SPIKE: Searching MARC Authority Records via MARC authority app

~~MODSOURCE-352~~ - SPIKE: Searching MARC Authority Records via MARC authority app CLOSED

Overview

This page is created with the purpose to define the technical approach to search and retrieve MARC authority records via MARC Authority app. The solution should be given considering the requirements listed below:

1) These results need to return via the UI.

Number of results returned	Time to retrieve all results and display results
1 million	3 seconds
1 - 3 million	5 seconds
3 - 5 million	8 seconds
5 - 10 million	10 seconds
10 - 15 million	12 seconds

2) Requirements for the HTTP response:

- Results should be paginated
- Offset and limit parameters will determine pagination
- Results should contain a JSON response with keys: records (array of associated instance UUIDS), recordCount (integer)
- A search with no results should have an empty records array and count of 0

3) The search needs to support such options:

- Keyword search
- Fielded search
- Phrase search
- Boolean (AND/OR/NOT)
- Exact phrase
- Truncation
- Wildcard

Reference page: Search MARC Authorities

Solution

Authority records are getting imported only into SRS in the current implementation of data-import. The SRS uses Postgres database to store the records. Most of the requirements, given above, include full-text search, are not supported by Postgres so efficiently as expected.

The solution is to use a search engine, that is designed to index and search data more productive. We will use the ElasticSearch. It's able to achieve fast search responses because instead of searching the text directly, it searches an index. It uses a structure based on documents instead of tables and schemas and comes with extensive REST APIs for storing and searching the data.

ElasticSearch is already used in FOLIO. It stores Instances, Holdings, and Items. The mod-search provides a REST API to search records in CQL format. The SRS will still store originally imported Authority records (MARC + Parsed) in Postgres, but the Search API will be provided by the mod-search. Here is documentation that explains supported search types and options: https://github.com/folio-org/mod-search/blob/master/README.md#supported-search-types

We can organize storing and searching Authority records similar to Instances/Holdings/Items, having added more steps to the Authorities import workflow.

Authorities import workflow

1. MARC file is uploaded from UI to **mod-data-import**
2. MARC records are packed into batches and put to Kafka queue (EVENT - DI_RAW_RECORDS_CHUNK_READ)
3. **mod-srm** reads batches from the queue, validates and passes to **mod-srs** via Kafka queue (EVENT - DI_RAW_RECORDS_CHUNK_PARSED)
4. **mod-srs** stores Authority records into PostgreSQL database and returns the result back via Kafka queue (EVENT - DI_PARSED_RECORDS_CHUNK_SAVED)
5. **mod-srm** reads the profile and creates JSON payload (containing parsed MARC, profile, mapping parameters) for processing. Exports it to an appropriate Kafka queue, one event per MARC entry - DI_SRS_MARC_AUTHORITY_RECORD_CREATED

6. **+ mod-inventory** receives the event, maps the incoming SRS Authority record to Authority domain object using default mapping rules. Stores it in **mod-inventory-storage** sending HTTP request
7. **+ mod-inventory-storage** receives the incoming Authority domain object, saves it into a database to start reindexing, sends Authority domain object in Event to Kafka
8. **+ mod-search** receives the event and indexes the incoming Authority record according to the indexing mapping file

   Steps 6,7,8 are missing and need to be implemented

Implementation steps

Here is a list of steps, each step covers some area in a corresponding module. I added raw estimations to the steps that Spitfire can implement, other steps require the help of the Falcon team in estimating.

- Create schema for Authority domain object ~~MODINV-504~~ - Create schema for Authority domain object CLOSED
- Create default Action profile and Mapping profile
  ~~MODDICONV-209~~ - Create default Action profile and Mapping profile for Authorities CLOSED
- Create mapping rules ~~MODSOURMAN-573~~ - Create mapping rules for AUTHORITY records CLOSED
- Create processor to generate Authority domain objects from Marc records by the mapping rules
  ~~MODDICORE-181~~ - Create mapper to generate Authority objects CLOSED
- Create CRUD REST API to receive Authority domain object and save it into storage
  ~~MODINVSTOR-787~~ - Create CRUD REST API for Authorities CLOSED
- Create Handler to generate Authority domain objects from Marc records
  ~~MODINV-501~~ - Create Handler to generate Authorities from MARC Authority records CLOSED
- Handle an update of Authority record via mod-quick-marc
  ~~MODINV-503~~ - Edit MARC Authorities via quickMARC | Update Inventory Authorities CLOSED
- Automatic creation of "authority" Kafka topic ~~MODINVSTOR-788~~ - Automatic creation of Kafka topic for Authorities CLOSED
- Sending of "Domain Events" to Kafka topic when Authority record is created/updated/deleted
  ~~MODINVSTOR-789~~ - Send "Domain Events" to Kafka Authorities topic CLOSED
- Add search API for Authority Records ~~MSEARCH-195~~ - Add search API for Authority's Record CLOSED
- Add /authorities/ids API for Authority Records search
  ~~MSEARCH-197~~ - POC: Add /authorities/ids API for Authority Records search CLOSED
- Refactor /index API to support Authority Records ~~MSEARCH-196~~ - Refactor /reindex API to support Authority Records CLOSED
- Create Karate tests to cover Authority Records search in mod-search
  ~~FAT-990~~ - Create Karate tests to cover Authority Records search in mod-search CLOSED
- Mod-search Authority Records search performance tests ~~PERF-200~~ - Mod-search Authority Records search performance tests CLOSED

Separate searching for heading values (1xx) and auxiliary values (4xx 5xx)

In order to support separate search in heading values (1xx) only and over heading values (1xx) and auxiliary values (4xx 5xx) the 2 separate fields should be created in Inventory (and Elasticsearch) for each field, that user should be able to search in.