

Tutorial : Managing Users and Groups in Documentum REST Services

This article shows a series of JSON message samples to create, retrieve, update, and delete dm_user and dm_group objects in Documentum REST Services.

Manage Users

Get Users Collection

Assumed that you get a repository resource with below request.

```
GET

Accept: application/vnd.emc.documentum+json

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

From the repository response, you'll find a link relation "" which points to Users collection.

```
{  "properties":{      .....  }  "links":[      .....  {      "rel":"http://identifiers.emc.com/linkrel/users",      "href":
```

With this link relation, we can make a GET method to get users in this repository.

```
GET

Accept: application/vnd.emc.documentum+json

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

From the response, Users collection is returned. It's a regular feed collection resource that supports inline, pagination, filter, view and other common parameters.

```
{  "id": "http://localhost:8080/dctm-rest/repositories/REPO/users"  "title": "Users",  "author": [  {      "name": "EMC Documentum"
```

For any single user resource, please make a GET method to the href of the link relation *edit* on specific feed entries.

```
GET

Accept: application/vnd.emc.documentum+json

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==
```

The response returns the resource of the user.

```
{ "name": "user", "type": "dm_user", "definition": "http://demo-server/dctm-rest/repositories/REPO/types/dm_user", "properties":
```

Create a User

In Users collection resource, we can create a user by making a POST method, with the new user request payload. Properties **"user_name"** and **"user_login_name"** are mandatory.

```
POST

Accept: application/vnd.emc.documentum+json

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

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

{"properties":{"user_name": "demo-user", "user_login_name": "demo-
login", "user_source": "inline password", "user_password": "password"}}
```

The response returns the link of the newly created user in header **"Location"**. The response also returns the new user representation in the response body. Please note user password is NOT returned.

```
HTTP 201 Created
Location:
```

```
{ "name": "user", "type": "dm_user", "definition": "http://demo-server/dctm-rest/repositories/REPO/types/dm_user", "proper
```

Update a User

The user resource representation uses the link relation *edit* to indicate whether the accessor has the privilege to update it. On the user resource we are going to update, if the accessor sees the link relation "*edit*", it indicates that the accessor has the privilege to update this user.

```
{  ....  "links":[  ....  {  "rel":"edit",  "href":"http://demo-server/dctm-rest/repositories/REPO/users/dem
```

Please update this user by POST user properties we are going to change. For example, update the user state to "locked".

POST

Accept: application/vnd.emc.documentum+json

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

Authorization: Basic ZG1hZG1pbjpwYXNzd29yZA==

```
{"properties":{"user_state": 1}}
```

The response returns 200 if the update is successful, and the updated user representation is returned in response body.

```
{  ....  "properties":{"  ....  "user_state":1,  ....  }  "links":[  ....  ]}
```

Delete a User

The user representation uses the link relation to indicate whether the accessor has the privilege to delete it. On the user resource we have obtained, if the accessor sees the link relation , it indicates that the accessor has the privilege to delete this user.

```
{  ....  "links":[  ....  {  "rel":"http://identifiers.emc.com/linkrel/delete",  "href":"http://demo-server/dctm
```

Now you can delete this user by making a DELETE method to this link.

DELETE

Accept: application/vnd.emc.documentum+json

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

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

The response returns status code 204 without a response body if the deletion is successful.

Manage Groups

The CRUD of groups are similar to users CRUD. On a repository resource, we can find the link relation "" which points to the Groups collection resource.

Besides the CRUD, we can also manage the members of a group. For more details about the group members management, please check [Tutorial : Managing Group Members in Documentum REST Services](#)

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