

Advanced Searches in Documentum REST Services (1): AQL

Version 36

created by [wwwwwch](#) on Oct 31, 2016 11:48 PM, last modified by [wwwwwch](#) on Dec 7, 2016 9:35 PM

rest-version 7.3+

Overview

Since Documentum REST Services 7.3, users can execute full-text search with Abstract Query Language (AQL), which can provide more sophisticated search criteria, comparing with [Search Service with Simple Search Language in 7.2](#). Besides, search persistence and search template are supported as well.

This tutorial series introduces a user scenario and shows how to fulfill the cases step by step. When finishing the tutorial, you should know

- what search related services can do
- how to consume the search services

As the first post of the series, you will learn what is AQL and how to execute search with AQL.

Part 1. Advanced Searches in Documentum REST Services (1): AQL

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

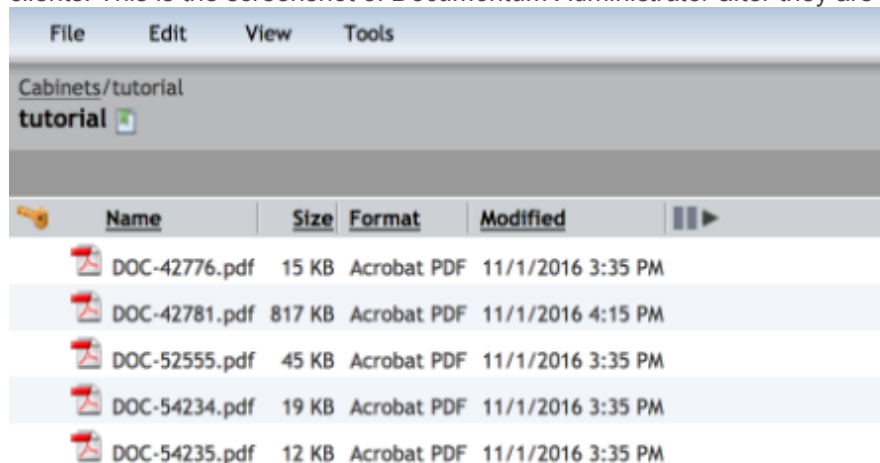
Part 3. [Advanced Searches in Documentum REST Services \(3\): Saved Searches](#)

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

Preliminary

To consume full-text search services in Documentum, the xPlore server including the index agent, should be deployed and configured for Content Server. Details can be found in the product documentation of xPlore.

To run the full-text search services, we need to prepare some documents in the repository. There are several sample documents attached under this article. You can find them and import them to the repository folder by any Documentum clients. This is the screenshot of *Documentum Administrator* after they are imported.



Please also set the attribute **keywords** on these 5 documents. Here is a table for the details.

Document	Keywords
DOC-42776	extensibility
DOC-42781	search
DOC-52555	client
DOC-54234	extensibility
DOC-54235	extensibility

This is how it looks after editing *keywords* attribute.

Properties: Info

Info Permissions History


DOC-42776.pdf
 Type: dm_document
 Format: pdf

Name:

Title:

Subject:

Keywords: [Edit](#) extensibility

Authors: [Edit](#)

Full Content Size: 15312

Discover Search Service

The advanced search service's link relation and URI pattern is the same search service for Documentum REST 7.2. In the representation of repository resource, there is a link relation <http://identifiers.emc.com/linkrel/search> pointing to the search service.

```

01.  <!--Representation of Repository Resource-->
02.  <repository xmlns="http://identifiers.emc.com/vocab/documentum" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03.    <id>42910</id>
04.    <name>ubuntudb</name>
05.    ...
06.    <links>
07.      ...
08.      <link hreftemplate="http://localhost:8080/dctm-rest/repositories/ubuntudb/search{?collections,
09.        rel="http://identifiers.emc.com/linkrel/search"/>
10.      <link href="http://localhost:8080/dctm-rest/repositories/ubuntudb/saved-searches"
11.        rel="http://identifiers.emc.com/linkrel/saved-searches"/>
12.      ...
13.    </links>
14.  </repository>

```

Abstract Query Language

In [Tutorial: Full-text Search with Documentum REST Services 7.2](#), the search criteria is quite simple - a keyword *tutorial*. What if users want to specify the document size, owner and name? AQL comes for such requirements in release 7.3.

The full specification of AQL won't be covered in this article; instead, we use samples to demonstrate how to construct AQL. Here is the first sample of an AQL request in XML.

```

01.  <search xmlns="http://identifiers.emc.com/vocab/documentum"
02.    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
03.    <types>
04.      <type>dm_document</type>
05.    </types>
06.    <expression-set operator="AND">
07.      <expressions>
08.        <fulltext>tutorial</fulltext>
09.        <property name="object_name" operator="BEGINS_WITH">DOC</property>
10.        <property-list name="owner_name" operator="IN">
11.          <values>
12.            <value>dmdadmin</value>
13.          </values>
14.        </property-list>
15.        <property-range name="r_full_content_size" operator="BETWEEN">
16.          <from>800000</from>
17.          <to>900000</to>
18.        </property-range>

```

```

19.     </expressions>
20.   </expression-set>
21. </search>

```

The above AQL sample describes these search criteria:

1. the object type must be **dm_document** or its sub types
2. A *full-text* expression defines the keyword **tutorial**
3. A *property* expression defines the object name starting with **DOC**
4. A *property-list* expression defines the owner name is in the name list of **dmadmin**
5. A *property-range* expression defines the document size is in the range **800000 - 900000** bytes
6. Conditions 2~5 must be met together since the joiner is **AND**

Execute Advance Search

The snippet below makes a POST request to search service and AQL statements above is the request body, which is not shown here.

```

01. POST /dctm-rest/repositories/ubuntutdb/search?include-total=true HTTP/1.1
02. Host: localhost:8080
03. Content-Type: application/vnd.emc.documentum+xml
04. Accept: application/xml
05.
06. <search>...</search> //the above AQL sample

```

It will return the search results as a feed. For this request, there is only 1 result.

```

01. <feed xmlns="http://www.w3.org/2005/Atom" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
02.   ...
03.   <dm:total xmlns:dm="http://identifiers.emc.com/vocab/documentum">1</dm:total>
04.   ...
05.   <entry>
06.     <id>0900a79e80002deb</id>
07.     <title>DOC-42781.pdf</title>
08.     <author>
09.       <name>dmadmin</name>
10.     </author>
11.     <summary>Generated by Jive on 2016-11-01-07:00 1 Tutorial: Full-text...-type,page,q,sort,timeze
12.       Tutorial: Full-text Search with... What to Search in the Tutorial Assuming there is a bri
13.       some data and content are populated for this tutorial. 1
14.     </summary>
15.     <updated>2016-11-01T08:15:44.000+00:00</updated>
16.     <link href="http://localhost:8080/dctm-rest/repositories/ubuntutdb/objects/0900a79e80002deb" re
17.     <content src="http://localhost:8080/dctm-rest/repositories/ubuntutdb/objects/0900a79e80002deb"
18.     <relevance:score xmlns:relevance="http://a9.com/-/opensearch/extensions/relevance/1.0/">1.0
19.     </relevance:score>
20.     <dm:terms xmlns:dm="http://identifiers.emc.com/vocab/documentum">
21.       <dm:term>Tutorial</dm:term>
22.       <dm:term>dmadmin</dm:term>
23.       <dm:term>doc</dm:term>
24.       <dm:term>tutorial</dm:term>
25.     </dm:terms>
26.   </entry>
27. </feed>

```

If you decrease the **from** value in the document size range expression in AQL, more than one search results will be returned.

As a sample, this AQL request is quite straightforward. In fact, AQL is powerful and flexible, with which users can define complex search criteria to meet complicated requirements

Conclusion

Starting from Documentum REST 7.2, full-text search is included, with basic functionality via Simple Search Language. The release 7.3 makes a big step forward with AQL. With this post, you get to know:

- AQL can do much more than Simple Search Language

- AQL has many kinds of expressions for users to work out complex search criteria
- Use HTTP method POST to execute the AQL request

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

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