



Understanding mod-search and Indexing

WOLFcon 2024
Felix Hemme, Antje Niemann
September 25, 2024

Our Motivation

Share knowledge about mod-search and indexing

- How does it work?
- What customization options are available?
- What should be considered when upgrading and reindexing?
- What is missing?





About us

- Library computer scientists at GBV head office (VZG)
- VZG operates FOLIO platform, on which 40 tenants are now running
- In the future, we will host a system of ~200 libraries
- Important for us: Scalability, reliability and performance
- Metadata stored in mod-inventory-storage (MIS) updated by mod-inventory-update (MIU)
- No MARC in SRS
- No authority records in FOLIO



Elasticsearch / OpenSearch

- Search engine technology has been in use for indexing data in the Inventory app (regardless of metadata source)
- Replaced the search with PostgreSQL since the Kiwi release
- Both Elasticsearch and the free fork OpenSearch are supported




mod-search

- Integration with Elasticsearch / OpenSearch for advanced search capabilities
- Provides a search functionality for instance and authorities via REST API
- Configurable indexing and search settings
- It uses the Contextual Query Language (CQL) as a formal language to query records using filters, boolean conditions, etc.



mod-search

- Documentation
 - [github folio-org/mod-search README](#)
 - [mod-search API Documentation](#)
 - FOLIO Wiki
 - [Search - using Elasticsearch or OpenSearch](#)
 - [mod-search](#)
 - [ElasticSearch Reindex Performance Recommendations](#)
 - [Open Search](#)
 - [Call Numbers Browse](#)
 - ...
- Slack channel **#metadata-management** (no dedicated #mod-search channel)
- [JIRA](#)  mod-search (MSEARCH)
Software project



How an metadata update is performed

- The event streaming service Kafka is used to communicate record changes in MIS to the OpenSearch index.
- Every change in MIS generates a Kafka event that describes the change.
- These events are published as messages in a Kafka topic and contain information about the type of change (insert, update, delete) and the data affected.
- mod-search is configured accordingly to subscribe to the Kafka topics.
- If a message is received from the appropriate topic, mod-search starts to process it and updates the indices accordingly.
- This makes it possible to always search the latest versions of the records. Exception: There may be a time delay if there is an increased data traffic.



How an metadata update is performed

Why not communicate the changes directly via the APIs?

MIS is the source of truth. Write processes in MIS are completed faster if changes can be outsourced to Kafka and there is no need to wait for mod-search first (asynchronous processing).

Our Findings





All Index

The screenshot shows the 'Inventory' application interface. The top navigation bar includes 'Inventory', 'Agreements', 'Bulk edit', 'Check in', 'Apps', and 'AMI Medienzentrum'. The main content area is titled 'Inventory' and shows '0 records found'. A large red question mark is centered on the screen. Below it, a message reads: 'No results found for "history". Please check your spelling and filters.' The search filter panel on the left is open, showing a list of search criteria. The 'All' option is highlighted in blue and is also enclosed in a red box. Other search criteria include Keyword, Contributor, Title, Identifier, ISBN, ISSN, OCLC number, Instance notes, Instance administrative notes, Subject, Effective call number, Instance HRID, Instance UUID, Authority UUID, Query search, and Advanced search. There are also expandable sections for 'Suppress from discovery' and 'Statistical code'.



All Index

- Search by all feature is optional and disabled by default
- It can be enabled for a tenant by [POST](#) to `/search/config/features`

```
{  
  "feature": "search.all.fields",  
  "enabled": true  
}
```

- Also, search by all fields can be enabled globally by passing to mod-search ENV variable:

```
SEARCH_BY_ALL_FIELDS_ENABLED=true
```



All Index

- When enabled, the All index can be used for Inventory searches
- Only records that have been edited since the All index was activated are indexed in it
- It is therefore usually necessary to reindex in order to make the existing metadata available

Search & filter <

Search Browse

Instance Holdings Item

All
water supply

Search

Reset all Advanced search

Inventory
66 records found

<input type="checkbox"/> Title ^	Contributors	Publishers
<input type="checkbox"/> Adapting to a changing Colorado River : making future water deliveries more reliable through robust management strategies / David G. Groves, Jordan R. Fischbach, Evan Bloom, Debra Knopman, Ryan Keefe	Groves, David G. ; Knopman, Debra S. ; Bloom, Evan T. ; Fischbach, Jordan R. ; Keefe, Ryan ; Environment, Energy, and Economic Development Program (Rand Corporation) ; Rand Justice, Infrastructure, and Environment (Organization) ; Rand Corporation ; United States. Bureau of Reclamation	RAND (2013)
<input type="checkbox"/> Adaptive Strategies for Water Heritage : Past, Present and Future / edited by Carola Hein	Hein, Carola *1964-*	Springer (2020)



All Index

- The All index can be deactivated by sending a [DELETE](#) to `/search/config/features/search.all.fields`



Recreating the Index

Via POST to `/search/index/inventory/reindex`

- recreateIndex: **false** (default) / true
 - false → Old index is kept until new index is built.
 - true → Existing index is dropped before building the new one.

```
POST {{baseUrl}}/search/index/inventory/reindex

Params Authorization Headers (12) Body Scripts Se

none form-data x-www-form-urlencoded raw

1 {
2  "recreateIndex": true
3 }
```

- resourceName: **instance** (default), authority, location, linked-data-instance, linked-data-work, linked-data-authority



Recreating the Index

- Monitoring is not really implemented.
- With the UUID of the indexing run, one can use the API `/instance-storage/reindex/<uuid>` to see under “published” how many records have been **staged** for indexing.

```
{
  "id": "d8dacc4b-6b4d-4250-a897-7b519f38bef0",
  "published": 5699897,
  "jobStatus": "Ids published",
  "submittedDate": "2023-07-26T06:40:14.190+00:00"
}
```

- Missing: Information about the progress of the indexing process and the end time



Language Analyzer

- Each tenant is allowed to pick up to 5 languages from pre-installed list for indexes (e.g. title, contributors etc.). This can be done by [POST](#) to `/search/config/languages`

```
{  
  "code": "eng",  
  "languageAnalyzer": "eng-analyzer"  
}
```

- The code here is an ISO-639-2/B three-letter code. There is a list of pre-installed language analyzers in the [mod-search README.md](#)
- Adding a language requires a whole reindex
- Initial languages can be defined via ENV variable `INITIAL_LANGUAGES`. If the variable is not defined, only eng code is added.



Discrepancy between MIS and OpenSearch Index

mod-inventory-storage	mod-search
<pre>/instance-storage/instances?limit=0&query=cql.allRecords=1 /instance-storage/instances?limit=0&query=hrid==* /instance-storage/instances?limit=0</pre>	<pre>/search/instances?&limit=0&query=cql.allRecords=1 /search/instances?&limit=0&query=hrid==*</pre>

```
{  
  "instances": [],  
  "totalRecords": 315418,  
  "resultInfo": {  
    "totalRecords": 315418,  
    "facets": [],  
    "diagnostics": []  
  }  
}
```

```
{  
  "totalRecords": 315418,  
  "instances": []  
}
```



Discrepancy between MIS and OpenSearch Index

- If there is a discrepancy between MIS and OpenSearch, we want to know what records are affected.
- We are using a Python script that retrieves the HRIDs from the SQL database of MIS (faster than an API request) and OpenSearch.
- Enables the output of the HRID of records that are missing in MIS or mod-search.



Use of mod-search in GBV

- Indexing of Inventory records for searching in Inventory App
 - resourceName: **instance** (default)
- In combination with OpenSearch
- Option “All index” is activated for all tenants by setting the ENV variable `SEARCH_BY_ALL_FIELDS_ENABLED=true`
- Language Analyzers `eng` and `ger`
- Default settings seem to work well in most cases
- Reindexing of 5 mio titles in less than three hours (`recreateIndex : true`)
- Python script to extract discrepancies between MIS and mod-search

Discussion and Knowledge Sharing





Comments or additions to our findings?

What are your experiences with mod-search and indexing?

What improvements to mod-search and indexing seem most urgent to you?



Thank you

Antje Niemann (antje.niemann@gbv.de)

Felix Hemme (felix.hemme@gbv.de)