

Tutorial: Commenting On Objects in Documentum REST Services

Commenting on documents or pages is a useful feature in an interactive application. Since 7.3, Documentum REST Services supports to comment on Sysobjects with the integration of Documentum Collaboration Service (DCS). This tutorial is going to coach you how to use this new and powerful resource.

This doc shows a series of JSON message samples in Documentum REST Services to

- Get access to **Comments** collection from a **Sysobject**
- Make a new **Comment**
- **Reply** to other's *Comment*

Preliminary

The commenting function is depending on DCS. Therefore, the Collaboration Services Archive (DAR) files must be installed in the repository prior to following this tutorial. The installation of DCS requires an additional license. Please refer to [Documentum page](#) for further information of installing the DCS.

Section 1: Getting access to 'Comments' resource

Assume that you get a document resource.

GET http://localhost:8080/dctm-rest/repositories/REPO/objects/080000058000191f Accept: application/vnd.emc.documentum+json

From the document resource response, you will find a link relation . This link gives you access to all comments made to this specific document object.

You could then make a GET request on the comments link to retrieve the collection of comments for this document. In this tutorial, we show some samples on commenting on this document.

Section 2: Getting 'Comments' feed resource

Get on the href URL of the link relation from the document response.

```
GET http://localhost:8080/dctm-rest/repositories/REPO/objects/080000058000191f/comments Accept: application/vnd.emc.documentum+xml
```

In this demonstration, the document is newly created, so you will find that the comments collection is empty. To create a first comment on this document, you can make a POST request to this feed resource.

Section 3: Commenting on specific Sysobject

Now please post a comment with a request body. The minimal request payload is to have a property **content-value** with the comment in plain text. You can also post a comment in HTML format, but please make sure to escape HTML markups in the REST XML or JSON representation.

```
POST http://localhost:8080/dctm-rest/repositories/REPO/objects/080000058000191f/comments Content-Type: application/vnd.emc.documentum+xml
```

```
{ "content-value": "This is a comment." }
```

If your post is successful, a new comment will be created for this document and returned to the client with its representation.

```
HTTP/1.1 201 Created Content-Type: application/vnd.emc.documentum+json
```

```
{ "json-root": "comment", "object-id": "08000005800048e6", "comment-id": "18662", "owner-name": "dadmin", "creation-date": "2017-12-18T08:00:00Z" }
```

Please note:

- Each comment is also a sysobject so it has its own 'object-id'.
- The 'comment-id' is only used in comments resource for ordering.
- Since this comment is the root comment, it has no parent id.
- If you get to its 'parent' link, it will redirect you back to the comments feed.

The appearance of link relations and are depending on property values of '**can-reply**' and '**can-delete**', respectively. As you are the owner of both the document and the comment, you get both link relations from the response. In this release, comment resources are not available for modification.

Now, you have already created a topmost comment, so you can move on to add a reply on this comment by make a POST request to the link relation .

Section 4: Replying to a Comment

Now please post a reply with a request body. The request payload is the same as a new comment.

POST http://localhost:8080/dctm-rest/repositories/REPO/objects/080000058000191f/comments/18662/replies Content-Type: applica

```
{ "content-value": "This is a reply." }
```

If your post is successful, a new reply comment will be added under this comment, and returned to the client with its representation.

Compare with the former response, you can find out that this reply has a '**parent-id**' which is pointed to the former comment by its '**comment-id**'. Also, the '**parent**' link is now pointed to the parent comment, not the comments feed resource.

For your own exploration, you can continue to add more replies on this reply as a recursive operation.

Section 5: Getting 'Comments' feed resource again after commenting

Now let's get back to the document's comments feed again (find the link relation <http://identifiers.emc.com/linkrel/comments> from the document). You will find the topmost comment in the feed (YES -- there is only **ONE** comment *"This is a comment"* under the topmost feed).

If you include a query parameter '**inline=true**', you can see the details in '**content**' section of this comment like this:

Content-Type: application/vnd.emc.documentum+json Status: 200 OK

```
{ "id": "http://localhost:8080/dctm-rest/repositories/REPO/objects/080000058000191f/comments", "title": "Comments on object 080000058000191f" }
```

If you follow the inline comment's link relation to make another GET request, you will get another feed with the single reply comment *"This is a reply"* in the feed.

What it infers is that a sysobject's comments in Documentum REST Services are organized hierarchically. When you make a GET request to a document's link relation, or a comment's link relation, you will only get one level of child comments under that parent.

Now it's your turn to explore the commenting service in Documentum REST. Please leave your comments or feedbacks below the page.

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