Meresco Reindex

About reindexing a server using the Reindex component.

Author: Seek You Too
Organization: Seek You Too
Contact: info@cq2.org

Contents

1 Introduction

2 Reindex

1 Introduction

The Reindex component can be instructed to reindex all records. This can be necessary when you've changed your Application DNA, e.g. added an extra field to your index for drilldown.

This document will explain how Reindex and some other components work together to reindex your system.

2 Reindex

Meresco can reindex itself if it stores the input data. Fields for indexing and crosswalks etc. are based on these input data. The reindex process will take care of fetching this input data and then starting the "normal" upload/index process again. The advantage is that it uses the same mechanism for indexing as records inserted from outside the system, e.g. records sent with SRUUpdate.

The Reindex component depends on the component Venturi and StorageComponent for reindexing. The StorageComponent will list all available record identifiers, Venturi takes care of fetching all necessary parts. This is best explained by example.

```
storageComponent = StorageComponent(storagePath, partsRemovedOnDelete=['meta'])
# part of a typical uploadHelix
uploadHelix = \
  (TransactionScope("record"),
    (Venturi(
      should=[
        ('metadata', '/document:document/document:part[@name="metadata"]/text()'),
        ('meta', '/document:document/document:part[@name="meta"]/text()'),
        ('header', '/document:document/document:part[@name="header"]/text()'),],
      could=[
        (’rating’, ’/rating:rating’) ],
      namespaceMap={
        ’document’:’http://meresco.com/namespace/harvester/document’,
        ’rating’: "http://meresco.com/namespace/record/rating"}
      ),
      (FilterMessages(allowed=[’getStream’]),
        (storageComponent,)
      ),
      (XmlPrintLxml(),
        (storageComponent,)
      ),
      ...
      # here follow components like crosswalk etc.
      ...
    )
  )
...
#the above uploadHelix is used as observer of the SRUUpdate component
#a typical reindex part will look like:
...
(PathFilter(’/reindex’),
  (IpFilter([’127.0.0.1’]),
    (ReindexConsole(),
      (Reindex(’meta’),
        (FilterMessages(allowed=[’listIdentifiers’]),
          (storageComponent,)
        ),
        (FilterMessages(disallowed=[’listIdentifiers’]),
          uploadHelix
        )
      )
    )
  )
)
...
```

Explanation of some components:

- Venturi, separates the incoming xml document with the mentioned xpaths. It takes care that all the should parts are present. An important feature for reindexing is that it will fetch the parts from the storage if not present in the input xml. Venturi is always at top of the uploadhelix and so enables the possibility of reindexing.
- StorageComponent, is responsible for storing and retrieving records and their different parts.
- ReindexConsole, handles a http request, sends a ’reindex’ message to the Reindex components and sends all identifiers back to the httpclient.
- Reindex, uses the listIdentifiers call to send each identifer to the uploadHelix so the real reindexing will take place.