

Design - Admin Search

There is currently no search functionality included in the core repository. Tooling in support of [External Search](#) is provided to address most search functionality.

However, there has been interest in providing at least some level of built-in search functionality to address basic discovery scenarios. To guide planning and development, please provide concrete use cases your repository applications have for search that are not well-served by external search options.

List Repository URIs Based on Last Modification Time

Kevin Ford - 6 Feb 2017

There should be a way to request a list of repository resources based on last modification time. The use case is as follows:

One overcast Chicago day, connectivity was lost between the server hosting Fedora and its embedded ActiveMQ broker and the server hosting Karaf/Camel subscribed to the broker's 'fedora' topic. The weather, though a strangely inserted detail in the preceding sentence, had no bearing on the loss in connectivity. It just happened.

When connectivity was restored, Karaf/Camel of course re-subscribed to the 'fedora' topic, but only new messages were received. Any messages published to the broker from the time connectivity was lost to the time connectivity was restored were not retrieved and processed by Karaf/Camel. As a result, resources updated or modified during this period were not propagated to a search index (Solr) and a triplestore. Regardless, relying on a JMS topic will invariably result in missed messages [1].

Although it was possible to pinpoint the precise time network connectivity was lost, it is impossible to rectify this situation presently except to reindex the entire repository because it is not possible to query Fedora for a list of changed, modified, or deleted resources since a specific point in time. In order to ensure the 2,000-5,000 resources created, modified, or deleted during the network loss were accurately reflected in the search index and triplestore, it was necessary to reindex 480,000 resources.

While it is possible to configure Fedora's embedded ActiveMQ to use a queue instead of a topic, and to further expand the infrastructure to include a distributed broker [2], it seems reasonable that a repository be able to provide a list of URIs of created, modified, and/or deleted resources from a specific point in time.

Such a feature could also assist with auditing the contents of a repository when compared against the documents indexed by Solr and mirrored in a triplestore (assuming *all* resources have been propagated to those applications). Were there Fedora messages that were not communicated and/or processed by Camel in the last 7 days and, if so, how many and what were they?

[1] <https://jira.duraspace.org/browse/FCREPO-2005>

[2] <https://wiki.duraspace.org/display/FEDORA4x/Setup+Camel+Message+Integrations>

List Repository URIs Based on Path

Kevin Ford - 6 Feb 2017

There should be a way to request a list of resource URIs under a specific repository path, whether it is the root path or a sub-path. Use case follows:

One overcast day in Chicago, we wished to check the response of each and every resource under a specific container path to specific HTTP methods. (As above, the weather had no bearing on our action. In fact, we can't even be certain it was overcast, but it was definitely winter and odds are, therefore, it was overcast. But I digress.)

Not starting with a set of Fedora-reported URIs – that is, relying on the data copied to a triplestore, for example – would be a problem because we needed to know whether Fedora had knowledge of the URI *before* we tested it. We therefore had to start with Fedora.

Because this was a custom, one-off exercise, we quickly wrote a Python script for this purpose. Naturally, it required logic to 'crawl' the Fedora repository starting at a specific container path. This wasn't difficult, but it required some special consideration to ensure efficient memory usage for what we knew would be a healthy list of URIs, a surprise recursion error (easily corrected, but still), and use of an RDF library to parse the RDF from Fedora (admittedly we could have achieved the same end by solely operating on a JSON serialization or even an XML serialization using JSON or XML parsing methods).

This strategy – crawling a repository starting from a specific path – has been implemented in Java at least twice [1, 2] and probably more times too. This seems like a basic enough desire that it seems reasonable to think someone has written Ruby code or PHP code to perform the same action.

Given that a list of URIs under a specific path is frequently needed for operational or administrative purposes such that multiple developers are (re)creating code for this purpose in a variety of languages, this would be a good indication of the value of such a feature. Implementing it directly in Fedora would also ultimately save the time of downstream developers.

[1] <https://github.com/awoods/fcrepo-java-client-etc>

[2] <https://github.com/fcrepo4-exts/fcrepo-camel-toolbox/tree/master/fcrepo-reindexing>