

# Advanced Searches in Documentum REST Services (3): Saved Searches

---

## Overview

This tutorial is the third post of the Search Services Series in Documentum REST 7.3. With the knowledge of *AQL* and facets, users can execute real time search with sophisticated criteria. In a web application, a search criteria may be constructed from a complicated form. To make the same search criteria, users have to follow the same steps to prepare and submit the form carefully and repeatedly. In Documentum REST 7.3, we introduce a new resource **Saved Searches** which help users to persist the search criteria to the server side, just like creating persistent objects in Content Server.

Part 1. [Advanced Searches in Documentum REST Services \(1\): AQL](#)

Part 2. [Advanced Searches in Documentum REST Services \(2\): Facets](#)

Part 3. Advanced Searches in Documentum REST Services (3): Saved Searches

Part 4. [Advanced Searches in Documentum REST Services \(4\): Search Templates](#)

## Preliminary

Please refer to the *Preliminary* section in [Advanced Searches in Documentum REST Services \(1\): AQL](#) to prepare the environment.

## Discover Saved Searches Resource

Since from release 7.3, the repository resource adds an additional link relation which points to the Saved Searches resource.

```
<!--Representation of Repository Resource--> <repository xmlns="http://identifiers.emc.com/vocab/documentum" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://identifiers.emc.com/vocab/documentum http://identifiers.emc.com/vocab/documentum.xsd">
```

## Save a Search

Here is a sample of a saved search representation. There are three meta attributes defining search name (*object\_name*), description (*title*) and access permission (*r\_is\_public*). The most important part is *query-document*, which contains an AQL statement. If the AQL contains special characters, we need to escape the characters or use a CDATA to save the AQL in the saved-search message.

```
<!--Saved Search Representation--> <saved-search definition="http://localhost:8080/dctm-rest/repositories/REPO/types/dmc_search_template.xml" object_name="Sample Search" title="Sample Search" r_is_public="true">
```

The POST request below is to create a new saved search. The request body contains the full representation of a saved search in XML.

```
POST /dctm-rest/repositories/ubuntudb/saved-searches HTTP/1.1 Host: localhost:8080 Content-Type: application/vnd.emc.documentum+xml
```

```
<!--Request Body--> <saved-search definition="http://localhost:8080/dctm-rest/repositories/REPO/types/dmc_search_template.xml" object_name="Sample Search" title="Sample Search" r_is_public="true">
```

If the search is saved successfully, the response will return HTTP status code 201 and the new saved search resource in the response body. The saved search resource supports regular get, update and deletion operations.

```
<!--Response Body--> <saved-search definition="http://localhost:8080/dctm-rest/repositories/ubuntudb/types/dm_smart_list" xmlns="http://identifiers.emc.com/vocab/documentum" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://identifiers.emc.com/vocab/documentum http://identifiers.emc.com/vocab/documentum.xsd">
```

Probably you have noticed that there are three new link relations in the saved search representation. Let's dig the first two and leave the third one to [Advanced Searches in Documentum REST Services \(4\): Search Templates](#).

•  
•

- 

## Execute Saved Search

A saved search has no values if it cannot be executed. As you can imagine, the link relation allows users to execute the saved search. The snippet below is the GET request to execute saved search.

```
GET /dctm-rest/repositories/ubuntudb/saved-searches/0800a79e80002ec2/execution HTTP/1.1 Host: localhost:8080 Accept: applic
```

You can still add additional URL parameters like *items-per-page*, *include-total*, *page* and *inline* on the execution URL to customize the search result. The search result for the saved search execution is exactly the same as a POST search service request with the same AQL.

```
<!--Response Body (entries are omitted)--> ... <dm:facet xmlns:dm="http://identifiers.emc.com/vocab/documentum"> <dm:face
```

## Navigate Facets

Navigating facets in [Advanced Searches in Documentum REST Services \(2\): Facets](#) tells that, to get facet results, users need to make a POST method with the original AQL to the facet URL. However, for a saved search resource, getting facet results is much simpler. Since the server saves the AQL and knows what it is, users don't need to submit the AQL again. A GET method will return the facet result to users.

By following the href with GET request, users can navigate facet results and there are only two entries returned as expectation.

```
GET /dctm-rest/repositories/ubuntudb/saved-searches/0800a79e80002ec2/execution.xml?facet-id-constraints=facet_keywords%3D
```

```
<!--Response Body--> <feed xmlns="http://www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

## Saved Search Results

Users not only can execute the realtime search on on a saved search, but also can save its search results on the server. When creating a saved search, the attribute ***has\_results in*** the saved search request can be set to ***true***, indicating to save the result. With this feature, users can get a snapshot of the search result from the REST services without querying to the xPlore index server, if stale results are affordable.

## Enable or Refresh Saved Search Results

For a saved search resource that hasn't enabled saved search results option yet, users can turn on the saving by making a PUT method to the href of link relation .

```
PUT /dctm-rest/repositories/ubuntu/saved-searches/0800a79e80002ec2/results HTTP/1.1 Host: localhost:8080 Cache-Control: no-cache
```

The response is same as normal search execution, which is not shown here. Please note that **facet results won't be saved**.

Now we can get the saved search again and check the attributes *has\_results* and *results\_count*.

```
<!--Saved Search Representation with Results Enabled--> <saved-search definition="http://localhost:8080/dctm-rest/repositories/ubuntu/saved-searches/0800a79e80002ec2/results">
```

Once the option is enabled, users can still keep the saved results updated by PUT to the href time to time.

## Disable Saved Search Results

It's easy to disable this option. Below is a DELETE request to the href of link relation <http://identifiers.emc.com/linkrel/saved-search-results>, which disables saved search results.

DELETE /dctm-rest/repositories/ubuntudb/saved-searches/0800a79e80002ec2/results HTTP/1.1 Host: localhost:8080 Cache-Contr

After disabling the option, the attributes *has\_results* and *results\_count* become *false* and *-1*.

<!--Saved Search Representation with Results Disabled--> <saved-search definition="http://localhost:8080/dctm-rest/repositories/ub

## Saved Search As Template

This topic is related to search template. Refer [Advanced Searches in Documentum REST Services \(4\): Search Templates](#) for the details.

## Conclusion

After completing the tutorial, you get to know:

- a saved search contains meta attributes and query document
- saving a new search is actually to create a persistent object with attributes and AQL
- executing a saved search is similar with real time executing search with AQL, except that HTTP GET is used instead of POST
- saved results is not fresh data, but it can still be updated
- saved results can be disabled or enabled after saved search creation

**Next:** *Part 4. [Advanced Searches in Documentum REST Services \(4\): Search Templates](#)*

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