

# OpenText Content Server API

## Table des matières

<b>OPENTEXT CONTENT SERVER API</b>	<b>1</b>
OPENTEXT KEY NAMES :	1
AUTHENTICATION :	1
CREATE A DOCUMENT :	2
CREATE A FOLDER :	3
CREATE A COMPOUND DOCUMENT :	4
CREATE A CATEGORY :	6
MOVE A NODE :	7
UPDATE A NODE :	8
COPY A NODE :	9
DELETE A NODE :	10

---

## OpenText KEY names :

### OTCSTICKET :

OpenText Content Server Ticket

---

## Authentication :

Token that allows someone to perform any action on a specific OpenText node.

- TYPE : POST
- API : /v1/auth
- Parameters :
  - userName
  - password
- Responses : 200 ok, 400 Invalid Parameter value, 401 invalid username/password
  - 500 Argument username is required
  - Argument password is required

## Create a document :

- TYPE : POST
- API : /v1/nodes
- Parameters :
  - type : 144
  - parent\_id
  - name
  - file
- Responses :
  - 200 ok, 400 (General errors like REST API URL could not be found in the mapping registry, item with name already exists, missing parameters etc.
  - 401 Authentication Required

For a document the type is 144.

Parent\_id is the location where we want to create the document (id of the folder).

Name corresponds to the name of the document.

File corresponds to the actual file that we want to upload.

**If we get a 200 code response. We also get the Node ID of the created document.**

## To create a document :

Request	
Type	API
POST	/v1/nodes

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
type	144
parent_id	<i>*folder id*</i>
name	myDoc1
file	<i>*Upload your file here*</i>

JSON Response	
Key	Value
id	<i>*Node id of the document*</i>

---

## Create a folder :

- TYPE : POST
- API : /v1/nodes
- Parameters :
  - type : 0
  - parent\_id
  - name
- Responses :
  - 200 ok, 400 (General errors like REST API URL could not be found in the mapping registry, item with name already exists, missing parameters etc.
  - 401 Authentication Required

For a document the type is 0.

Parent\_id is the location where we want to create the folder (id of the folder).

Name corresponds to the name of the document.

**If we get a 200 code response. We also get the Node ID of the created folder.**

## To create a folder :

Request	
Type	API
POST	/v1/nodes

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
type	0
parent_id	<i>*folder id*</i>
name	myFolder1

JSON Response
---------------

Key	Value
id	<i>*Node id of the folder*</i>

## Create a compound document :

a compound document is *a document that “combines multiple document formats, either by reference, by inclusion, or both.*

Source : [https://en.wikipedia.org/wiki/Compound\\_document](https://en.wikipedia.org/wiki/Compound_document)

For a compound document the type is 136.

Parent\_id is the location where we want to create the compound document (id of the folder).

Name corresponds to the name of the compound document.

File corresponds to the actual compound document we want to upload.

**If we get a 200 code response. We also get the Node ID of the created compound document.**

## To create a compound document :

Request	
Type	API
POST	/v1/nodes

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
type	136
parent_id	<i>*folder id*</i>
name	myCD1

JSON Response	
Key	Value
id	<i>*Node id of the compound document*</i>



## Create a category :

Normally we create a category in the category volume.

For a compound document the type is 131.

Parent\_id is the location where we want to create the category (id of the folder/volume).

Name corresponds to the name of the category.

**If we get a 200 code response. We also get the Node ID of the created category.**

### To create a category :

Request	
Type	API
POST	/v1/nodes

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
type	131
parent_id	<i>*folder or volume id*</i>
name	myCat1

JSON Response	
Key	Value
id	<i>*Node id of the category*</i>

---

## Move a node :

- TYPE : PUT
- API : /v1/nodes/{id}
- Path Parameter : id
- Query Parameters :
  - parent\_id
- Responses :
  - 200 ok
  - 400 (General errors like REST API URL could not be found in the mapping registry, item with name already exists, missing parameters etc.
  - 401 Authentication Required
  - Details of Moved Node

Path parameter is the id of the node we want to move.

Parent\_id is the location where we want to move the node (id of the folder/volume).

**If we get a 200 code response. We also get the details of the moved node.**

### To move a node :

Request	
Type	API
PUT	/v1/nodes{id}

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
parent_id	<i>*folder or volume id*</i>

JSON Response	
Key	Value
id	<i>*Details about the moved node*</i>

---

## Update a node :

### Updating Name and Description

- TYPE : PUT
- API : /v1/nodes/{id}
- Path Parameter : id
- Query Parameters :
  - Name
  - description
- Responses :
  - 200 ok
  - 400 (General errors like REST API URL could not be found in the mapping registry, item with name already exists, missing parameters etc.
  - 401 Authentication Required

Path parameter is the id of the node we want to update.

If we get a 200 code response. We also get the details of the updated node.

### To update a node :

Request	
Type	API
PUT	/v1/nodes{id}

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
name	NewName
description	NewDescription

JSON Response	
Key	Value
id	<i>*Details about the updated node*</i>

---



## Copy a node :

- TYPE : POST
- API : /v1/nodes
- Query Parameters :
  - original\_id
  - parent\_id
- Responses :
  - 200 ok
  - 400 (General errors like REST API URL could not be found in the mapping registry, item with name already exists, missing parameters etc.
  - 401 Authentication Required
  - Details of Copied Node

Original id is the id of the node we want to copy.

Parent\_id is the id of the parent where our node is located (id of the folder/volume).

**If we get a 200 code response. We also get the details of the copied node.**

### To copy a node :

Request	
Type	API
POST	/v1/nodes

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

Body	
KEY	Value
original_id	<i>*id of the node*</i>
parent_id	<i>*id of the parent node*</i>

JSON Response	
Key	Value
id	<i>*Details about the copied node*</i>

---

## Delete a node :

- TYPE : Del
- API : /v1/nodes/{id}
- Path Parameters : id
- Responses :
  - 200 ok
  - 400 Could not get node for {id}  
Could not process object, invalid action 'delete'  
The REST API URL could not be found in the mapping's registry
  - 500 Could not get a node for {id}
  - 401 Authentication Required

Path parameter is the id of the node we want to delete.

**If we get a 200 code response. We do NOT get any response.**

### To delete a node :

Request	
Type	API
DELETE	/v1/nodes/{id}

Header	
KEY	Value
OTCSTICKET (OpenText Content Server Ticket)	<i>*Your ticket*</i>

**Made by :**

**Matthias Brat**

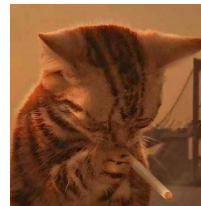
**Socials:**



[github.com/matthiasbrat](https://github.com/matthiasbrat)



[stackoverflow.com/users/17921879/matthias-b](https://stackoverflow.com/users/17921879/matthias-b)



[matthiasbrat.dev](https://matthiasbrat.dev)