

Tutorial: Manage Document Contents in REST Services

This doc shows a series of JSON message samples to create, update, and delete primary contents and renditions in Documentum REST Services(seeing [IIG RESTful Web Services](#)).

1. Assumed that you get a folder resource.

```
GET http://rest-demo/dctm-rest/repositories/acme01/folders/0b0004d2800051b6
```

```
Accept: application/vnd.emc.documentum+json
```

```
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

From the response, you'll find link relations "<http://identifiers.emc.com/linkrel/documents>" and "<http://identifiers.emc.com/linkrel/objects>" which point to the folder child documents and folder child sysobjects feeds, respectively.

```
{
  "name": "folder",
  "type": "dm_folder",
  "definition": "http://rest-demo/dctm-rest/repositories/acme01/types/dm_folder",
  "properties": {"r_object_id": "0b0004d2800051b6"...},
  "links": [
    {"rel": "self"...},
    {"rel": "edit"...},
    {"rel": "http://identifiers.emc.com/linkrel/delete"...},
    {"rel": "http://identifiers.emc.com/linkrel/parent-links"...},
    {"rel": "parent"...},
    {"rel": "http://identifiers.emc.com/linkrel/folders"...},
    {
      "rel": "http://identifiers.emc.com/linkrel/documents",
      "href": "http://rest-dmo/dctm-rest/repositories/acme01/folders/0b0004d2800051b6/documents"
    },
    {
      link relations for child documents and child sysobjects
      "rel": "http://identifiers.emc.com/linkrel/objects",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/folders/0b0004d2800051b6/objects"
    },
    {"rel": "http://identifiers.emc.com/linkrel/child-links"...},
    {"rel": "http://identifiers.emc.com/linkrel/cabinet"...},
    {"rel": "http://identifiers.emc.com/linkrel/relations"...}
  ]
}
```

These two feeds allow you to create new documents/sysobjects under the folder.

2. Create a contentful document

You can post a multi-part HTTP request to the feed to create a contentful document directly with both properties and a primary content. Seeing the sample:

```
POST http://rest-demo/dctm-rest/repositories/acme01/
folders/0b0004d2800051b6/documents

Accept: application/vnd.emc.documentum+json

Content-Type: multipart/form-data;
boundary=vFKDeJ2jOu0ZQqKu9BGfHj9otiMJq44EwXfiV

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

```
--vFKDeJ2jOu0ZQqKu9BGfHj9otiMJq44EwXfiV
```

Content-Disposition: form-data; name="object"

Content-Type: application/vnd.emc.documentum+json;charset=UTF-8

```
{"properties":{"r_object_type": "dm_document", "object_name":  
"readme.txt"}}
```

```
--vFKDeJ2jOu0ZQqKu9BGfHj9otiMJq44EwXfiV
```

Content-Disposition: form-data; name="content"

Content-Type: plain/text; charset=UTF-8

Hello, ReSTer!

```
--vFKDeJ2jOu0ZQqKu9BGfHj9otiMJq44EwXfiV--
```

From the response, a new document resource is returned.

HTTP 201 Created

Location: [http://rest-demo/dctm-rest/repositories/acme01/
objects/090004d28000ba84](http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84)

Content-Type: application/vnd.emc.documentum+json;charset=UTF-8

Content-Length: 4042

```

{
  "name": "object",
  "type": "dm_document",
  "definition": "http://rest-demo/dctm-rest/repositories/acme01/types/dm_document",
  "properties": {"r_object_id": "090004d28000ba84"...},
  "links": [
    {"rel": "self"...},
    {"rel": "edit"...},
    {"rel": "http://identifiers.emc.com/linkrel/delete"...},
    {"rel": "canonical"...},
    {"rel": "http://identifiers.emc.com/linkrel/parent-links"...},
    {"rel": "parent"...},
    {"rel": "http://identifiers.emc.com/linkrel/cabinet"...},
    {
      "rel": "contents",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84/contents"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/primary-content",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84/contents/content"
    },
    {"rel": "http://identifiers.emc.com/linkrel/checkout"...},
    {"rel": "version-history"...},
    {"rel": "http://identifiers.emc.com/linkrel/current-version"...},
    {"rel": "http://identifiers.emc.com/linkrel/relations"...}
  ]
}

```

primary content and contents link relation

From the document resource, you'll find two link relations "contents" and "<http://identifiers.emc.com/linkrel/primary-content>" which point to the contents feed and the primary content of this document, respectively.

3. Create a contentless document

If your client app does not support HTTP multipart POST, you can create a contentful document in 2 steps. Firstly, create a contentless document under the folder.

POST <http://rest-demo/dctm-rest/repositories/acme01/folders/0b0004d2800051b6/documents>

Accept: application/vnd.emc.documentum+json

Content-Type: application/vnd.emc.documentum+json

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

```
{"properties":{"r_object_type": "dm_document", "object_name":  
"readme.txt"}}
```

Then, you could create a new primary content under the document contents feed.

4. Create a primary content

Following the "contents" link relation on the document resource, you can create a new primary content under the contents feed.

```
POST http://rest-demo/dctm-rest/repositories/acme01/  
objects/090004d28000ba84/contents
```

```
Accept: application/vnd.emc.documentum+json
```

```
Content-Type: text/plain
```

```
Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

```
Hello, ReSTer!
```

The response returns you a content metadata resource for the newly created content.

```
{
  "name": "content",
  "properties": {
    "object_name": "readme.txt",
    "r_object_id": "060004d280003fc9",
    "rendition": 0,
    "full_format": "atd",
    "format": "270004d280000215",
    "full_content_size": 14.0,
    "set_time": "2014-02-18T02:02:48.000+0000",
    "i_vstamp": 0,
    "mime_type": "text/plain",
    "dos_extension": "atd",
    "format_name": "atd",
    "parent_id": ["090004d28000ba84"],
    "page": [0],
    "page_modifier": []
  },
  "links": [
    {
      "rel": "self",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84/contents/content"
    },
    {
      "rel": "enclosure",
      "title": "ACS",
      "href": "http://acs-host/ACS/servlet/ACS?command=read&version=2.3&docbaseid=0004d28000ba84&docbasepath=C%3A\\acme01\\objects\\090004d28000ba84\\contents\\content"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/content-media",
      "title": "ACS",
      "href": "http://acs-host/ACS/servlet/ACS?command=read&version=2.3&docbaseid=0004d28000ba84&docbasepath=C%3A\\acme01\\objects\\090004d28000ba84\\contents\\content"
    },
    {
      "rel": "parent",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84"
    }
  ]
}
```

The content metadata resource contains both metadata and media links of the content.

Please note that the format of the content is set as "atd" according to its mime type "text/plain". This is due to the fact of one-many mapping between mime type and formats in Content Server format type. The REST server picks up the default format for the mime type (in name ascending sequence) if the format is not set by the client explicitly. You could customize it in two ways:

- append a URI query parameter "format=<YOUR_PREFERRED_FORMAT_NAME>" at the contents feed URI for POST.
- Update REST server runtime properties to specify the default mime-to-format mapping in **dctm-rest.war/WEB-INF/classes/rest-api-runtime.properties** file.

```
rest.mime.format.mapping.enabled=true

# [repositoryName].[mime_type]=[format], support wildcard repository name
*.text/plain=text

my_repository.application/msword=msw12
```

5. Create a rendition

Besides the primary content, you can also create renditions under the same contents feed.

```
POST http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84/contents?format=crtex&page=0&primary=false

Accept: application/vnd.emc.documentum+json

Content-Type: text/plain

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

Hello, ReSTer, for rendition!
```

For the rendition creation, you need to explicitly specify the format name, page number or page modifier. And it is mandatory to set "primary=false" if the content to create is actually a rendition.

The REST server returns you a rendition.

```

{
  "name": "content",
  "properties": {
    "object_name": "readme.txt",
    "r_object_id": "060004d280003fca",
    "rendition": 2,
    "full_format": "crttext",
    "format": "270004d28000011b",
    "full_content_size": 29.0,
    "set_time": "2014-02-18T02:08:14.000+0000",
    "i_vstamp": 0,
    "mime_type": "text/plain",
    "dos_extension": "txt",
    "format_name": "crttext",
    "parent_id": ["090004d28000ba84"],
    "page": [0],
    "page_modifier": [""]
  },
  "links": [
    {
      "rel": "self",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84,"
    },
    {
      "rel": "enclosure",
      "title": "ACS",
      "href": "http://acs-host/ACS/servlet/ACS?command=read&version=2.3&docbaseid=0004c"
    },
    {
      "rel": "http://identifiers.emc.com/linkrel/content-media",
      "title": "ACS",
      "href": "http://acs-host/ACS/servlet/ACS?command=read&version=2.3&docbaseid=0004c"
    },
    {
      "rel": "parent",
      "href": "http://rest-demo/dctm-rest/repositories/acme01/objects/090004d28000ba84."
    }
  ]
}

```

6. Update a content/rendition

To update an existing content or rendition, POST on the contents feed with the binary of the new content/rendition. In the URI of the contents feed, you need to specify on which format, page number and page modifier to update, and it is required to set the parameter "overwrite=true". For instance,

```
POST http://rest-demo/dctm-rest/repositories/
acme01/objects/090004d28000ba84/contents?
format=crtxt&modifier=&page=0&overwrite=true

Accept: application/vnd.emc.documentum+json

Content-Type: text/plain

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

Hello, ReSTer, for rendition update!
```

7. Delete a content/rendition

To delete an existing content or rendition, DELETE on the "self" link relation of the content resource.

```
DELETE http://rest-demo/dctm-rest/repositories/acme01/
objects/090004d28000ba84/contents?format=crtxt&modifier=&page=0

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

8. Demo of large contentful document import

In below screenshot, I demonstrated the large content (a 4GB file) import using Chrome web browser + Documentum REST Services. In this demo, I used the Chrome extension [DHC](#) to submit the request. It also works on some other REST clients, like [POSTMan](#).

Usecase: Import a 4GB file to a cabinet of the remote repository using HTTP multipart (a document = properties + content).

The screenshot shows a REST client interface with the following components:

- REQUEST:**
 - Method: HTTP
 - URL: `localhost:8080/dctm-rest/repositories/ACME/folders/0c002c778000012f/documents?format=text` (Note: "format=text" is highlighted with a red box and labeled "specify the content format").
 - POST type: POST
 - Send button.
- HEADERS:**
 - Content-Type: multipart/form-data
 - set an authorization button.
- BODY:**
 - Form type: form (selected from a dropdown, labeled "set the body type as form").
 - Parts:
 - object (Text): `{"properties":{"object_name": "a.txt"}}` (labeled "1st part is the JSON representation").
 - binary (File): Select readme#4GB.txt (labeled "2nd part is the local content file").
 - Multipart type: multipart/form-data.
- RESPONSE:**
 - Status: 201 Created
 - HEADERS:
 - Content-Length: 3 KBytes
 - Content-Type: application/vnd.emc.documentum+j...
 - Date: 2015 Feb 4 11:18:17
 - Location: <http://localhost:8080/dctm-rest/rep...>
 - Server: Apache-Coyote/1.1
 - BODY:

```

1 {
2   name: "document",
3   type: "dm_document",
4   definition: "http://localhost:8080/dctm-rest/repositories/ACME/types/dm_document",
5   properties: {

```
 - elapsed time: 8m 18s (labeled "transfer time").

In the DHC client, I added two parts to form the request, where the first is the properties of the new document, and the latter is the content file. Since I am not able to specify the media type for each part, I worked around it a bit. Documentum REST Services takes JSON as the default media type, so I used JSON representation for the properties; besides, I specified the content format in the URI query, to overcome the limitation that I am not able specify the media type for the 2nd part. The request takes around 8 minutes to complete the 4GB file upload. As the result, the response returns the new document representation.

[Learn more about Documentum REST Services >>](#)