

Linking MARC bib fields to MARC authority headings /references

 MODQM-254 - Technical design: approach for manual linking/unlinking MARC bib fields to MARC authority headings/references (Names)
CLOSED

Purpose

Catalogers want the ability to link a MARC bib field(s) to MARC authority heading(s)/reference(s) because authority records are seen as the source of truth about a person/place/corporation/conference/subject/genre. This linking allows the library user to learn more and feel 100% confident that the information provided is accurate and allows them to search/browse for their research.

Scope

Phase 1 will focus on handling Personal/Corporate/Meeting names. Additional spikes will be created for the technical design for handling Subjects (i.e. Topical terms, Genre/form) and Titles (series/uniform titles)

Business requirements

- Applies to MARC bib and MARC authority record types only. Cannot link any record type with source = FOLIO.
- Linking a bib field(s) to an authority heading(s)/reference(s) will be based on the rules defined in the section called link/matching rules
- Linking should also indicate if the user has linked the MARC bib field to the Authority record field that is authRefType = Authorized OR authRefType = Reference
- Once link is made need to support an indicator to support the following
 - On Inventory app > Browse list > Indicate which contributors/subjects/and future browses such as serials/genres/titles are controlled by an authority record
 - On Instance record and View bib source > indicate if MARC bib tags are controlled by MARC authority heading
 - On MARC authority app > Search results list and Browse list > indicate number of bib records tied to authority records
 - Dashboard app > Widget > Allow for the creation of a widget that allows a cataloger to view which bib records have been linked to an authority record AND if possible time+date stamp to support a widget that shows the number of bib records linked to an authority record over the past 30 days for example. AND a widget to show the number of bib records unlinked over the past 30 days for example. Or just the number of bib records linked OR unlinked as widgets.
 - Inventory app/MARC authority app > In-app report > Allow for the creation of a widget that allows a cataloger to view which bib records have been linked to an authority record AND if possible time+date stamp to support a widget that shows the number of bib /authority records linked to an authority record over the past 30 days for example. AND a widget to show the number of bib/authority records unlinked over the past 30 days for example. Or just the number of bib/authority records linked OR unlinked as widgets.
 - Inventory app/MARC authority app > Facet or Filter > Allow a user to filter search or browse lists by whether bib or authority record is linked.
- Design needs to consider that future development will support linking/unlinking source = FOLIO
- Design needs to consider performance when saving links and the entire bib record. Also needs to consider performance / length of time before that link is reflected on MARC Authority app search/browse list
- Designs needs to account for optimistic locking
- Design needs to consider editing and deriving new bib record workflow
- Design needs to consider handling deleting a MARC authority record that is linked to a MARC bib record
 - We can allow the deletion to occur but need to support a warning message to inform a user the impact of deleting the MARC authority record to the linked bib records. Description of impact will be under the section called unlink behavior handling.
- User will need permissions assigned to link/unlink MARC bibs + MARC authority records
- Also design should consider that once link is made then provide a link to the linked MARC authority record associated with the applicable MARC bib tag /field

Additional details: Link behavior handling

- When link happens Then the linked MARC bib tag, linked MARC bib indicator, linked MARC bib subfield code(s) and linked MARC bib subfield value(s) will be read-only
 - User may add additional bib subfield code and bib subfield value AND they will not be linked AND will be editable
- Also hardcode the linked MARC bib field/tag \$0 as a matching value to be used for automated linking in the future.

Unlink behavior handling

- Allow a user to unlink a MARC bib field/tag from a MARC authority record with the following actions
 - a.) Unlink the MARC authority from the MARC bib tag/tag via quickMARC bib record
 - b.) Delete MARC authority tag linked to a MARC bib tag via quickMARC authority record
 - c.) Delete the MARC authority record via MARC authority app
- When any of the above unlink actions occur Then formerly linked MARC bib tag, MARC bib indicator, MARC bib subfield code and MARC bib subfield value are editable. The link/match between the MARC bib field/tag to MARC authority record no longer exist. Maintain the contents that were populated by the authority record.
 - d.) Delete the MARC bib tag via the quickMARC bib record

- When this unlink action occurs Then display an updated Delete MARC tag field confirmation modal to inform user that deleting MARC bib tag (s) will also remove link between MARC authority records.

MARC bib + MARC authority linking rules

Description	Controlled MARC bib tag	MARC bib indicator code 1	MARC bib indicator code 2	MARC bib subfield codes - minimum	Controlling to MARC authority tag	Controlling MARC authority subfield codes - minimum
Personal name	100	/	/	a	100	a (whatever subfields are present will populate MARC bib 100 and control it.)
Subject Added Entry - Personal name	600	/	/	a	100	a (whatever subfields are present will populate MARC bib 100 and control it.)
Added Entry - Personal name	700	/	/	a	100	a (whatever subfields are present will populate MARC bib 100 and control it.)
Corporate name	110	/	/	a	110	a (whatever subfields are present will populate MARC bib 110 and control it.)
Subject Added Entry - Corporate name	610	/	/	a	110	a (whatever subfields are present will populate MARC bib 100 and control it.)
Added Entry - Corporate name	710	/	/	a	110	a (whatever subfields are present will populate MARC bib 100 and control it.)
Meeting name	111	/	/	a	111	a (whatever subfields are present will populate MARC bib 111 and control it.)
Subject Added Entry - Meeting name	611	/	/	a	111	a (whatever subfields are present will populate MARC bib 111 and control it.)
Added Entry - Meeting name	711	/	/	a	111	a (whatever subfields are present will populate MARC bib 111 and control it.)

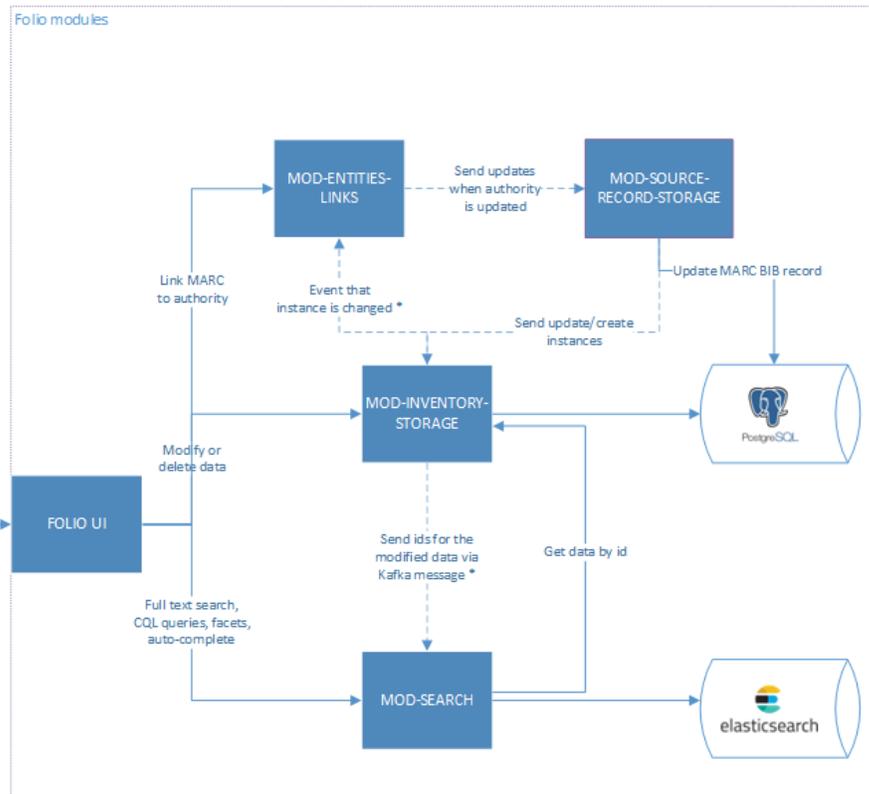
Errors

- If a user attempts to link a MARC bib tag 100/110/111 to a MARC authority tag that does not match with what is on the above list then show an error message that the link is not allowed please select an authority record with the corresponding personal/corporate/meeting name heading type.

NFRs

The instances/bibs could be updated during some time (up to 10 minutes for popular authority) after authority is updated.

Implementation



The main implementation of the functionality should be done in the new module called "mod-entities-links". Also it should be done in SRS, mod-inventory-storage, mod-search and on UI of quick marc.

Mod-entities-links

The new module should be created for handling links between Instances fields and authorities. The module should contain the following database (postgres) table ("linking_table").

id	authority_id (btree index)	instance_id (btree index)	bib_record_tag	bib_record_field_id	linked_sub_fields
Integer, auto_increment	The same authority id for Inventory and SRS	Instance Id, which SRS currently knows	the number (e.g. 100 or 700)	The value of the id field (e.g. \$0).	The characters (numbers or alphabetic) of existing fields of authority records, that will control the subfields of linked bibs.

For the bib_record_field_id column, the \$0 sub-field should contain the value which is generated based on [Managing Authority source files](#).

API

- `POST /instances/{instance_id}/link` - Should create row in the table (see above).

Messaging

Mod-entities-links listens for inventory.authority domain events Kafka topic. If the message is received the there is such authority in link table then the record in the following table ("job_table") is created:

id	authority_data	processed_count	completed	next_record_id
Integer, auto_increment	json field, that contains the authority data from Kafka_message	number of processed records	is completed flag	id latest of the row in the links table for the current batch

Then mod-entities-links iterates over all records in links table that has the authority_id as in Kafka message by batches of 500 records (of "linking_table") and after every batch is processed update the `next_record_id` field with the latest id in the batch and increment value (by the number in batch) of the processed_count. And the next batch takes the records from linking table, that has `id > job_table.next_record_id`. `NextRecordId` is also saved in memory.

Saving in database is done for ensuring the consistency in case if module fails during the processing of batch. On @PostConstruct of the module it should read the job_table and if there is job with completed equals to "false" it should start processing of this job starting from *next_record_id*. It could cause double processing of the job if another instance is up during the job processing, but it is not a problem, as the process is idempotent.

For every row in a batch mod-entities-links should:

1. Send Kafka message to update SRS to update the corresponding bib records. The \$9 subfield should be populated with authority id. The similar mechanism in SRS could be leveraged that was implemented for propagation of mod-inventory *instance_id* to SRS. mod-entities-links should request SRS for the mapping between authority inventory fields and source authority record fields only once for the batch.
2. When SRS updates the record, the change is automatically propagated to mod-inventory-storage and mod-seach (as it is right now).

Consequent modification of the same record (nice to have)

Before start of the job_processing mod_links should check if the job for such authority already in progress and if it is so, it should stop such process and make this job *completed*.

Source-record-storage

The method that is used for propagation of mod-inventory id to SRS (via Kafka messaging) should be adjusted in order to support updating of any field of the record, so that it should support updates from mod-entities-links.

If \$9 is populated in the MARC record that is being imported and there is existing MARC record in SRS that will be updated, the source-record-storage should request mod-entities-links to get the fields that should be updated (*linked_sub_fields*) and *authority_id*. All subfields of fields except the linked fields should be retained. Mark protection rules should be ignored for it.

Source-record-manager

Mapping profile should be adjusted to take the value from the \$9 field and set it to the subfield "*authority_id*" in "contributors" field. For the update, the *authority_id* should be requested from mod-entities-links by *instance_id* and *bib_record_field_id*.

Inventory-storage

The sub-field "*authority_id*" should be added to the "contributors" field in mod-inventory-storage. This sub-field should contain the value of the \$9 sub-field of MAR bib field.

Existing functionality: When the changes to authority happens in inventory the Kafka domain event message is sent accordingly and it triggers changes in mod-search and mod-entities-links.

Mod-search

The "*authority_id*" subfield should be added to the "contributors" field. It should be *keyword* field.

For the contributors browsing the contributors index should be changed, so that if contributors with the same names has different *authority_id*, it should be different records in the index. The mark near the record in the UI table should be shown based on the "*authority_id*". If the filtering for having link to authority should be performed, the ES query should be changed to have "*authority_id*" in *EXISTS* clause.