

# Tutorial: Managing Lightweight Objects in Documentum REST Services

---

In Documentum, lightweight object types are a special type used to minimize the storage footprint for multiple objects that share the same system information. Documentum REST Services in release 7.3 adds full support for the lightweight object operations. This tutorial will cover these contents for the lightweight operations.

- Reading lightweight types
- CRUD lightweight objects
- Materializing lightweight objects
- Dematerializing lightweight objects
- Reparenting lightweight objects

## 1. Get Lightweight Object Types

Documentum REST Services in 7.3 has enhanced the Type(s) resources to support lightweight object types. On the existing Type(s) resources, the existed lightweight and shareable object type(s) can be obtained. The '**type\_category**' property is used in the '**filter**' query parameter to get lightweight or shareable object type

- **type\_category=4** for the lightweight type
- **type\_category=2** for the shareable type

The following are sample URIs:

- GET /dctm-rest/repositories/{repositoryName}/types?filter=type\_category=4
- GET /dctm-rest/repositories/{repositoryName}/types?filter=type\_category=2

Compared to release 7.2, the new Type resource representation has improvements in these areas:

- Two new properties are added on the Type resource:

- **shared-parent**: the URI of the parent shareable type, available only when the type is a lightweight object type
- **category**: the category of a type, can be one of aspect, shareable, lightweight, data table, and standard

```
{
  "name" : "my_lightweight_document",
  "label" : "Document",
  "category" : "lightweight",
  "parent" : "http://localhost:8080/dctm-rest/repositories/REPO/types/my_shareable_document",
  "shared-parent" : "http://localhost:8080/dctm-rest/repositories/REPO/types/my_shareable_document",
  "properties" : [
    {
      "name" : "r_object_type",
      "repeating" : false,
      "type" : "string",
      "length" : 32,

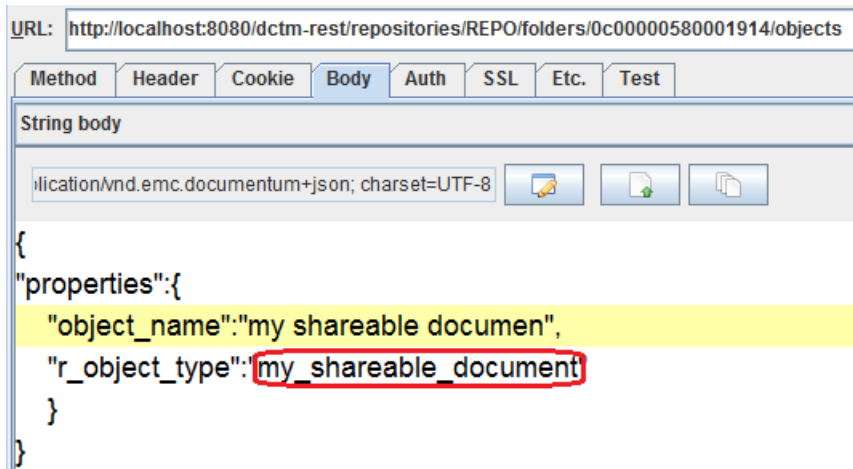
```

- Two new link relations are added on the Type resource:
  - <http://identifiers.emc.com/linkrel/parent-shareable-type>: available for the lightweight type, to get the parent shareable type
  - <http://identifiers.emc.com/linkrel/lightweight-types>: available for the shareable type, to get its child lightweight type

```
{ "name" : "my_lightweight_document", "label" : "Document", "category" : "lightweight", "parent" : "http://localhost:8080/dctm-rest"
```

## 2. Create a Shareable Type Object

Since multiple lightweight objects can share a shareable object, before creating the lightweight object, a shareable object must be created first. The shareable object can be created as a normal sysobject, except the object type to be a shareable object type.



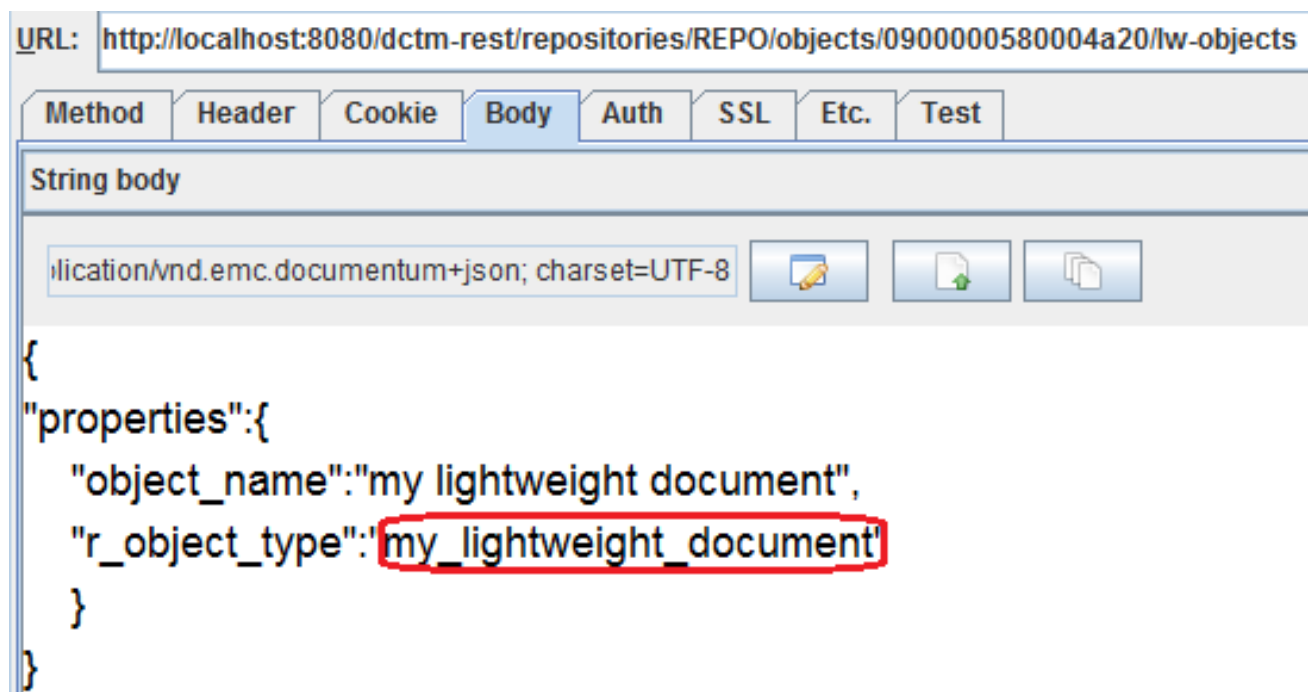
Once the object is created, there will one new link relation on the shareable object, <http://identifiers.emc.com/linkrel/lightweight-objects>. Following this link, the client can GET all lightweight objects that share this shareable object, or POST to create a new lightweight object for this shareable object.

```
{
  "rel" : "http://identifiers.emc.com/linkrel/permission-set",
  "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a20/permission-set"
},
{
  "rel" : "http://identifiers.emc.com/linkrel/permissions",
  "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a20/permissions"
},
{
  "rel" : "http://identifiers.emc.com/linkrel/lightweight-objects",
  "href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a20/lw-objects"
},
-
```

### 3. Create a Lightweight Object

As this is a newly created shared object, it does not have any lightweight objects under it. Here we are going to create a new lightweight object under the newly created shareable

object. So following the link relation "", the client can make a POST request. The request body is same as a normal sysobject, except the object type to be the lightweight object type.



URL: <http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a20/lw-objects>

Method Header Cookie **Body** Auth SSL Etc. Test

String body

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

```
{
  "properties":{
    "object_name":"my lightweight document",
    "r_object_type":"my_lightweight_document"
  }
}
```

Other operations like update, delete, etc., are the same as normal sysobjects.

#### 4. Materialize and Dematerialize The Lightweight Object

When a lightweight object has a **private copy of a parent**, the object is **materialized**. A dematerialized lightweight object can be materialized. And vice versa, a materialized lightweight object can be dematerialized. The dematerialization means that the lightweight object abandons its private copy of the parent, and re-shares the original shareable object. In Documentum REST services, users can use the link relations to distinguish whether a lightweight object is materialized or dematerialized. The default lightweight object is dematerialized.

- On a dematerialized lightweight object, there is the other link relation <http://identifiers.emc.com/linkrel/materialize> instead. The client can make a PUT method to materialize it.

```
"rel" : "http://identifiers.emc.com/linkrel/permissions",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/permissions"
},
{
"rel" : "http://identifiers.emc.com/linkrel/shared-parent",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/parent"
},
{
"rel" : "http://identifiers.emc.com/linkrel/materialize",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/materialization"
},
```

- On a materialized lightweight object, there is one link relation <http://identifiers.emc.com/linkrel/dematerialize> on its resource representation. To dematerialize it, the client can send a DELETE method to the href of this link relation.

```
-
"rel" : "http://identifiers.emc.com/linkrel/permissions",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/permissions"
},
{
"rel" : "http://identifiers.emc.com/linkrel/shared-parent",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/parent"
},
{
"rel" : "http://identifiers.emc.com/linkrel/dematerialize",
"href" : "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/materialization"
},
```

## 5. Reparent the Lightweight Object

More than one lightweight objects can share one shareable object. A lightweight object can be **reparented** to another shareable object, to share the parent's properties. The 'reparent' is only valid for the lightweight object. There is one link relation <http://identifiers.emc.com/linkrel/shared-parent> to reparent lightweight objects. To reparent for a lightweight object, the client can make a POST method with the object href which points to a new sharable object URI.

```
"i_sharing_type" : "0000000000000000",  
"i_orig_parent" : "0900000580004a20",  
"allow_propagating_changes" : false,  
"i_sharing_parent" : "0900000580004a21",  
"r_object_id" : "0900000580004a21"
```



POST to <http://identifiers.emc.com/linkrel/shared-parent>

URL: <http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a21/parent>

Method Header Cookie Body Auth SSL Etc. Test

String body

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

```
{  
  "href": "http://localhost:8080/dctm-rest/repositories/REPO/objects/0900000580004a22"  
}
```



```
"i_sharing_type" : "0300000580000211",  
"i_orig_parent" : "0900000580004a22",  
"allow_propagating_changes" : false,  
"i_sharing_parent" : "0900000580004a22",  
"r_object_id" : "0900000580004a21"
```

## 6. Demo

Please click the following picture to see the demo animation.

## Tutorial: Managing Lightweight Objects in Documentum REST Services

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

- URL:** `http://localhost:8080/dctm-rest/repositories/REPO/types?filter=type_category=2`
- Method:** ☒ GET, ☐ POST, ☐ PUT, ☐ PATCH, ☐ DELETE, ☐ HEAD, ☐ OPTIONS, ☐ TRACE
- HTTP Response:** Status:
- Body:**
- Test Result:**

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