must have the following size: 172x40,  **type**: string,  **example**: "https://example.com/logo.png"; * **imageEmbedded** - path to the image file used to show in the embedded mode (see the [config](#type) section to find out how to define the **embedded** document type). The image must have the following size: 248x40,  **type**: string,  **example**: "https://example.com/logo\_em.png"; * **url** - the link which will be used when someone clicks the logo image (can be used to go to your web site, etc.),  **type**: string,  **example**: "https://example.com". | object |  |
| showReviewChanges | Defines if the review changes panel is automatically displayed or hidden when the editor is loaded. The default value is **false**. | boolean | false |
| zoom | Defines the document display zoom value measured in percent. Can take values larger than **0**. For text documents and presentations it is possible to set this parameter to **-1** (fitting the document to page option) or to **-2**(fitting the document page width to the editor page). The default value is **100**. | integer | 100 |
|  | | | |

\**- available for editing only for ONLYOFFICE Document Server Integration Edition*

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"editorConfig": {

"customization": {

"chat": **true**,

"commentAuthorOnly": **false**,

"compactToolbar": **false**,

"customer": {

"address": "My City, 123a-45",

"info": "Some additional information",

"logo": "https://example.com/logo-big.png",

"mail": "john@example.com",

"name": "John Smith and Co.",

"www": "example.com"

},

"feedback": {

"url": "https://example.com",

"visible": **true**

},

"forcesave": **false**,

"goback": {

"text": "Go to Documents",

"url": "https://example.com"

},

"logo": {

"image": "https://example.com/logo.png",

"imageEmbedded": "https://example.com/logo\_em.png",

"url": "https://www.onlyoffice.com"

},

"showReviewChanges": **false**,

"zoom": 100

},

...

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

### Embedded

Description

The embedded section is for the **embedded** document type only (see the [config](https://api.onlyoffice.com/editors/config/" \l "type) section to find out how to define the **embedded** document type). It allows to change the settings which define the behavior of the buttons in the embedded mode.

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| embedUrl | Defines the URL to the document serving as a source file for the document embedded into the web page. | string | "https://example.com/embedded?doc=exampledocument1.docx" |
| fullscreenUrl | Defines the URL to the document which will open in full screen mode. | string | "https://example.com/embedded?doc=exampledocument1.docx#fullscreen" |
| saveUrl | Defines the URL that will allow the document to be saved onto the user personal computer. | string | "https://example.com/download?doc=exampledocument1.docx" |
| shareUrl | Defines the URL that will allow other users to share this document. | string | "https://example.com/view?doc=exampledocument1.docx" |
| toolbarDocked | Defines the place for the embedded viewer toolbar, can be either **top** or **bottom**. | string | "top" |
|  | | | |

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"editorConfig": {

"embedded": {

"embedUrl": "https://example.com/embedded?doc=exampledocument1.docx",

"fullscreenUrl": "https://example.com/embedded?doc=exampledocument1.docx#fullscreen",

"saveUrl": "https://example.com/download?doc=exampledocument1.docx",

"shareUrl": "https://example.com/view?doc=exampledocument1.docx",

"toolbarDocked": "top"

},

...

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

### Plugins

Description

The plugins section allows to connect the special add-ons to your Document Server installation which will help you add additional features to document editors.

Parameters

| Name | Description | Type | Example |
| --- | --- | --- | --- |
| autostart | Defines the array of the identifiers (as entered in [config.json](https://api.onlyoffice.com/plugin/config" \l "guid)) for the plugins, which will automatically start when the editor opens, and the order the plugins will run one-by-one. | array of string |  |
| pluginsData | Defines the array of absolute paths to the plugin configuration files ([config.json](https://api.onlyoffice.com/plugin/config)), which is defined relatively to the url path. | array of string |  |
| url | Defines the absolute path to the directory where the plugins are stored. Deprecated since version 4.3, please use the absolute paths in [pluginsData](https://api.onlyoffice.com/editors/config/editor/plugins" \l "pluginsData)field. | string | "https://example.com/plugins/" |
|  | | | |

Example

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"editorConfig": {

"plugins": {

"autostart": [

"asc.{0616AE85-5DBE-4B6B-A0A9-455C4F1503AD}",

"asc.{FFE1F462-1EA2-4391-990D-4CC84940B754}",

...

],

"pluginsData": [

"https://example.com/plugin1/config.json",

"https://example.com/plugin2/config.json",

...

]

},

...

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed and the plugins are placed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

## Events

Description

The events section allows to change all the functions pertaining to the events.

* **onAppReady** - the function called when the application is loaded into the browser.

Example

**var** onAppReady = **function**() {

**console**.log("ONLYOFFICE Document Editor is ready");

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onAppReady": onAppReady,

...

},

...

});

* **onCollaborativeChanges** - the function called when the document is co-edited by the other user in the *strict* co-editing mode.

Example

**var** onCollaborativeChanges = **function** () {

**console**.log("The document changed by collaborative user");

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onCollaborativeChanges": onCollaborativeChanges,

...

},

...

});

* **onDocumentReady** - the function called when the document is loaded into the document editor.

Example

**var** onDocumentReady = **function**() {

**console**.log("Document is loaded");

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onDocumentReady": onDocumentReady,

...

},

...

});

* **onDocumentStateChange** - the function called when the document is modified. It is called with the parameter: *{"data": true}* when the current user is editing the document and with the parameter: *{"data" : false}* when the current user's changes are sent to the **document editing service**.

Example

**var** onDocumentStateChange = **function** (event) {

**if** (event.data) {

**console**.log("The document changed");

} **else** {

**console**.log("Changes are collected on document editing service");

}

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onDocumentStateChange": onDocumentStateChange,

...

},

...

});

* **onDownloadAs** - the function called with the link to the edited file when the [downloadAs](https://api.onlyoffice.com/editors/methods" \l "downloadAs) method is being called. The link to the document to be downloaded is sent in the *"data"* parameter.

Example

**var** onDownloadAs = **function** (event) {

**console**.log("ONLYOFFICE Document Editor create file: " + event.data);

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onDownloadAs": onDownloadAs,

...

},

...

});

* **onError** - the function called when an error or some other specific event occurs. The error message is sent in the *"data"* parameter.

Example

**var** onError = **function** (event) {

**console**.log("ONLYOFFICE Document Editor reports an error: code " + event.data.errorCode + ", description " + event.data.errorDescription);

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onError": onError,

...

},

...

});

* **onOutdatedVersion** - the function called after the [error](#key) is shown, when the document is opened for editing with the old [document.key](https://api.onlyoffice.com/editors/config/document" \l "key)value, which was used to edit the previous document version and was successfully saved. When this event is called the editor must be reinitialized with a new *document.key*.

Example

**var** onOutdatedVersion = **function** () {

location.reload(true);

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onOutdatedVersion": onOutdatedVersion,

...

},

...

});

* **onReady** - the function called when the application is loaded into the browser. Deprecated since version 5.0
* **onRequestEditRights** - the function called when the user is trying to switch the document from the viewing into the editing mode by clicking the *Edit Document* button. When the function is called, the editor must be initialized again, in editing mode. If the method is not declared the *Edit* button will not be displayed.

**onRequestEditRights** parameter is obligatory when the [editorConfig](https://api.onlyoffice.com/editors/config/editor" \l "mode) *mode* parameter is set to **view** and the *permission* to *edit* the document ([document permissions](#edit)) is set to **true** so that the user could switch to the editing mode.

Example

**var** onRequestEditRights = **function** () {

**console**.log("ONLYOFFICE Document Editor requests editing rights");

**document**.location.reload();

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onRequestEditRights": onRequestEditRights,

...

},

...

});

* **onRequestHistory** - the function called when the user is trying to show the document version history by clicking the *Version History*button. To show the document version history you must call the [refreshHistory](https://api.onlyoffice.com/editors/methods" \l "refreshHistory) method.

Example

**var** onRequestHistory = **function**() {

docEditor.refreshHistory({

"currentVersion": 2,

"history": [

{

"changes": changes, //the *changes* from [the history object](#history) returned after saving the document

"created": "2010-07-06 10:13 AM",

"key": "af86C7e71Ca8",

"serverVersion": serverVersion, //the *serverVersion* from [the history object](#history) returned after saving the document

"user": {

"id": "F89d8069ba2b",

"name": "Kate Cage"

},

"version": 1

},

{

"changes": changes,

"created": "2010-07-07 3:46 PM",

"key": "Khirz6zTPdfd7",

"user": {

"id": "78e1e841",

"name": "John Smith"

},

"version": 2

},

...

]

});

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onRequestHistory": onRequestHistory,

...

},

...

});

* **onRequestHistoryClose** - the function called when the user is trying to go back to the document from viewing the document version history by clicking the *Back to Document* button. When the function is called, the editor must be initialized again, in editing mode. If the method is not declared the *Back to Document* button will not be displayed.

Example

**var** onRequestHistoryClose = **function**() {

**document**.location.reload();

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onRequestHistoryClose": onRequestHistoryClose,

...

},

...

});

* **onRequestHistoryData** - the function called when the user is trying to click the specific document version in the document version history. To show the changes corresponding to the specific document version you must call the [setHistoryData](https://api.onlyoffice.com/editors/methods" \l "setHistoryData) method. The document version number is sent in the *"data"* parameter.

Example

**var** onRequestHistoryData = **function**(event) {

**var** version = event.data;

docEditor.setHistoryData({

"changesUrl": "https://example.com/url-to-changes.zip", //the *changesUrl* from [the JSON object](#changesurl) returned after saving the document

"key": "Khirz6zTPdfd7",

"previous": {

"key": "af86C7e71Ca8",

"url": "https://example.com/url-to-the-previous-version-of-the-document.docx"

},

"url": "https://example.com/url-to-example-document.docx",

"version": version

})

};

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", {

"events": {

"onRequestHistoryData": onRequestHistoryData,

...

},

...

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Methods

After initializing **document editor** you will get the object that can be used to call the methods.

**var** docEditor = **new** DocsAPI.DocEditor("placeholder", config);

* **denyEditingRights** - deny editing. This method can be called when you want to make the document editing unavailable.
* docEditor.denyEditingRights(message);

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| message | Defines the text messages for dialog. | string | optional |

* **destroyEditor** - destroy *docEditor* object. This method can be called when you want to reinit document editor with another configurations. Used since version 4.3.
* docEditor.destroyEditor();
* **downloadAs** - download the edited file. This method can be called only when the existence of the [onDownloadAs](https://api.onlyoffice.com/editors/config/events" \l "onDownloadAs) events. **Document editing service** asynchronously creates a document and triggers the **onDownloadAs** event with a link in parameter.
* docEditor.downloadAs();
* **refreshHistory** - show the document version history. This method must be called after the [onRequestHistory](https://api.onlyoffice.com/editors/config/events" \l "onRequestHistory) events.
* docEditor.refreshHistory({
* "currentVersion": 2,
* "history": [
* {
* "changes": changes, //the *changes* from [the history object](#history) returned after saving the document
* "created": "2010-07-06 10:13 AM",
* "key": "af86C7e71Ca8",
* "serverVersion": serverVersion, //the *serverVersion* from [the history object](#history) returned after saving the document
* "user": {
* "id": "F89d8069ba2b",
* "name": "Kate Cage"
* },
* "version": 1
* },
* {
* "changes": changes,
* "created": "2010-07-07 3:46 PM",
* "key": "Khirz6zTPdfd7",
* "serverVersion": serverVersion,
* "user": {
* "id": "78e1e841",
* "name": "John Smith"
* },
* "version": 2
* },
* ...
* ],
* });

Show the error message explaining why the version history can not be displayed.

docEditor.refreshHistory({

"error": "Exception",

});

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| currentVersion | Defines the current document version number. | integer | required |
| error | Defines the error message text. | string | optional |
| history | Defines the array with the document versions. | array | required |
| history.changes | Defines the *changes* from [the history object](#history) returned after saving the document. | object | optional |
| history.created | Defines the document version creation date. | string | required |
| history.key | Defines the unique document identifier used for document recognition by the service. | string | required |
| history.user | Defines the user who is the author of the document version. | object | optional |
| history.user.id | Defines the identifier of the user who is the author of the document version. | string | optional |
| history.user.name | Defines the name of the user who is the author of the document version. | string | optional |
| history.version | Defines the document version number. | integer | required |

* **setHistoryData** - send the link to the document for viewing the version history. This method must be called after the [onRequestHistoryData](https://api.onlyoffice.com/editors/config/events" \l "onRequestHistoryData)events.
* docEditor.setHistoryData({
* "key": "Khirz6zTPdfd7",
* "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJrZXkiOiJLaGlyejZ6VFBkZmQ3IiwidXJsIjoiaHR0cDovL2V4YW1wbGUuY29tL3VybC10by1leGFtcGxlLWRvY3VtZW50LmRvY3giLCJ2ZXJzaW9uIjoyfQ.N9N9IMwX5e6kdfx4wssAPrGMnzPAZCd4PwDf2D8mJ8s",
* "url": "https://example.com/url-to-example-document.docx",
* "version": 2
* });

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

If after editing and saving the document the link *changesurl* to the file with changes data is returned, download the file by this link and send the file url in *changesUrl* parameter. The url address of the document previous version must be sent in *previous.url* parameter.

docEditor.setHistoryData({

"changesUrl": "https://example.com/url-to-changes.zip",

"key": "Khirz6zTPdfd7",

"previous": {

"key": "af86C7e71Ca8",

"url": "https://example.com/url-to-the-previous-version-of-the-document.docx"

},

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjaGFuZ2VzVXJsIjoiaHR0cDovL2V4YW1wbGUuY29tL3VybC10by1jaGFuZ2VzLnppcCIsImtleSI6IktoaXJ6NnpUUGRmZDciLCJwcmV2aW91cyI6eyJrZXkiOiJhZjg2QzdlNzFDYTgiLCJ1cmwiOiJodHRwOi8vZXhhbXBsZS5jb20vdXJsLXRvLXRoZS1wcmV2aW91cy12ZXJzaW9uLW9mLXRoZS1kb2N1bWVudC5kb2N4In0sInVybCI6Imh0dHA6Ly9leGFtcGxlLmNvbS91cmwtdG8tZXhhbXBsZS1kb2N1bWVudC5kb2N4IiwidmVyc2lvbiI6Mn0.9dgDsaVLFQ6RtoX\_1s2pBVJHGnyMjxDXKC2TpC2nXb4",

"url": "https://example.com/url-to-example-document.docx",

"version": 2

});

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Send the error message explaining why the document version can not be displayed.

docEditor.setHistoryData({

"error": "Exception",

"version": 2

});

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| changesUrl | Defines the url address of the file with the document changes data, which can be downloaded by the *changesurl* link from [the JSON object](#changesurl) returned after saving the document. | string | optional |
| error | Defines the error message text. | string | optional |
| key | Defines the document identifier used to unambiguously identify the document file. | string | required |
| previous | Defines the object of the previous version of the document if *changesUrl* address was returned after saving the document. | object | optional |
| previous.key | Defines the document identifier of the previous version of the document. | string | required |
| previous.url | Defines the url address of the previous version of the document. | string | required |
| token | Defines the encrypted signature added to the parameter in the form of a [token](#setHistoryData). | string | optional |
| url | Defines the url address of the current version of the document. Can be downloaded by the *url* link from [the JSON object](#url) returned after saving the document. | string | required |
| version | Defines the document version number. | integer | required |

* **showMessage** - Display tooltip with the message. This method can be called only after the [onAppReady](https://api.onlyoffice.com/editors/config/events" \l "onAppReady) events.
* docEditor.showMessage(message);

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| message | Defines the message text. | string | required |

# Callback handler

The **document editing service** informs the **document storage service** about status of the document editing using the callbackUrl from [JavaScript API](https://api.onlyoffice.com/editors/basic). The **document editing service** use the POST request with the information in body.

## Parameters and their description:

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| actions | Defines the object received if the new user connected to the document co-editing or disconnected from it. In the first case the type field value is **1**, in the other case - **0**. The userid field value is the identifier of the user who connected to or disconnected from the document co-editing. | array of object | optional |
| changeshistory | Defines the array of objects with the document changes history. The object is present when the status value is equal to **2** or **3** only. Must be sent as a property changes of the object sent as the argument to the [refreshHistory](https://api.onlyoffice.com/editors/methods" \l "refreshHistory) method. Deprecated since version 4.2, please use [history](#history) instead. | array of object | optional |
| changesurl | Defines the link to the file with the document editing data used to track and display the document changes history. The link is present when the status value is equal to **2** or **3**only. The file must be saved and its address must be sent as *changesUrl* parameter using the [setHistoryData](https://api.onlyoffice.com/editors/methods" \l "setHistoryData) method to show the changes corresponding to the specific document version. | string | optional |
| forcesavetype | Defines the type of initiator when the [force saving](#forcesave) request is performed. Can have the following values:   * **0** - the force saving request is performed to the [command service](https://api.onlyoffice.com/editors/command), * **1** - the force saving request is performed each time the saving is done (e.g. the **Save**button is clicked), which is only available when the [forcesave](https://api.onlyoffice.com/editors/config/editor/customization" \l "forcesave) option is set to true. * **2** - the force saving request is performed by timer with the settings from the server config.   The type is present when the status value is equal to **6** or **7** only. | integer | optional |
| history | Defines the object with the document changes history. The object is present when the status value is equal to **2** or **3** only. It contains the object serverVersion and changes, which must be sent as properties serverVersion and changes of the object sent as the argument to the [refreshHistory](https://api.onlyoffice.com/editors/methods" \l "refreshHistory) method. | object | optional |
| key | Defines the edited document identifier. | string | required |
| status | Defines the status of the document. Can have the following values:   * **0** - no document with the key identifier could be found, * **1** - document is being edited, * **2** - document is ready for saving, * **3** - document saving error has occurred, * **4** - document is closed with no changes, * **6** - document is being edited, but the current document state is saved, * **7** - error has occurred while force saving the document. | integer | required |
| url | Defines the link to the edited document to be saved with the document storage service. The link is present when the status value is equal to **2** or **3** only. | string | optional |
| userdata | Defines the custom information sent to the [command service](#userdata) in case it was present in the request. | string | optional |
| users | Defines the list of the identifiers of the users who opened the document for editing; when the document has been changed the **users** will return the identifier of the user who was the last to edit the document (for status **2** and status **6** replies). | array of string | optional |

Status **1** is received every user connection to or disconnection from document co-editing.

Status **2** (**3**) is received [10 seconds](#tenseconds) after the document is closed for editing with the identifier of the user who was the last to send the changes to the document editing service.

Status **4** is received after the document is closed for editing with no changes by the last user.

Status **6** (**7**) is received when the force saving request is performed.

Sample of JSON object sent to the "callbackUrl" address by document editing service when two users are co-editing the document

{

"actions": [{"type": 1, "userid": "78e1e841"}],

"key": "Khirz6zTPdfd7",

"status": 1,

"users": ["6d5a81d0", "78e1e841"]

}

Sample of JSON object sent to the "callbackUrl" address by document editing service when the user changed the document and closed it for editing

{

"actions": [{"type": 0, "userid": "78e1e841"}],

"changesurl": "https://documentserver/url-to-changes.zip",

"history": {

"changes": changes,

"serverVersion": serverVersion

},

"key": "Khirz6zTPdfd7",

"status": 2,

"url": "https://documentserver/url-to-edited-document.docx",

"users": ["6d5a81d0"]

}

Sample of JSON object sent to the "callbackUrl" address by document editing service when the last user closed the document for editing without changes

{

"key": "Khirz6zTPdfd7",

"status": 4

}

Sample of JSON object sent to the "callbackUrl" address by document editing service after the [forcesave](https://api.onlyoffice.com/editors/command) command had been received

{

"changesurl": "https://documentserver/url-to-changes.zip",

"forcesavetype": 0,

"history": {

"changes": changes,

"serverVersion": serverVersion

},

"key": "Khirz6zTPdfd7",

"status": 6,

"url": "https://documentserver/url-to-edited-document.docx",

"users": ["6d5a81d0"],

"userdata": "sample userdata"

}

Response from the document storage service

The **document storage service** must return the following response, otherwise the **document editor** will display an error message:

{

"error": 0

}

The **document manager** and **document storage service** are either included to Community Server or must be implemented by the software integrators who use ONLYOFFICE Document Server on their own server.

.Net (C#) document save example

**public** **class** **WebEditor** : **IHttpHandler**

{

**public** **void** **ProcessRequest**(HttpContext context)

{

**string** body;

**using** (**var** reader = **new** StreamReader(context.Request.InputStream))

body = reader.ReadToEnd();

**var** fileData = **new** JavaScriptSerializer().Deserialize<Dictionary<**string**, **object**>>(body);

**if** ((**int**) fileData["status"] == 2)

{

**var** req = WebRequest.Create((**string**) fileData["url"]);

**using** (**var** stream = req.GetResponse().GetResponseStream())

**using** (**var** fs = File.Open(PATH\_FOR\_SAVE, FileMode.Create))

{

**var** buffer = **new** **byte**[4096];

**int** readed;

**while** ((readed = stream.Read(buffer, 0, 4096)) != 0)

fs.Write(buffer, 0, readed);

}

}

context.Response.Write("{\"error\":0}");

}

}

PATH\_FOR\_SAVE is the absolute path to your computer folder where the file will be saved including the file name.

Java document save example

**public** **class** **IndexServlet** **extends** **HttpServlet** {

@Override

**protected** **void** **doPost**(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

PrintWriter writer = response.getWriter();

Scanner scanner = **new** Scanner(request.getInputStream()).useDelimiter("\\A");

String body = scanner.hasNext() ? scanner.next() : "";

JSONObject jsonObj = (JSONObject) **new** JSONParser().parse(body);

**if**((**long**) jsonObj.**get**("status") == 2)

{

String downloadUri = (String) jsonObj.**get**("url");

URL url = **new** URL(downloadUri);

java.net.HttpURLConnection connection = (java.net.HttpURLConnection) url.openConnection();

InputStream stream = connection.getInputStream();

File savedFile = **new** File(pathForSave);

**try** (FileOutputStream **out** = **new** FileOutputStream(savedFile)) {

**int** read;

final **byte**[] bytes = **new** **byte**[1024];

**while** ((read = stream.read(bytes)) != -1) {

**out**.write(bytes, 0, read);

}

**out**.flush();

}

connection.disconnect();

}

writer.write("{\"error\":0}");

}

}

pathForSave is the absolute path to your computer folder where the file will be saved including the file name.

Node.js document save example

**var** fs = **require**("fs");

app.post("/track", **function** (req, res) {

**var** updateFile = **function** (response, body, path) {

**if** (body.status == 2)

{

**var** file = syncRequest("GET", body.url);

fs.writeFileSync(path, file.getBody());

}

response.write("{\"error\":0}");

response.end();

}

**var** readbody = **function** (request, response, path) {

**var** content = "";

request.on("data", **function** (data) {

content += data;

});

request.on("end", **function** () {

**var** body = **JSON**.parse(content);

updateFile(response, body, path);

});

}

**if** (req.body.hasOwnProperty("status")) {

updateFile(res, req.body, pathForSave);

} **else** {

readbody(req, res, pathForSave)

}

});

pathForSave is the absolute path to your computer folder where the file will be saved including the file name.

PHP document save example

<?php

**if** (($body\_stream = file\_get\_contents("php://input"))===FALSE){

echo "Bad Request";

}

$data = json\_decode($body\_stream, TRUE);

**if** ($data["status"] == 2){

$downloadUri = $data["url"];

**if** (($new\_data = file\_get\_contents($downloadUri))===FALSE){

echo "Bad Response";

} **else** {

file\_put\_contents($path\_for\_save, $new\_data, LOCK\_EX);

}

}

echo "{\"error\":0}";

?>

$path\_for\_save is the absolute path to your computer folder where the file will be saved including the file name.

Ruby document save example

**class** ApplicationController < ActionController::Base

def index

body = request.body.read

file\_data = **JSON**.parse(body)

status = file\_data["status"].to\_i

**if** status == 2

download\_uri = file\_data["url"]

uri = URI.parse(download\_uri)

http = Net::HTTP.new(uri.host, uri.port)

**if** download\_uri.start\_with?("https")

http.use\_ssl = true

http.verify\_mode = OpenSSL::SSL::VERIFY\_NONE

end

req = Net::HTTP::Get.new(uri.request\_uri)

res = http.request(req)

data = res.body

File.open(path\_for\_save, "wb") **do** |file|

file.write(data)

end

end

render :text => "{\"error\":0}"

end

end

path\_for\_save is the absolute path to your computer folder where the file will be saved including the file name.

# Command service

For the interaction with the **document command service** the POST requests are used. The request parameters are entered in JSON format in the request body. The requests are sent to the https://documentserver/coauthoring/CommandService.ashx address where the **documentserver** is the name of the server with the ONLYOFFICE Document Server installed.

In **ONLYOFFICE Document Server** prior to version 4.2 the GET request with the parameters in the QueryString were used.

## Parameters and their description:

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| c | Defined the type of command. Supported values:   * **drop** - disconnect the users with the identifiers present in the users parameter from the **document editing service** (these users will be able to view the document, but will not be allowed to make changes to it); * **forcesave** - force saving the document being edited without closing it (the document editing might be continued after this command, so this will not be the final saved document version); * **info** - receive a document status; * **version** - receive the current version number of Document Server. | string | required |
| key | Defines the document identifier used to unambiguously identify the document file. | string | required |
| userdata | Defines some custom identifier which will help distinguish the specific request in case there were more than one. | string | optional |
| users | Defines the list of the user identifiers (used for the c=drop parameter value). | array of strings | optional |

Sample of JSON object sent to **document command service** used to disconnect the user with the 6d5a81d0 identifier from the document editing service

{

"c": "drop",

"key": "Khirz6zTPdfd7",

"users": [ "6d5a81d0" ]

}

Sample of JSON object sent to **document command service** used for force saving the document with the 6d5a81d0 identifier being edited without closing it

{

"c": "forcesave",

"key": "Khirz6zTPdfd7",

"userdata": "sample userdata"

}

Sample of JSON object sent to **document command service** used to receive the status of the document with the Khirz6zTPdfd7 identifier

{

"c": "info",

"key": "Khirz6zTPdfd7"

}

The request result is returned in JSON form.

Reply format

{

"error": 0,

"key": "Khirz6zTPdfd7"

}

Sample of JSON object sent to **document command service** used to receive the current version number of Document Server

{

"c": "version"

}

The request result of version is returned in JSON form.

Reply format

{

"error": 0,

"version": "4.3.1.4"

}

The **document editing service** informs the **document storage service** about the result caused by command using the [callback handler](https://api.onlyoffice.com/editors/callback).

Possible error codes and their description

| Error code | Description |
| --- | --- |
| 0 | No errors. |
| 1 | Document key is missing or no document with such key could be found. |
| 2 | Callback url not correct. |
| 3 | Internal server error. |
| 4 | No changes were applied to the document before the **forcesave** command was received. |
| 5 | Command not correсt. |
| 6 | Invalid token. |

# Conversion API

For the interaction with the **document conversion service** the POST requests are used. The request parameters are entered in JSON format in the request body. The requests are sent to the https://documentserver/ConvertService.ashx address where the **documentserver** is the name of the server with the ONLYOFFICE Document Server installed.

In **ONLYOFFICE Document Server** prior to version 4.2 the GET request with the parameters in the QueryString were used.

## Parameters and their description:

| Parameter | Description | Type | Presence |
| --- | --- | --- | --- |
| async | Defines the conversion request type: asynchronous or not. Supported values:   * true * false   When the asynchronous request type is used, the reply is formed instantly. In this case to get the result it is necessary to send requests without parameter change until the conversion is finished. | boolean | optional |
| codePage | Defines the file encoding when converting from csv or txt format. Main supported values:   * **932** - Japanese (Shift-JIS), * **950** - Chinese Traditional (Big5), * **1250** - Central European (Windows), * **1251** - Cyrillic (Windows), * **65001** - Unicode (UTF-8).   You can find all the supported values [in this file](https://github.com/ONLYOFFICE/server/blob/master/Common/sources/commondefines.js). | integer | optional |
| delimiter | Defines the delimiter characters for separating values when converting from csv format. Supported values:   * **0** - no delimiter, * **1** - tab, * **2** - semicolon, * **3** - colon, * **4** - comma, * **5** - space. | integer | optional |
| filetype\* | Defines the type of the document file to be converted. | string | required |
| key | Defines the document identifier used to unambiguously identify the document file. | string | required |
| outputtype\* | Defines the resulting converted document type. | string | required |
| thumbnail | Defines the option is used since version 4.2. Settings for the thumbnail when specifying the image formats (bmp, gif, jpg, png) as **outputtype**. The object has the following parameters:   * **aspect** - sets the mode to fit the image to the height and width specifyed. Supported values: 0 - stretch file to fit height and width, 1 - keep the aspect for the image,  **type**: integer,  **example**: 1; * **first** - specifies if the thumbnails should be generated for the first page only or for all the document pages. If false, the zip archive containing thumbnails for all the pages will be created. The default value is true,  **type**: boolean,  **example**: true; * **height** - the thumbnail height in pixels (default: 100),  **type**: integer,  **example**: 100; * **width** - the thumbnail width in pixels (default: 100),  **type**: integer,  **example**: 100. | object | optional |
| title | Defines the converted file name. | string | optional |
| url | Defines the link to the document to be converted. | string | required |

\* - in the tables below you can see possibility of conversion your documents into the most known file formats, where the **Input format** column corresponds to the values of the **filetype** parameter and the **Output format** columns correspond to the values of the **outputtype** parameter.

## Text document file formats

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input format** | **Output format** | | | | | | | | |
| bmp | docx | gif | jpg | odt | pdf | png | rtf | txt |
| doc | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| docm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| docx | ✓ |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| dot | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| dotm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| dotx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| epub | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| fodt | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| html | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| mht | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| odt | ✓ | ✓ | ✓ | ✓ |  | ✓ | ✓ | ✓ | ✓ |
| pdf | ✓ |  | ✓ | ✓ |  |  | ✓ |  |  |
| rtf | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  | ✓ |
| txt | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| xps | ✓ |  | ✓ | ✓ |  | ✓ | ✓ |  |  |

## Spreadsheet file formats

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input format** | **Output format** | | | | | | | |
| bmp | csv | gif | jpg | ods | pdf | png | xlsx |
| csv | ✓ |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| fods | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ods | ✓ | ✓ | ✓ | ✓ |  | ✓ | ✓ | ✓ |
| xls | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| xlsm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| xlsx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| xlt | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| xltm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| xltx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## Presentation file formats

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input format** | **Output format** | | | | | | |
| bmp | gif | jpg | odp | pdf | png | pptx |
| fodp | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| odp | ✓ | ✓ | ✓ |  | ✓ | ✓ | ✓ |
| pot | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| potm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| potx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| pps | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ppsm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ppsx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ppt | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| pptm | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| pptx | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |

Sample of JSON object sent to **document conversion service** used to convert the file from docx format to pdf format

{

"async": false,

"filetype": "docx",

"key": "Khirz6zTPdfd7",

"outputtype": "pdf",

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Sample of JSON object sent to **document conversion service** used to generate png thumbnail of file in docx format

{

"filetype": "docx",

"key": "Khirz6zTPdfd7",

"outputtype": "png",

"thumbnail": {

"aspect": 0,

"first": true,

"height": 150,

"width": 100

},

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

The request result is returned in XML form.

Reply format

<?xml version="1.0" encoding="utf-8"?>

<**FileResult**>

<**FileUrl**>[Link to the converted file at the server]</**FileUrl**>

<**Percent**>[Conversion progress percentage]</**Percent**>

<**EndConvert**>[Conversion is finished - True|False]</**EndConvert**>

</**FileResult**>

When forming the link to the resulting file, the same server name is used which was made the conversion request to.

Reply example

<?xml version="1.0" encoding="utf-8"?>

<**FileResult**>

<**FileUrl**>https://documentserver/ResourceService.ashx?filename=output.doc</**FileUrl**>

<**Percent**>100</**Percent**>

<**EndConvert**>True</**EndConvert**>

</**FileResult**>

Example of the intermediate reply to the asynchronous request (with the parameter async=true)

<?xml version="1.0" encoding="utf-8"?>

<**FileResult**>

<**FileUrl**></**FileUrl**>

<**Percent**>95</**Percent**>

<**EndConvert**>False</**EndConvert**>

</**FileResult**>

Reply format when an error occurred

<?xml version="1.0" encoding="utf-8"?>

<**FileResult**>

<**Error**>Error code</**Error**>

</**FileResult**>

Example of the reply when an error occurred

<?xml version="1.0" encoding="utf-8"?>

<**FileResult**>

<**Error**>-3</**Error**>

</**FileResult**>

Possible error codes and their description

| Error code | Description |
| --- | --- |
| -1 | Unknown error. |
| -2 | Timeout conversion error. |
| -3 | Conversion error. |
| -4 | Error while downloading the document file to be converted. |
| -6 | Error while accessing the conversion result database. |
| -8 | Invalid token. |

# Signature

ONLYOFFICE Document Server [uses](https://api.onlyoffice.com/editors/security) **tokens** generated using the JSON Web Tokens standard.

This feature is used in **Document Server** starting with version 4.2

For the validation setup it is necessary to edit the configuration file which can be found at the following path:

For Linux - /etc/onlyoffice/documentserver/**default.json**.

For Windows - %ProgramFiles%\ONLYOFFICE\DocumentServer\config\**default.json**.

Restart the services for the config changes to take effect:

supervisorctl restart all

Parameters

| Parameter | Description | Type | Example |
| --- | --- | --- | --- |
| services.CoAuthoring.secret.inbox.string | Defines the secret key to generate the token in the [browser](https://api.onlyoffice.com/editors/signature/browser)for the **document editor** opening and calling the methods and the [requests](https://api.onlyoffice.com/editors/signature/request) to the **document command service** and **document conversion service**. | string | secret |
| services.CoAuthoring.secret.outbox.string | Defines the secret key to generate the token in the [requests](https://api.onlyoffice.com/editors/signature/request)by **document editing service** to "callbackUrl" address. | string | secret |
| services.CoAuthoring.token.enable.browser | Specifies the enabling the token validation in the [config](https://api.onlyoffice.com/editors/signature/browser) for the **document editor** opening and calling the methods. | boolean | false |
| services.CoAuthoring.token.enable.request.inbox | Specifies the enabling the token validation in the [requests](https://api.onlyoffice.com/editors/signature/request) to the **document command service** and **document conversion service**. | boolean | false |
| services.CoAuthoring.token.enable.request.outbox | Specifies the enabling the token validation in the [requests](https://api.onlyoffice.com/editors/signature/request)by **document editing service** to "callbackUrl" address. | boolean | false |

Sample default.json configuration

{

"services": {

"CoAuthoring": {

"secret": {

"inbox": {

"string": "secret"

},

"outbox": {

"string": "secret"

},

},

"token": {

"enable": {

"browser": true,

"request": {

"inbox": true,

"outbox": true

}

}

}

}

}

}

## Browser

When performing the client-side browser requests to ONLYOFFICE Document Server a token is added to the parameters to validate the data.

When a file is [opened](https://api.onlyoffice.com/editors/open) for editing in ONLYOFFICE Document Server, the [token](#token) is added to the configuration to validate the parameters.

The payload for the JWT token in the JSON format must have the same structure as the [config](https://api.onlyoffice.com/editors/advanced). The parameter list to be signed is not strictly regulated, but we recommend that you specify the following parameters:

{

"document": {

"fileType": "docx",

"key": "Khirz6zTPdfd7",

"permissions": {

"download": true,

"edit": true,

"print": true,

"review": true

},

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

},

"editorConfig": {

"callbackUrl": "https://example.com/url-to-callback.ashx",

"mode": "edit",

"user": {

"id": "78e1e841",

"name": "Smith"

}

}

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Sample token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkb2N1bWVudCI6eyJmaWxlVHlwZSI6ImRvY3giLCJrZXkiOiJLaGlyejZ6VFBkZmQ3IiwicGVybWlzc2lvbnMiOnsiZG93bmxvYWQiOnRydWUsImVkaXQiOnRydWUsInByaW50Ijp0cnVlLCJyZXZpZXciOnRydWV9LCJ0aXRsZSI6IkV4YW1wbGUgRG9jdW1lbnQgVGl0bGUuZG9jeCIsInVybCI6Imh0dHA6Ly9leGFtcGxlLmNvbS91cmwtdG8tZXhhbXBsZS1kb2N1bWVudC5kb2N4In0sImVkaXRvckNvbmZpZyI6eyJjYWxsYmFja1VybCI6Imh0dHA6Ly9leGFtcGxlLmNvbS91cmwtdG8tY2FsbGJhY2suYXNoeCIsIm1vZGUiOiJlZGl0IiwidXNlciI6eyJpZCI6Ijc4ZTFlODQxIiwibmFtZSI6IlNtaXRoIn19fQ.dQLHmsYqKoAG6vpcLngrIUkstAycpi8dFfP4lHPk0Uc

When calling the [setHistoryData](https://api.onlyoffice.com/editors/methods" \l "setHistoryData) method to view the document [history](https://api.onlyoffice.com/editors/history) version in ONLYOFFICE Document Server, the [token](#token) is added to validate the parameters.

The payload for the JWT token in the JSON format must have the same structure as the method parameter. The parameter list to be signed is not strictly regulated, but we recommend that you specify all the parameter sent:

{

"changesUrl": "https://example.com/url-to-changes.zip",

"key": "Khirz6zTPdfd7",

"previous": {

"key": "af86C7e71Ca8",

"url": "https://example.com/url-to-the-previous-version-of-the-document.docx"

},

"url": "https://example.com/url-to-example-document.docx",

"version": 2

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Sample token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjaGFuZ2VzVXJsIjoiaHR0cDovL2V4YW1wbGUuY29tL3VybC10by1jaGFuZ2VzLnppcCIsImtleSI6IktoaXJ6NnpUUGRmZDciLCJwcmV2aW91cyI6eyJrZXkiOiJhZjg2QzdlNzFDYTgiLCJ1cmwiOiJodHRwOi8vZXhhbXBsZS5jb20vdXJsLXRvLXRoZS1wcmV2aW91cy12ZXJzaW9uLW9mLXRoZS1kb2N1bWVudC5kb2N4In0sInVybCI6Imh0dHA6Ly9leGFtcGxlLmNvbS91cmwtdG8tZXhhbXBsZS1kb2N1bWVudC5kb2N4IiwidmVyc2lvbiI6Mn0.9dgDsaVLFQ6RtoX\_1s2pBVJHGnyMjxDXKC2TpC2nXb4

## Request

When performing the HTTP requests with ONLYOFFICE Document Server an authorization header with the *token* is added to validate the request.

The *payload* for the JSON Web Token contains the following fields:

* **payload** - the object containing the request body parameters in JSON format if not empty.

Validation is performed for inbox requests with the commands from the **document storage service** to the [document command service](https://api.onlyoffice.com/editors/command).

Sample payload of request to receive the status of the edited document

{

"payload": {

"c": "info",

"key": "Khirz6zTPdfd7"

}

}

Sample of request to receive the status of the edited document

POST coauthoring/CommandService.ashx HTTP/1.1

Host: documentserver

Content-Type: application/json

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXlsb2FkIjp7ImMiOiJpbmZvIiwia2V5IjoiS2hpcno2elRQZGZkNyJ9fQ.hGQ8kquQWpg4MQYiSYaIzik0wotP1coAop6QfLruenA

{

"c": "info",

"key": "Khirz6zTPdfd7"

}

Validation is performed for incoming requests with the commands from the **document storage service** to the [document conversion service](https://api.onlyoffice.com/editors/conversionapi).

Sample payload of request to convert the document

{

"payload": {

"filetype": "docx",

"key": "Khirz6zTPdfd7",

"outputtype": "pdf",

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

}

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Sample of request to convert the document

POST ConvertService.ashx HTTP/1.1

Host: documentserver

Content-Type: application/json

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXlsb2FkIjp7ImZpbGV0eXBlIjoiZG9jeCIsImtleSI6IktoaXJ6NnpUUGRmZDciLCJvdXRwdXR0eXBlIjoicGRmIiwidGl0bGUiOiJFeGFtcGxlIERvY3VtZW50IFRpdGxlLmRvY3giLCJ1cmwiOiJodHRwOi8vZXhhbXBsZS5jb20vdXJsLXRvLWV4YW1wbGUtZG9jdW1lbnQuZG9jeCJ9fQ.Ec50Gvafu4niR2I1LPgETn1jkvvs1wKm0YeeI1RbRQs

{

"filetype": "docx",

"key": "Khirz6zTPdfd7",

"outputtype": "pdf",

"title": "Example Document Title.docx",

"url": "https://example.com/url-to-example-document.docx"

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Validation is performed for outgoing requests to "callbackUrl" address by **document editing service**.

Sample payload of request to "callbackUrl" address by document editing service when the last user closed the document for editing without changes

{

"payload": {

"key": "Khirz6zTPdfd7",

"status": 4

}

}

Sample of request to "callbackUrl" address by document editing service when the last user closed the document for editing without changes

POST url-to-callback.ashx HTTP/1.1

Host: example.com

Content-Type: application/json

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJwYXlsb2FkIjp7ImtleSI6IktoaXJ6NnpUUGRmZDciLCJzdGF0dXMiOjR9fQ.MfVoGT-aYuPJfjQAaxwBv7-CqBylDae2AF9K5TpLf-E

{

"key": "Khirz6zTPdfd7",

"status": 4

}

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

Validation is performed for outgoing requests to **document storage service** for file download.

Sample payload of request to document storage service for file download

{}

Sample of request to document storage service for file download

GET url-to-example-**document**.docx HTTP/1.1

Host: example.com

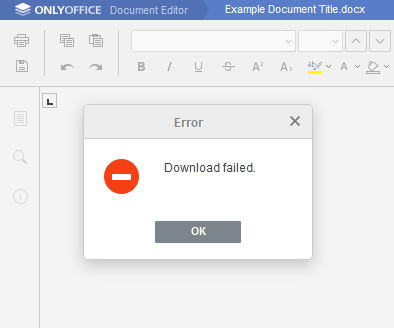
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.LwimMJA3puF3ioGeS-tfczR3370GXBZMIL-bdpu4hOU

Where the **example.com** is the name of the the server where **document manager** and **document storage service** are installed. See the [How it works](https://api.onlyoffice.com/editors/howitworks) section to find out more on Document Server service client-server interactions.

# Troubleshooting

The most common problems with the editors integration and the ways to solve them.

## Download failed



The "Download failed" message is displayed at the editors loading process.

The **Document editing service** cannot upload the file for editing.

Check if the link to the file specified in the [document.url](#url) is correct. The link must be accessible from the **document editing service**.

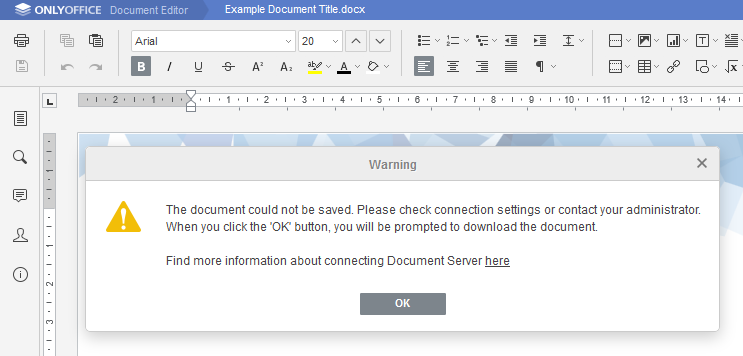
## No changes

When closing the editor after the editing, the file is unchanged in the **document manager**.

The **document editing service** could not send the data to the **document storage service**.

Check if the [editorConfig.callbackUrl](https://api.onlyoffice.com/editors/config/editor" \l "callbackUrl) link is correct. The saving in the **document manager** must be implemented through the [Callback handler](#implement)

## Could not be saved

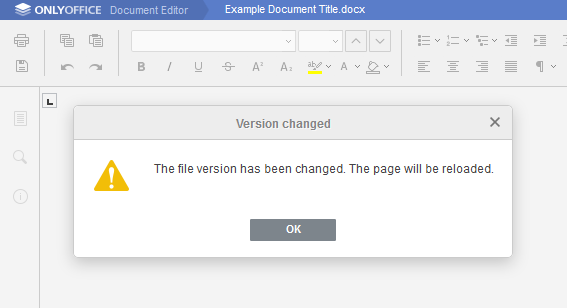


The editor is loaded with the "The document could not be saved" message.

The **document editing service** cannot connect to the **document storage service** at the [editorConfig.callbackUrl](https://api.onlyoffice.com/editors/config/editor" \l "callbackUrl) address.

Check if the [Callback handler](#implement) is working correctly. The **document storage service** must return [{"error": 0}](#error-0) in response.

## The file version has been changed

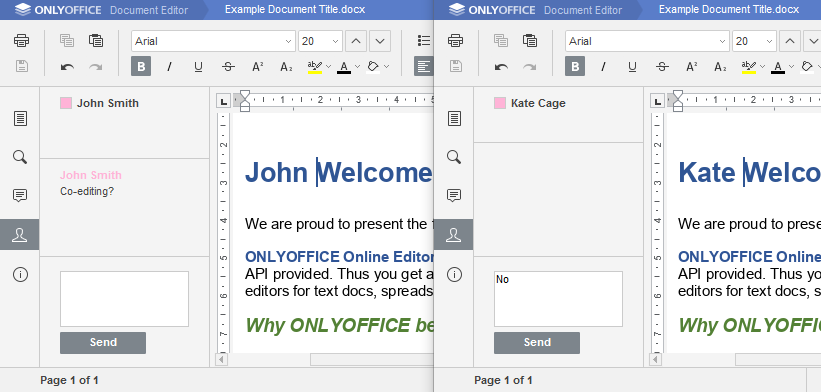


The editor is loaded with the "The file version has been changed. The page will be reloaded" message.

The **document editing service** cannot open the previously edited and saved file for editing.

Do not forget that every time the document is edited and saved, the [document.key](https://api.onlyoffice.com/editors/config/document" \l "key) must be generated anew.

## No collaborative editing

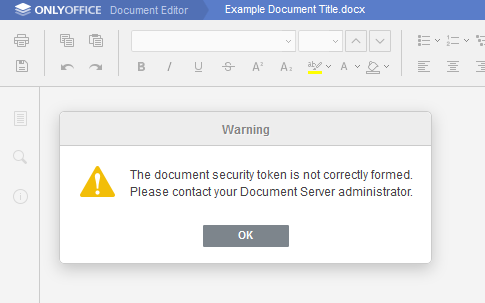


No co-editing is available when the document is opened for editing by different users.

The **document editing service** opens **two different** files for editing.

Check if the [document.key](https://api.onlyoffice.com/editors/config/document" \l "key) values coincide to be able to co-edit the same document. The key value must change after the save, must differ for different documents and coincide when co-editing one and the same document.

## Invalid token



The editor is loaded with the "The document security token is not correctly formed. Please contact your Document Server administrator" message.

The **document editing service** requests an encrypted **signature**.

Check if the [token](#token) is correct. The token must be generated in accordance with the [JWT (JSON Web Tokens) standard](https://api.onlyoffice.com/editors/security) and present in **ONLYOFFICE Document Server** config.