

Using curl to debug Documentum REST

Version 3

created by William Zhou on Feb 24, 2017 5:17 AM, last modified by William Zhou on Feb 26, 2017 9:05 AM

Overview

What is **curl**? curl is a command line tool and library for transferring data with URL syntax, supporting HTTP, HTTPS and many other network protocols. curl is free and open source software and have been integrated into many systems. The official site for curl is: <https://curl.haxx.se/>. Linux and Mac OS users get this tool out of the box from terminals. Windows users can get it from the official site's download page.

In this article, we talk about how to debug Documentum REST Services with curl.

Preliminaries

Documentum REST Server

Assumed that you have Documentum REST services deployed and up running at `http://localhost:8080/dctm-rest`. You have a repository named `REPO`, a full privileged test user `dmdadmin` and `password`.

Terminal

Assumed that you have installed curl on your client OS. You can try the very first command `curl --help` to test the tool.

```
01. | curl --help
```

curl has many command parameters. What's good for us is out of these parameters, we only need to use a few of them. This table describes parameters we will use in this tutorial.

Name	Syntax	Description	Example
-u	-u USER[:PSWD]	Server user and password. Basic authentication is assumed for HTTP.	-u dmdadmin:password
--url	--url URL	URL to work with. The prefix '--url' can be skipped.	http://localhost:8080/dctm-rest/product-info
-X	-X METHOD	Specify request command to use.	-X POST
-H	-H NAME[:VALUE]	Pass custom header LINE to server.	-H accept:application/*+xml
-d	-d DATA	Specify HTTP POST data.	-d '{"properties":{"object_name":"a.txt"}}'
-F	-F CONTENT	Specify HTTP multipart POST data.	-F file=@/documents/sample.pdf
-o	-o FILE	Write the response body to FILE instead of stdout	-o out-body.txt
-D	-D FILE	Write the response headers to FILE	-D out-headers.txt

With all these prepared, we can start the debugging.

Test

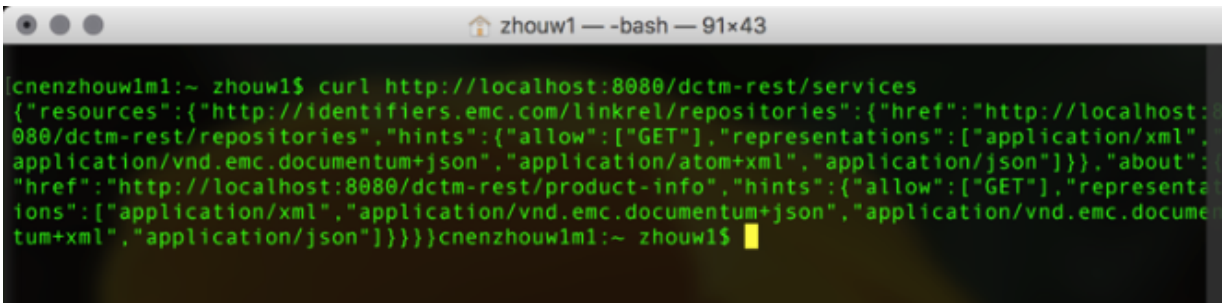
A note we need to highlight at the beginning of the test is, according to hypermedia design [[Make your Documentum REST clients hypermedia controlled](#)], it's discouraged to hardcode resource URI structures in the client code. But since the command-line tool does not have GUI help users to navigate down to link relations, we will use the hard-code URIs in these tests.

1. Get home document

The first test is to get the home document resource of the REST services. Please try with the below command.

```
01. | curl http://localhost:8080/dctm-rest/services
```

The command should return you a JSON response of home document resource.



```

cnenzhouwlm1:~ zhouw1$ curl http://localhost:8080/dctm-rest/services
{"resources":{"http://identifiers.emc.com/linkrel/repositories":{"href":"http://localhost:8080/dctm-rest/repositories","hints":{"allow":["GET"],"representations":["application/xml","application/vnd.emc.documentum+json","application/atom+xml","application/json"]}}},"about":{"href":"http://localhost:8080/dctm-rest/product-info","hints":{"allow":["GET"],"representations":["application/xml","application/vnd.emc.documentum+json","application/atom+xml","application/json"]}}}}
cnenzhouwlm1:~ zhouw1$

```

It's a simple format as some optional parameters are skipped. The complete command for the sample request can be as below.

```
01. | curl -X GET -H accept:* --url http://localhost:8080/dctm-rest/services
```

2. Choose XML or JSON

The default REST service representation is JSON, but XML is supported as well. You can require an XML response in two ways.

- By request header

```
01. | curl -H accept:application/*+xml http://localhost:8080/dctm-rest/services
```

- By URL extension (support GET only)

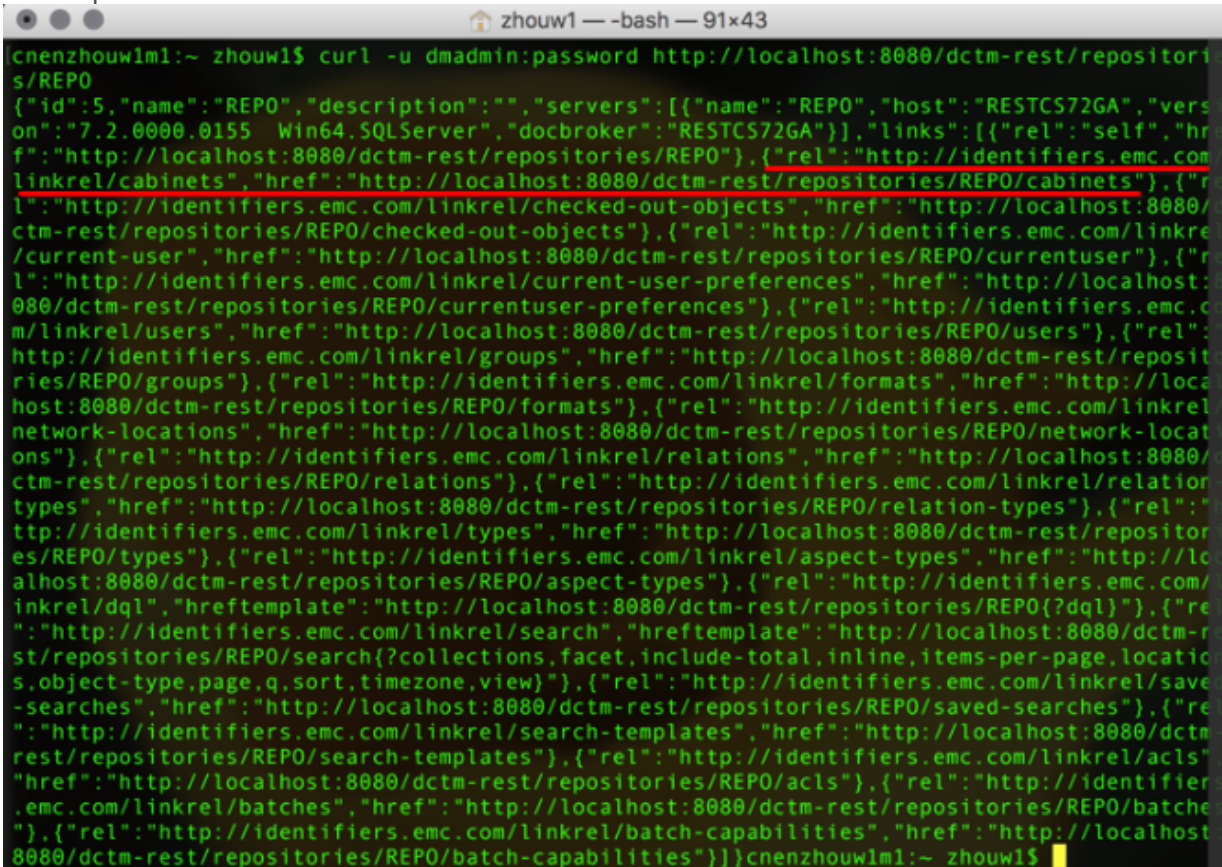
```
01. | curl http://localhost:8080/dctm-rest/services.xml
```

3. Get repository

A repository resource requires the client to provide login credentials. Here we add the username and password to the repository resource request.

```
01. | curl -u dmadmin:password http://localhost:8080/dctm-rest/repositories/REPO
```

The response should be like this.



```

cnenzhouwlm1:~ zhouw1$ curl -u dmadmin:password http://localhost:8080/dctm-rest/repositories/REPO
{"id":5,"name":"REPO","description":"","servers":[{"name":"REPO","host":"RESTC572GA","version":"7.2.0000.0155 Win64.SQLServer","docbroker":"RESTC572GA"}],"links":[{"rel":"self","href":"http://localhost:8080/dctm-rest/repositories/REPO"}, {"rel":"http://identifiers.emc.com/linkrel/cabinets","href":"http://localhost:8080/dctm-rest/repositories/REPO/cabinets"}, {"rel":"http://identifiers.emc.com/linkrel/checked-out-objects","href":"http://localhost:8080/dctm-rest/repositories/REPO/checked-out-objects"}, {"rel":"http://identifiers.emc.com/linkrel/current-user","href":"http://localhost:8080/dctm-rest/repositories/REPO/currentuser"}, {"rel":"http://identifiers.emc.com/linkrel/current-user-preferences","href":"http://localhost:8080/dctm-rest/repositories/REPO/currentuser-preferences"}, {"rel":"http://identifiers.emc.com/linkrel/users","href":"http://localhost:8080/dctm-rest/repositories/REPO/users"}, {"rel":"http://identifiers.emc.com/linkrel/groups","href":"http://localhost:8080/dctm-rest/repositories/REPO/groups"}, {"rel":"http://identifiers.emc.com/linkrel/formats","href":"http://localhost:8080/dctm-rest/repositories/REPO/formats"}, {"rel":"http://identifiers.emc.com/linkrel/network-locations","href":"http://localhost:8080/dctm-rest/repositories/REPO/network-locations"}, {"rel":"http://identifiers.emc.com/linkrel/relations","href":"http://localhost:8080/dctm-rest/repositories/REPO/relations"}, {"rel":"http://identifiers.emc.com/linkrel/relation-types","href":"http://localhost:8080/dctm-rest/repositories/REPO/relation-types"}, {"rel":"http://identifiers.emc.com/linkrel/types","href":"http://localhost:8080/dctm-rest/repositories/REPO/types"}, {"rel":"http://identifiers.emc.com/linkrel/aspect-types","href":"http://localhost:8080/dctm-rest/repositories/REPO/aspect-types"}, {"rel":"http://identifiers.emc.com/linkrel/dql","hreftemplate":"http://localhost:8080/dctm-rest/repositories/REPO/{?dql}"}, {"rel":"http://identifiers.emc.com/linkrel/search","hreftemplate":"http://localhost:8080/dctm-rest/repositories/REPO/search{?collections,facet,include-total,inline,items-per-page,location,object-type,page,q,sort,timezone,view}"}, {"rel":"http://identifiers.emc.com/linkrel/saved-searches","href":"http://localhost:8080/dctm-rest/repositories/REPO/saved-searches"}, {"rel":"http://identifiers.emc.com/linkrel/search-templates","href":"http://localhost:8080/dctm-rest/repositories/REPO/search-templates"}, {"rel":"http://identifiers.emc.com/linkrel/acls","href":"http://localhost:8080/dctm-rest/repositories/REPO/acls"}, {"rel":"http://identifiers.emc.com/linkrel/batches","href":"http://localhost:8080/dctm-rest/repositories/REPO/batches"}, {"rel":"http://identifiers.emc.com/linkrel/batch-capabilities","href":"http://localhost:8080/dctm-rest/repositories/REPO/batch-capabilities"}]}
cnenzhouwlm1:~ zhouw1$

```

4. Create a new cabinet

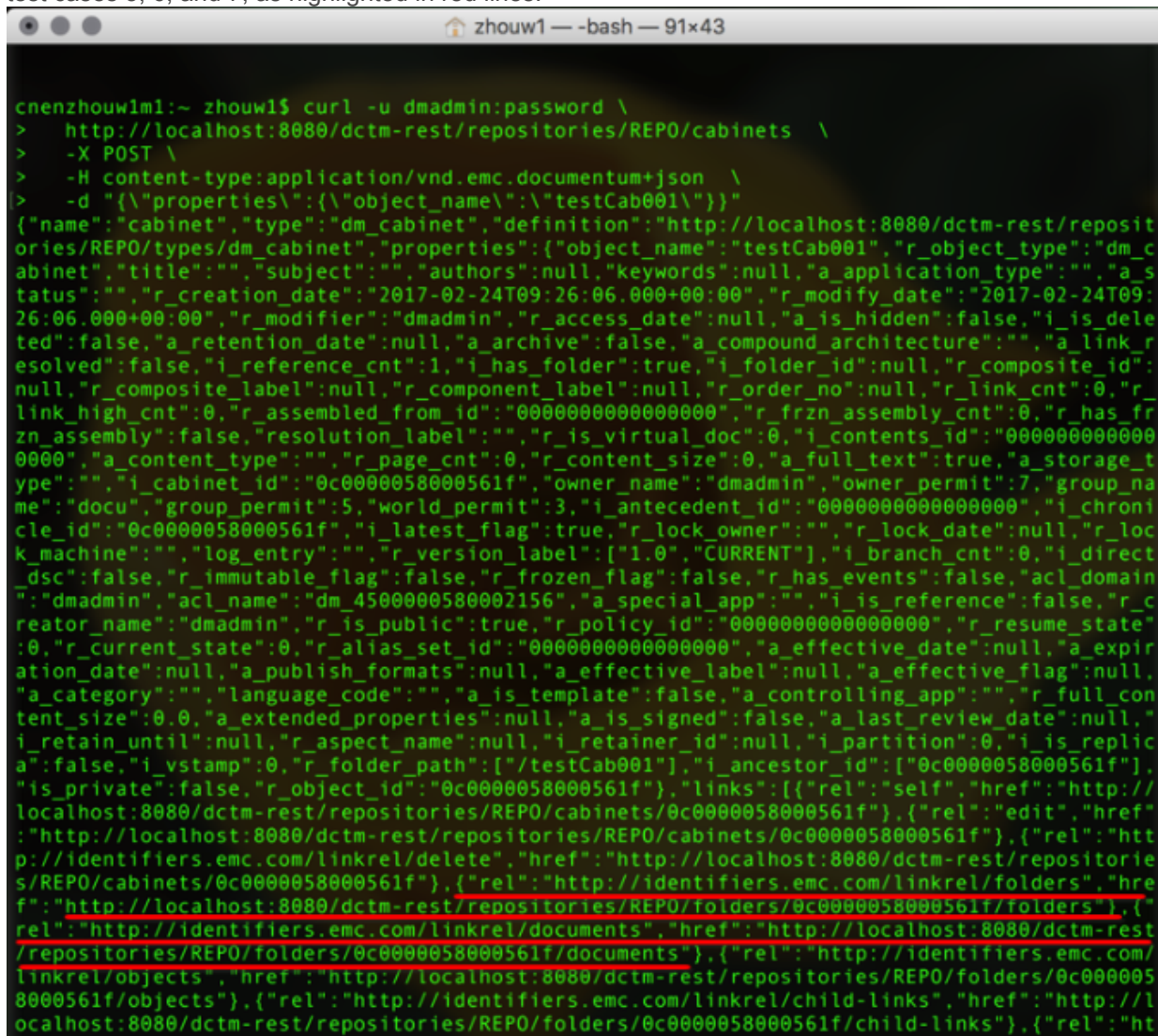
Next, let's create a new cabinet in the repository **REPO**. The red lines in the last screenshot shows the cabinets resource URI we are going to use in this test. I assume that the user **dmadmin** already gets the privilege to create cabinets. The new cabinet name would be **testCab001**.

```
01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/cabinets \
03. -X POST \
04. -H content-type:application/vnd.emc.documentum+json \
05. -d "{\"properties\":{\"object_name\":\"testCab001\"}}"
```

Please note that the request method must be **POST** for a creation. The content type header is required for a POST method, and here is the JSON type **application/vnd.emc.documentum+json**. The request body is double quoted, and as a result, any double quotes in the JSON message should be escaped by slash (****).

You can write the the long command in one line, but you are also allowed to write it in multiple lines for pretty looking. If you do that, don't forget to use slash (****) as the line separator.

And here is the result snippet. From the response, we will use the folders resource URI and documents resource URI to run test cases 5, 6, and 7, as highlighted in red lines.



```
cnenzhouw1m1:~ zhouw1$ curl -u dmadmin:password \
> http://localhost:8080/dctm-rest/repositories/REPO/cabinets \
> -X POST \
> -H content-type:application/vnd.emc.documentum+json \
> -d "{\"properties\":{\"object_name\":\"testCab001\"}}\"
{"name":"cabinet","type":"dm_cabinet","definition":"http://localhost:8080/dctm-rest/reposit
ories/REPO/types/dm_cabinet","properties":{"object_name":"testCab001","r_object_type":"dm_c
abinet","title":"","subject":"","authors":null,"keywords":null,"a_application_type":"","a_s
tatus":"","r_creation_date":"2017-02-24T09:26:06.000+00:00","r_modify_date":"2017-02-24T09:
26:06.000+00:00","r_modifier":"dmadmin","r_access_date":null,"a_is_hidden":false,"i_is_dele
ted":false,"a_retention_date":null,"a_archive":false,"a_compound_architecture":"","a_link_r
esolved":false,"i_reference_cnt":1,"i_has_folder":true,"i_folder_id":null,"r_composite_id":
null,"r_composite_label":null,"r_component_label":null,"r_order_no":null,"r_link_cnt":0,"r_
link_high_cnt":0,"r_assembled_from_id":"0000000000000000","r_frzn_assembly_cnt":0,"r_has_fr
zn_assembly":false,"resolution_label":"","r_is_virtual_doc":0,"i_contents_id":"000000000000
0000","a_content_type":"","r_page_cnt":0,"r_content_size":0,"a_full_text":true,"a_storage_t
ype":"","i_cabinet_id":"0c0000058000561f","owner_name":"dmadmin","owner_permit":7,"group_na
me":"docu","group_permit":5,"world_permit":3,"i_antecedent_id":"0000000000000000","i_chroni
cle_id":"0c0000058000561f","i_latest_flag":true,"r_lock_owner":"","r_lock_date":null,"r_loc
k_machine":"","log_entry":"","r_version_label":["1.0","CURRENT"],"i_branch_cnt":0,"i_direct
dsc":false,"r_immutable_flag":false,"r_frozen_flag":false,"r_has_events":false,"acl_domain
":"dmadmin","acl_name":"dm_4500000580002156","a_special_app":"","i_is_reference":false,"r_c
reator_name":"dmadmin","r_is_public":true,"r_policy_id":"0000000000000000","r_resume_stat
e":0,"r_current_state":0,"r_alias_set_id":"0000000000000000","a_effective_date":null,"a_expir
ation_date":null,"a_publish_formats":null,"a_effective_label":null,"a_effective_flag":null,
"a_category":"","language_code":"","a_is_template":false,"a_controlling_app":"","r_full_con
tent_size":0.0,"a_extended_properties":null,"a_is_signed":false,"a_last_review_date":null,"
i_retain_until":null,"r_aspect_name":null,"i_retainer_id":null,"i_partition":0,"i_is_replic
a":false,"i_vstamp":0,"r_folder_path":["/testCab001"],"i_ancestor_id":["0c0000058000561f"],
"is_private":false,"r_object_id":"0c0000058000561f"},"links":[{"rel":"self","href":"http://
localhost:8080/dctm-rest/repositories/REPO/cabinets/0c0000058000561f"}, {"rel":"edit","href"
:"http://localhost:8080/dctm-rest/repositories/REPO/cabinets/0c0000058000561f"}, {"rel":"htt
p://identifiers.emc.com/linkrel/delete","href":"http://localhost:8080/dctm-rest/repositorie
s/REPO/cabinets/0c0000058000561f"}, {"rel":"http://identifiers.emc.com/linkrel/folders","hre
f":"http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/folders"}, {"
rel":"http://identifiers.emc.com/linkrel/documents","href":"http://localhost:8080/dctm-rest
/repositories/REPO/folders/0c0000058000561f/documents"}, {"rel":"http://identifiers.emc.com/
linkrel/objects","href":"http://localhost:8080/dctm-rest/repositories/REPO/folders/0c000005
8000561f/objects"}, {"rel":"http://identifiers.emc.com/linkrel/child-links","href":"http://l
ocalhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/child-links"}, {"rel":"ht
```

5. Create a new folder

With the new cabinet resource, we will create a folder in it. The new folder name is **testFolder001**. Here is the curl command.

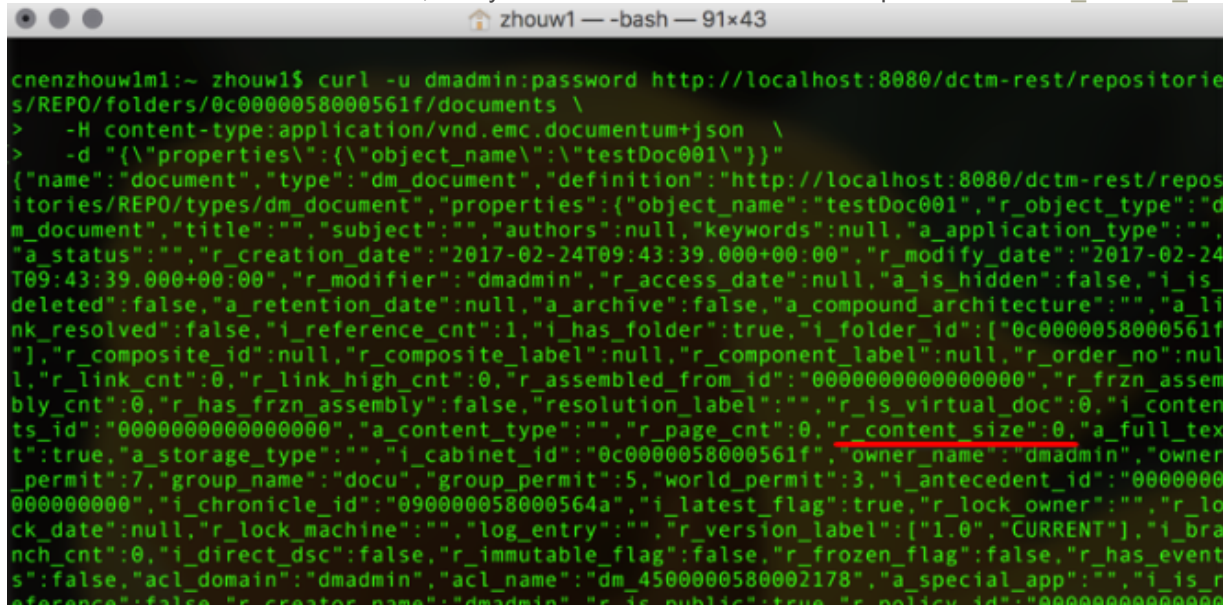
```
01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/folders \
03. -H content-type:application/vnd.emc.documentum+json \
04. -d "{\"properties\":{\"object_name\":\"testFolder001\"}}"
```


6. Create a new contentless document

Then we create a new document in the same cabinet. The resource URI is different, and the document name is `testDoc001`. Here is the curl command.

```
01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/documents \
03. -H content-type:application/vnd.emc.documentum+json \
04. -d '{"properties":{"object_name":"testDoc001"}}'
```

This document has no contents in it, and you can observe that from the response attribute `r_content_size`.



```
cnrenzhouw1m1:~ zhouw1$ curl -u dmadmin:password http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/documents \
> -H content-type:application/vnd.emc.documentum+json \
> -d '{"properties":{"object_name":"testDoc001"}}'
{"name":"document","type":"dm_document","definition":"http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document","properties":{"object_name":"testDoc001","r_object_type":"dm_document","title":"","subject":"","authors":null,"keywords":null,"a_application_type":"","a_status":"","r_creation_date":"2017-02-24T09:43:39.000+00:00","r_modify_date":"2017-02-24T09:43:39.000+00:00","r_modifier":"dmadmin","r_access_date":null,"a_is_hidden":false,"i_is_deleted":false,"a_retention_date":null,"a_archive":false,"a_compound_architecture":"","a_link_resolved":false,"i_reference_cnt":1,"i_has_folder":true,"i_folder_id":["0c0000058000561f"],"r_composite_id":null,"r_composite_label":null,"r_component_label":null,"r_order_no":null,"r_link_cnt":0,"r_link_high_cnt":0,"r_assembled_from_id":"0000000000000000","r_frzn_assembly_cnt":0,"r_has_frzn_assembly":false,"resolution_label":"","r_is_virtual_doc":0,"i_contents_id":"0000000000000000","a_content_type":"","r_page_cnt":0,"r_content_size":0,"a_full_text":true,"a_storage_type":"","i_cabinet_id":"0c0000058000561f","owner_name":"dmadmin","owner_permit":7,"group_name":"docu","group_permit":5,"world_permit":3,"i_antecedent_id":"0000000000000000","i_chronicle_id":"090000058000564a","i_latest_flag":true,"r_lock_owner":"","r_lock_date":null,"r_lock_machine":"","log_entry":"","r_version_label":["1.0","CURRENT"],"i_branch_cnt":0,"i_direct_desc":false,"r_immutable_flag":false,"r_frozen_flag":false,"r_has_event_s":false,"acl_domain":"dmadmin","acl_name":"dm_4500000580002178","a_special_app":"","i_is_reference":false,"r_creator_name":"dmadmin","r_is_public":true,"r_policy_id":"0000000000000000"}
```

In real world apps, we usually create a contentful document, right? So we will do that in test case 7.

7. Create a new contentful document

The contentful document creation in a folder is performed by issuing a multipart request, and curl supports multipart POST. The resource URI is same to step 6. In the request body,

- the first part is a JSON document describing the document properties.
- the second part is a local content file you want to import.

So please create a JSON file in your client OS, and name it as `doc-request.json`. It's content is as this.

```
01. {
02.   "properties": {"object_name":"testDoc002"}
03. }
```

Next, assume that you have a PDF file in your client OS, named as `sample.pdf`. Now let's run the curl command.

```
01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/documents \
03. -F file=@/Users/zhouw1/doc-request.json \
04. -F file=@/Users/zhouw1/sample.pdf
```

You don't need to specify the content type header in the multipart request since curl will do it for you. The response will tell you that this document's content size is not zero.

```

zhouw1 ~ zhouw1$ curl -u dmadmin:password http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f/documents \
> -F file=@/Users/zhouw1/doc-request.json \
> -F file=@/Users/zhouw1/sample.pdf
{"name":"document","type":"dm_document","definition":"http://localhost:8080/dctm-rest/repositories/REPO/types/dm_document","properties":{"object_name":"testDoc002","r_object_type":"dm_document","title":"","subject":"","authors":null,"keywords":null,"a_application_type":"","a_status":"","r_creation_date":"2017-02-24T09:54:52.000+00:00","r_modify_date":"2017-02-24T09:54:52.000+00:00","r_modifier":"dmadmin","r_access_date":null,"a_is_hidden":false,"i_is_deleted":false,"a_retention_date":null,"a_archive":false,"a_compound_architecture":"","a_link_resolved":false,"i_reference_cnt":1,"i_has_folder":true,"i_folder_id":["0c0000058000561f"],"r_composite_id":null,"r_composite_label":null,"r_component_label":null,"r_order_no":null,"r_link_cnt":0,"r_link_high_cnt":0,"r_assembled_from_id":"0000000000000000","r_frzn_assembly_cnt":0,"r_has_frzn_assembly":false,"resolution_label":"","r_is_virtual_doc":0,"i_content_id":"06000005800036cf","a_content_type":"emcmf","r_page_cnt":1,"r_content_size":1031734,"a_full_text":true,"a_storage_type":"filestore_01","i_cabinet_id":"0c0000058000561f","owner_name":"dmadmin","owner_permit":7,"group_name":"docu","group_permit":5,"world_permit":3,"i_antecedent_id":"0000000000000000","i_chronicle_id":"090000058000564b","i_latest_flag":true,"r_lock_owner":"","r_lock_date":null,"r_lock_machine":"","log_entry":"","r_version_label":{"1.0","CURRENT"},"i_branch_cnt":0,"i_direct_dsc":false,"r_immutable_flag":false,"r_frozen_flag":false,"r_has_events":false,"acl_domain":"dmadmin","acl_name":"dm_4500000580002184","a_special_app":"","i_is_reference":false,"r_creator_name":"dmadmin","r_is_public":true,"r_policy_id":"0000000000000000","r_resume_state":0,"r_current_state":0,"r_alias_set_id":"0000000000000000","a_effective_date":null,"a_expiration_date":null,"a_publish_formats":null,"a_effective_label":null,"a_effective_flag":null,"a_category":"","language_code":"","a_is_template":false,"a_controlling_app":"","r_full_content_size":1031734.0,"a_extended_properties":null,"a_is_signed":false,"a_last_review_date":null,"i_retain_until":null,"r_aspect_name":null

```

NOTE: If you are using REST 7.3 GA version, please upgrade it to 7.3 Patch 01 or later versions for this test case. curl does NOT specify the right content type in the 1st part, and REST 7.3 GA version doesn't allow clients to skip the 1st part content type.

8. Download a document content

The article [Tutorial: Manage Document Contents in REST Services](#) introduces how to get the content download URI from a document resource. We will construct the download URI directly here. For the content download in curl, it's better to specify an output file location; otherwise, the content binary will be printed in stdout. In this command, we export the content to output.pdf.

```

01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/objects/090000058000564b/content-media \
03. -o output.pdf

```

curl will report the progress and once it is done, you can check your local directory for the output file.

```

zhouw1 ~ zhouw1$ curl -u dmadmin:password \
> http://localhost:8080/dctm-rest/repositories/REPO/objects/090000058000564b/content-media \
> -o output.pdf
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 1007k  100 1007k    0     0  2385k      0  --:--:-- --:--:-- --:--:-- 2381k
cnzhenzhouw1m1:~ zhouw1$

```

9. Remove a document

Removing a document is by performing a DELETE on the document resource. It's straightforward enough.

```

01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/objects/090000058000564b \
03. -X DELETE

```

10. Dump and troubleshooting

curl provides several parameters to dump the response into local files. With `-D` and `--trace`, you can dump the complete response information.

```

01. curl -u dmadmin:password \
02. http://localhost:8080/dctm-rest/repositories/REPO/folders/0c0000058000561f \

```

```
03. -D out-headers.txt \  
04. --trace out-bin.txt
```

Summary

In this article, we quickly went through the REST API debugging with the tool curl. curl is very light and easy to use. Wishfully, it will help you in starting with Documentum REST Services.

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