Understanding integrated search

The integrated fulltext serach using Elatic search is a internal/active approach to indexing the content. Content will be added to a indexing queue every time it is updated - ensuring allways updated content, but consuming CPU ressources on the indexing server.

Beacuase file indexing is very CPU intensive, the file indexing functionality i seperated into a service that can run on a server seperated from te main application server. Anyway the fileindexer will run from a database queue, so in most cases seperation is not strictly required.

The basic search service requires

- TS file indexing service (queue handler)
- Elastic search server (search engine)

For multitenant setups a single TS file indexing service can service multiple instances, as long as they write requests to the same queue (using DB views). The Elastic search server can also handle multiple applications.

If PDF OCR functionality is needed the following components needs installation too

- Ghostscript (PDF to TIFF conversion)
- Tesseract (OCR library)

The above components for OCR must be installed on the file indexing server.

Behind the scenes

Indexes in Tempus Serva is stored in intermediate tables in the database. ElasticSearch indexes can be dropped an regenerated from the intermediate storage.

Data indexing

Data is mainly indexed in one large text blob.

> Submit data > Stored in **lucenedatastore** > Transfer to ElasticSearch

For solutions using version history data will be reused, in order to minimize the overhead.

File indexing

File indexes points to the record, not the file itself. Likewise permission checks will rely on read access to a record.

```
Upload file > Stored in lucenefilequeue > tsFileIndexingService > Transfer to ElasticSearch
```

The indexing service handles files in various conversion processes

- tsFileIndexingService > Apache Tika (most files)
- tsFileIndexingService > terrasect (tif images)
- tsFileIndexingService > GhostScript > terrasect (PDF images)

For multi application installation **lucenefilequeue** is made for sharing between applications.

ElasticSearch structure

Multiple applications can share the same ElasticSearch server

```
/ APPLICATION / SOLUTION / RECORD ID
```

Records contains the must general information

- Title
- Content (large text blob)
- SagID
- DataID
- FieldID (in case of subrecords)
- ModifiedAt
- ModifiedBy (UserID)

Serahc results are filtered against the Tempus Serva permission engine **on record level**.

Data and subrecords (such as files) are stored in the same area, with slight adjustment to their record ID: DataID + "f" + FileID

```
/tempusserva/crm/6541
/tempusserva/crm/6541f45
```

Revision #1
Created 10 April 2025 13:36:25 by Theis Villumsen
Updated 10 April 2025 13:36:50 by Theis Villumsen