

Librarian Merge Queue

Resources

- Flagging Duplicates: Adapt Librarian bar to suggest Merge Requests · Issue #4071
- Proposed:  Librarian Merge Queue UI

Background

Open Library uses a variety of means to expand its catalog, some of which are fully automated. The problem of identifying duplicate records is not trivial, and duplicates are inevitable. In order to keep our catalog tidy, we should aim to keep duplicates to a minimum. One way of doing so is to merge duplicate records into a single, primary record.

Merging records takes a great amount of care, as it can be quite difficult to undo an incorrectly merged record. Because of this, a very small number of people are permitted to merge records in Open Library. Regardless of this limitation, there are many librarians without merge permissions that suggest merges in the #openlibrary-merges-g Slack channel. Merge requests in this channel are ultimately handled by librarians who have merge permissions.

The process of flagging duplicate records, reviewing the flagged records, and either merging the records or rejecting the request could be more efficient if the merge requests were handled entirely in Open Library.

Proposal Overview

The goal of the librarian merge queue is to provide librarian merge tools to a larger group of Open Library patrons. If permissions are granted, patrons can suggest merges using the same UIs available to admins today. When a patron without merge permissions submits a merge request, the request will be added to a queue and will be reviewed by librarians with merge permissions. Merge tools will function as per usual for librarians with merge permissions (the merge will occur instantly on submission).

Librarians with merge permissions should be able to access a paginated view containing a list of summaries of all pending merge requests. Each list item can be deleted from this view. Clicking on a merge request list item should provide more details about the request. Ideally, the request would be displayed in the merge tools, and the librarians can modify the request if needed and merge at their discretion.

While a full-blown notification system isn't needed, merge approvers will still need to know when there are pending requests in the queue. This could be as simple as displaying a badge count somewhere in the top nav whenever there are items in the queue.

Why?

- Want to reduce duplicates on site, without introducing errors
 - When books are added to our catalog, sometimes duplicate records are created. These duplicates can be merged into a single record, but some merges are difficult/impossible to undo. Because of this risk, a few well-trained folks are permitted to make merges.
- Reducing the amount of time Librarians spend on manually fielding merge requests
 - Today, we have many volunteers working on improving our catalog of books. When they identify duplicate records, they post all of the OLIDs of the duplicates in the #open-library-merges channel in Slack. Librarians later review the suggested changes and merge the records, if the request was approved. This process is cumbersome.
- Engagement: Increasing contributions from patrons.

What?

- Make librarian merge tools available to a larger audience. These folks will be able to request merges, but not actually merge records.
- Display merge requests in a table, from which authorized librarians can approve or reject a request. Librarians can also leave notes explaining why a request was rejected (or any other information that they would like to share).
- Create a means of notifying authorized librarians that there are unresolved requests in the queue.

Risks

- This could create more work for librarians if everybody can request merges.
- Only 50 editions are included in a work merge --- all others are lost.
- Complexity of solution.
- Making sure that local DB works properly.
- Only trusted group should be allowed to use this

Prerequisites

1. ILE needs to get merged → needs a **decision** on this call e.g. timeline, solution (this milestone, whether it's Jim or Drini)
2. Modifications re: permissions group (right now it requires bot/api access?)
3. Some sort of table where the enqueued URLs go
 - ☐ **Decision:** @mek to locate our schema conversation
4. Design requires discussion

Decisions

- Allow patrons to submit onto the queue indiscriminately, however, once a librarian has gone to the merge UI url and selected editions, then the idea is to ~prevent merge (e.g. disable btn) if any non-primary work has > 50 editions.
- For now, author merges out of scope, at least for v0

Phases

Q: Is there a smaller v0?

- ☒ V0: <https://github.com/internetarchive/openlibrary/pull/6051>
 - Components
 - Database schema - community_edits_queue
 - Models - Edit.py core
 - Controller - plugins/upstream/edits.py (GET, POST)
 - View - pendingchanges.html (like recentchanges.html)
 - Integration
 - A page where one can view `requests`
 - A page where one can submit `work\_ids`, `author\_ids` to propose merges or any other ID to request a deletion
 - No integration into ILE
- ☐ V1: A **/usergroup/trusted-editors** has a ILE bar, can click on search results, and can click the “Propose Merge” button (which adds the word “Propose” because of their role) and this submits to the **merge_requests** table. A **/usergroup/librarian-mergers** at this stage sees no notification of new merge requests, however they may go to **/librarians/merge-queue** “**Merge Queue**” and see a list of incoming proposed merges. A librarian-merger then clicks on the url/link within the row of the table, which passes along a “Merge Request ID” **mrid=123** parameter within the URL. The **librarian-merger** proceeds to the “**Merge Works UI**”, they either click **Reject** (which updates the merge_requests db) or merger selects the right settings and runs **Merge** (which also updates the db table).
 - Q: Right now permissions require bot access? How to change to librarian-merger
 - We don’t need to solve this right now since bot access is fine for existing folks w/ merge permissions, and the goal here is to extend community of proposers
 - ILE merged into main
 - Create the db schema
 - Create branching fork in ILE bar (if librarian v. editor)
 - Create an ia/core.py model for interfacing w/ db schema (mergerequests.py)
 - Create a new route + html page for /librarians/merge-queue
 - Update the ILE Merge Works UI to respond to mrid + present option for reject/merge → we’d need to update the js endpoint to handle reject v. merge + call the ia/core.py mergerequests.py model
- V2: Lots of design work if we wanted to get this thing out to all patrons (maybe?)

- Open **Q's**: May not use ILE bar + may not use clicking on works to propose merge

User	Date	Actions
<u>Jim C.</u> Wants to merge 3 works into <u><i>Harry Potter and the Philos...</i></u>	4 days ago	[View merge] [v]

Tasks

- Create usergroup(s) for suggesting merges and approving merges
 - **/usergroup/trusted-editors**
 - Most of the people in slack channel
 - Can request merges
 - **/usergroup/librarians**
 - Smaller group of people (Lisa, etc)
 - These are a smaller subset
 - **/usergroup/librarian-mergers**
 - Even smaller group of people have been trained to do merges (Lisa etc)
 - can do work merges
- Ensure that the integrated librarian tool is restricted to the previously identified usergroups
- Merge integrated librarian tool into master
- Create DB table for merge requests
 - Fields:
 - ID: Primary identifier
 - Submitter: username of person that made the request
 - Status: Either "Pending", "Merged", or "Rejected"
 - Merge_url: URL of the merge request
 - Reviewer: The username of the person who reviewed the request
 - Note: Short note from reviewer (json blobs (can store timestamps, etc))
 - Created_on: Request timestamp
 - Modified_on: Update timestamp
- Create UI for merge request table (/librarians/merge-queue or something like this)
- Create means of notifying merge approvers when there is an item in the queue (put this in ILE blue bar).

