SUNY Geneseo
KnightScholar

Conference Presentations | IDS Project Proceedings Collection

7-25-2024

Using Alma APIs, SRU Lookups, and IDS Logic to Enhance Alma/ILLiad Integration

Timothy Jackson

Bill Jones

Angela Persico

Follow this and additional works at: https://knightscholar.geneseo.edu/ipp-conference-presentations

Recommended Citation

Jackson, Timothy; Jones, Bill; and Persico, Angela, "Using Alma APIs, SRU Lookups, and IDS Logic to Enhance Alma/ILLiad Integration" (2024). *Conference Presentations*. 50.
https://knightscholar.geneseo.edu/ipp-conference-presentations/50

This Presentation is brought to you for free and open access by the IDS Project Proceedings Collection at KnightScholar. It has been accepted for inclusion in Conference Presentations by an authorized administrator of KnightScholar. For more information, please contact KnightScholar@geneseo.edu.

Using Alma APIs, SRU Lookups, and IDS Logic to Enhance Alma/ILLiad Integration

Timothy Jackson, Fulfillment and Resource Sharing Program Manager, SUNY OLIS
Bill Jones, Digital Resources and Systems Librarian, SUNY Geneseo
Angela Persico, Resource Sharing and Reserves Coordinator, UAlbany

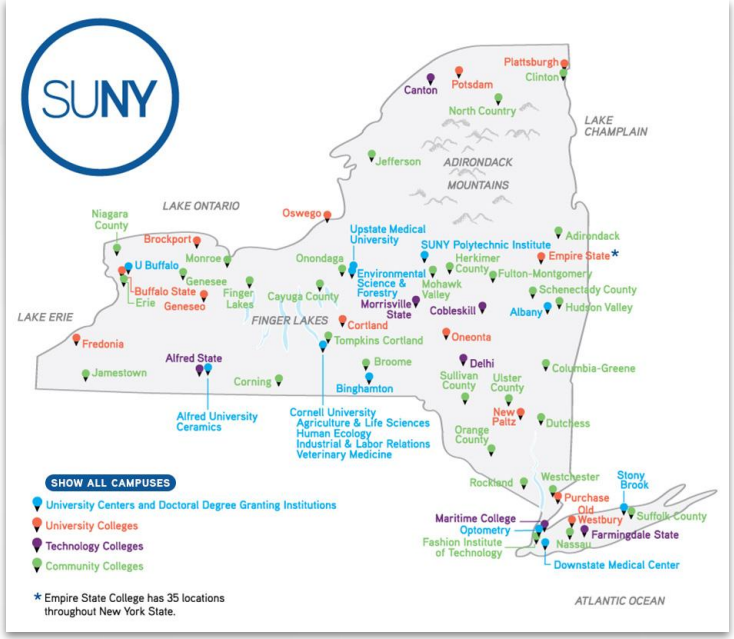
The backstory

https://vocal.media/geeks/spider-man-s-origin-story-what-secrets-could-be-hidden-in-the-tale-marvel-hasn-t-told

- SUNY system is comprised of 60 campuses located throughout New York state
  - University centers, comprehensive colleges, technical colleges, and community colleges

- SUNY traditionally used OCLC for resource sharing
  - ILLiad & WorldShare ILL

- All 60 SUNY campuses implemented Alma/Primo in 2019



- Alma now being used for physical resource sharing at nearly all SUNY libraries

- Some SUNYs also use Alma for digital resource sharing, but many continue to use ILLiad
  - Alma article lender network still small
  - ILLiad's compliance functionality is much better

- ILLiad request forms still available at many SUNY libraries, so some users continue to submit physical resource sharing requests in ILLiad instead of Alma

- Many of physical resource sharing requests submitted via ILLiad are for items that are locally available

- Library staff have to manually create hold requests in Alma, which is time consuming for library staff and increases request turnaround times for users

- Question from the University at Albany: Can we automatically convert an ILLiad request to an Alma hold request?

- Alma resource sharing requires less staff labor
  - Borrowing requests automatically sent to potential lending libraries
  - Lending requests automatically attached to record for requested item

- Alma resource sharing is faster than ILLiad
  - Alma turnaround times: 6-7 days
  - ILLiad turnaround times: 8-9 days

- Alma resource sharing usually provides longer loan periods (16 weeks) and allows users to manage all loans and requests within same system

- SUNY has done a lot of work to expand its Alma resource sharing network
  - Currently partnering with CUNY, CSCU, WRLC, PASCAL, FLVC, CSU, Orbis-Cascade, OCUL, and about a dozen independent institutions
  - All partnerships are fully reciprocal and have a 16 week loan period

- Alma resource sharing more cost effective than ILLiad & WorldShare ILL

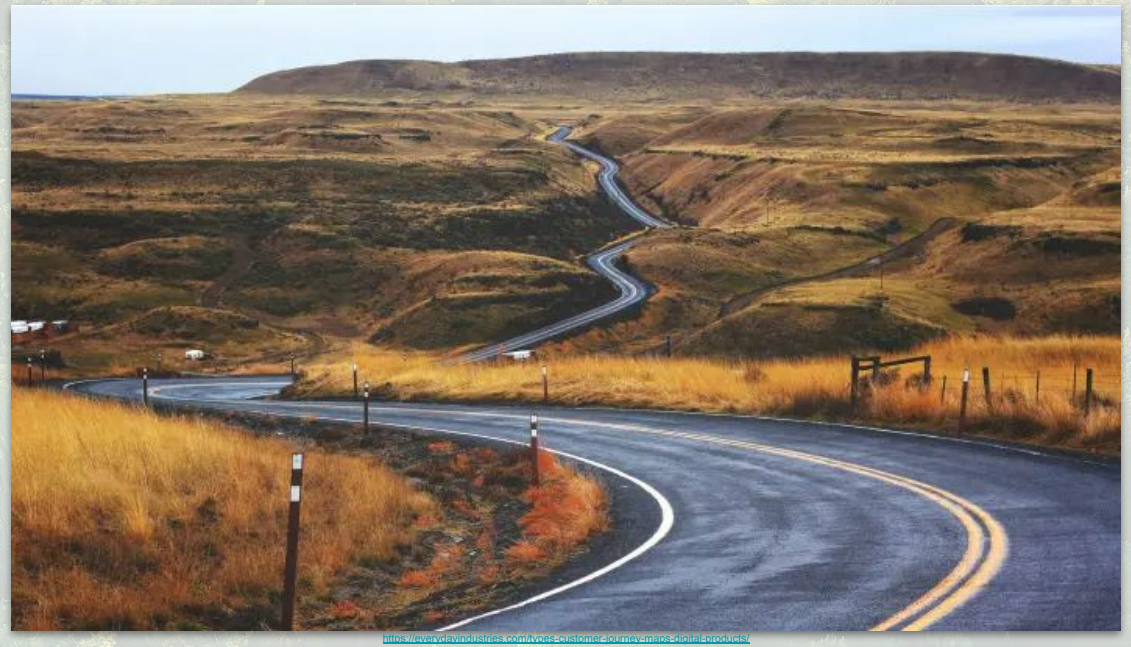
- My question: Can we automatically convert an ILLiad request into an Alma borrowing request?

What are we hoping to accomplish?

- Our goal is to create an ILLiad server addon that can automatically do the following things:
  - Convert an ILLiad borrowing request to an Alma hold request if the requested item is locally available
  - Convert an ILLiad borrowing request into an Alma borrowing request if the requested item is not available
  - Identify duplicate requests & requests for ebooks

- We hope that this addon will help us:
  - Reduce workload for library staff
  - Provide faster service to users
  - Reduce our resource sharing expenses

Adrien Coquet



https://everydayindustries.com/types-customer-journey-maps-digital-products/

Unintended Journey

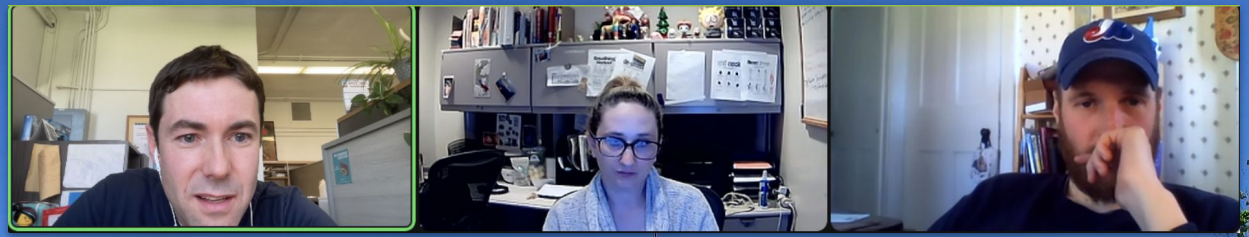
It all started as a Hold Requester Addon…

Once upon a time, in Upstate New York…

Hey Bill,
Do you think we could send a Hold Request to Alma from ILLiad?

Yes, I will build the thing.
SUNY GENESEO
89.3
WGSU

And then later that semester…



Hey Bill,
I have a better idea. Do you think you could make it so that we can send a Borrowing Request to Alma from ILLiad, too?

Absolutely.

Made with love.

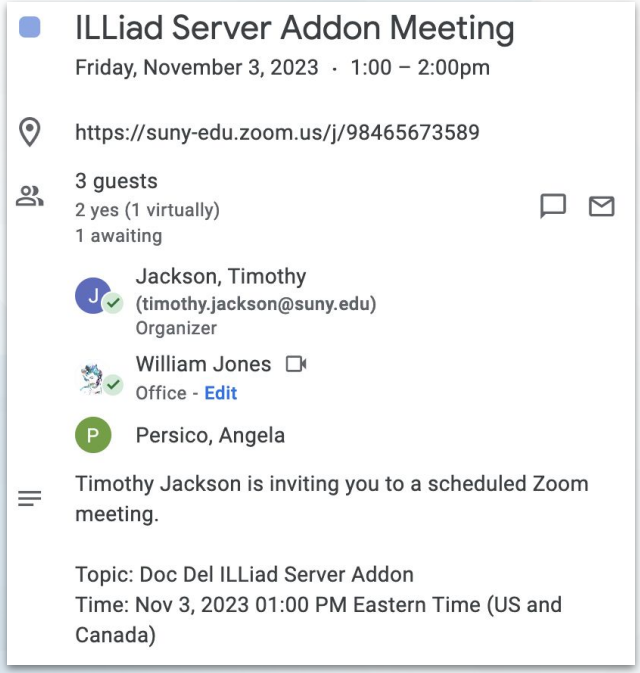
https://www.eatthis.com/burger-king-reclaim-the-flame/

Please keep in mind

Libraries have the option to submit:

- Only Hold Requests
- Only Borrowing Requests
- Both Hold and Borrowing Requests

Development Timeline



- Work on the Addon began November 3, 2023
- We have been meeting (almost) every Friday to discuss progress and enhancements
- During the week, we communicate via email and send logs to address potential bugs
- We continue to meet weekly

https://www.stewartsshops.com/news/great-all-day-food-options-at-stewarts/

System Requirements

1. ILLiad Customization Manager
2. Alma Users API with Read/Write
3. Alma SRU Lookup Enabled

Types of ILLiad Addons

Client Side Addon

- Installed in the ILLiad\Addons directory
- Run within the Client using a ribbon and buttons
- Error logs appear in C:\Users\username\Documents\ILLiad\Logs\ILLiadClient.txt
- Refreshed using Manage Addons -> Reset Cache

System Addon

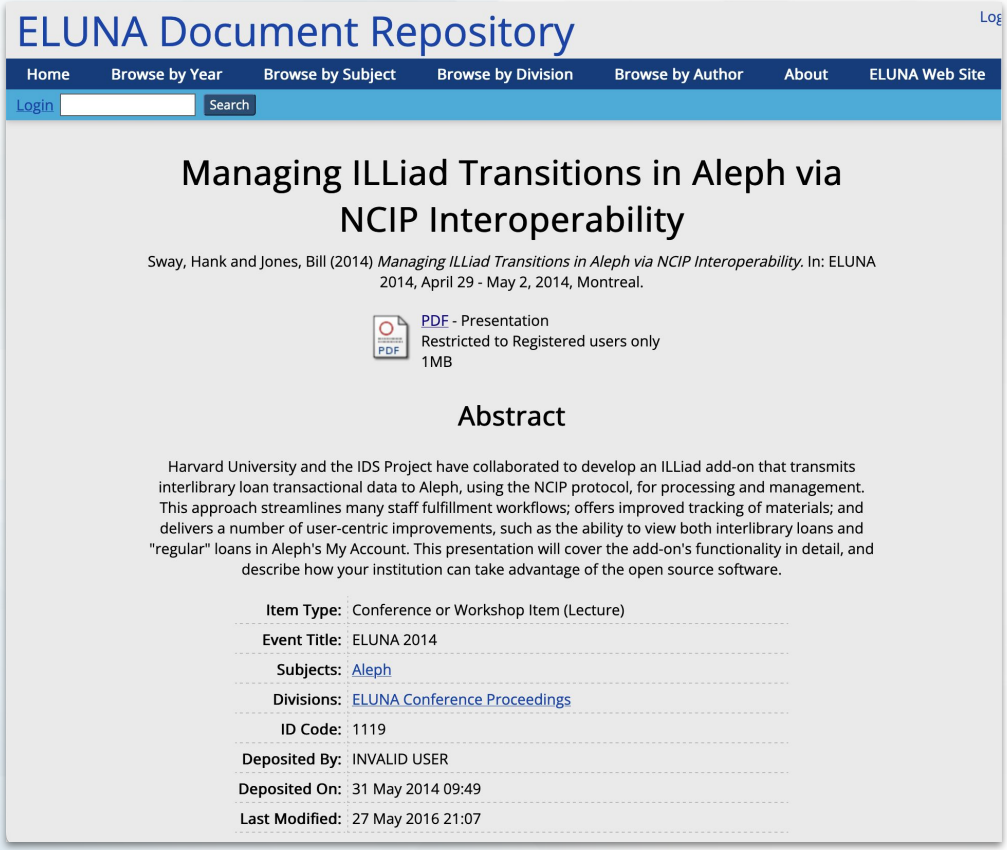
- Installed in the ILLiad\Addons directory
- Run within the Client through Registered Event Handlers
- Error logs appear in C:\Users\username\Documents\ILLiad\Logs\ILLiadClient.txt
- Refreshed by restarting ILLiad

Server Addon

- Installed through Customization manager and saved in a database
- Run outside of the Client on the server at specific time intervals
- Error logs appear on server in C:\Users\username\Documents\ILLiad\Logs\ILLiadClient.txt
- Refreshed by increasing the version number in the Config window

Runs every 'X' minutes

Improving my H-Index: Quick Citation Shout Out!



ELUNA Document Repository

Log

Home | Browse by Year | Browse by Subject | Browse by Division | Browse by Author | About | ELUNA Web Site

Login | Search

Managing ILLiad Transitions in Aleph via NCIP Interoperability

Sway, Hank and Jones, Bill (2014) *Managing ILLiad Transitions in Aleph via NCIP Interoperability.* In: ELUNA 2014, April 29 - May 2, 2014, Montreal.

PDF - Presentation
Restricted to Registered users only
1MB

Abstract

Harvard University and the IDS Project have collaborated to develop an ILLiad add-on that transmits interlibrary loan transactional data to Aleph, using the NCIP protocol, for processing and management. This approach streamlines many staff fulfillment workflows; offers improved tracking of materials; and delivers a number of user-centric improvements, such as the ability to view both interlibrary loans and "regular" loans in Aleph's My Account. This presentation will cover the add-on's functionality in detail, and describe how your institution can take advantage of the open source software.

Item Type:	Conference or Workshop Item (Lecture)
Event Title:	ELUNA 2014
Subjects:	Aleph
Divisions:	ELUNA Conference Proceedings
ID Code:	1119
Deposited By:	INVALID USER
Deposited On:	31 May 2014 09:49
Last Modified:	27 May 2016 21:07

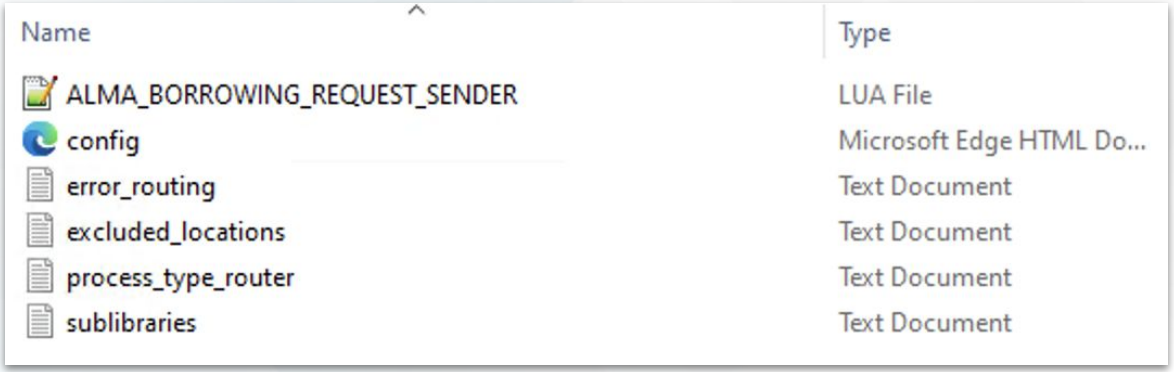
That last slide was from my 2014 ELUNA Presentation in Montreal with Hank Sway.

Some same ideas and techniques apply with webClient and XML messages.

2014 Slides:

https://documents.el-una.org/id/eprint/1119/

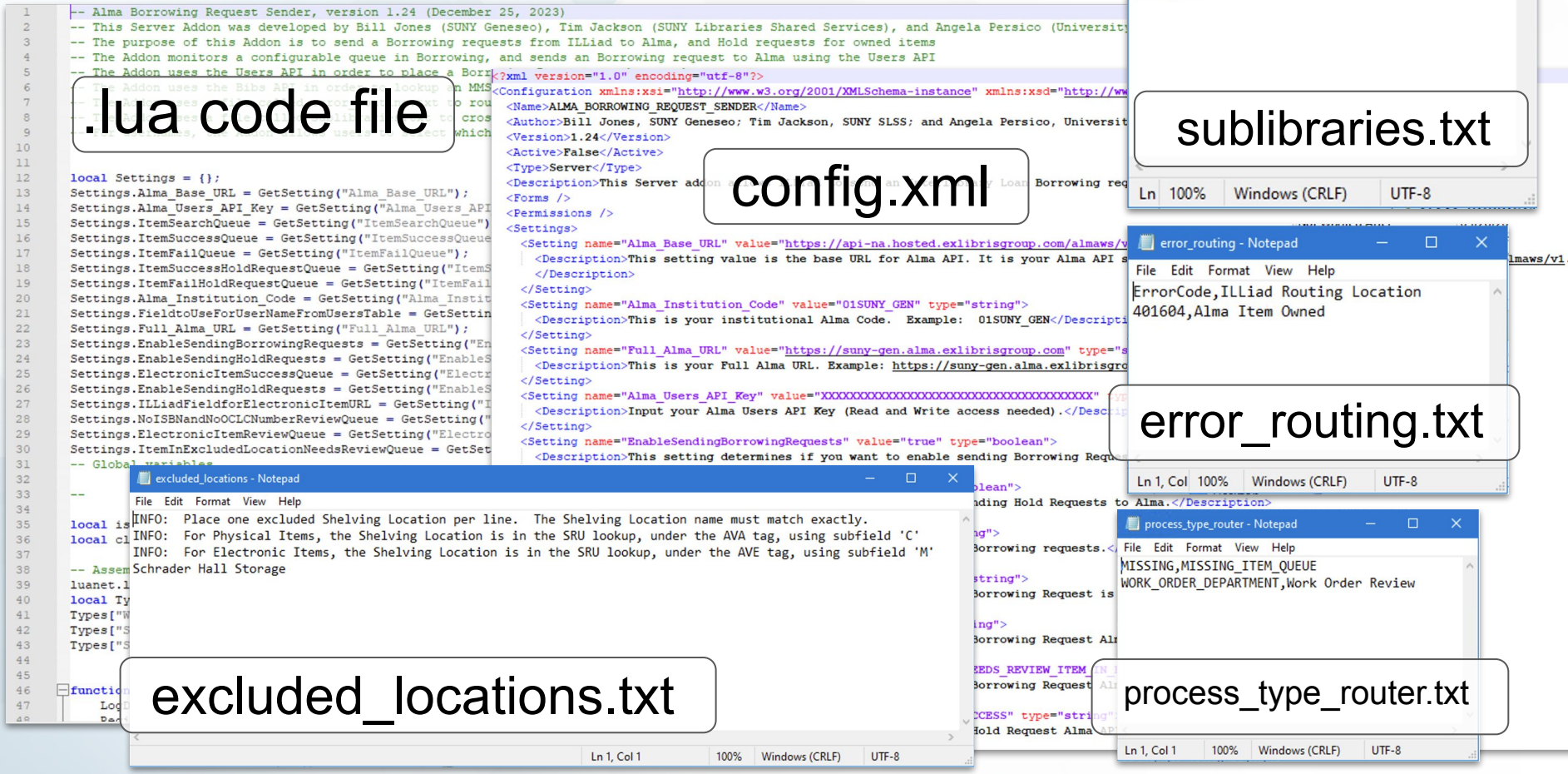
What does this ILLiad Server Addon look like?



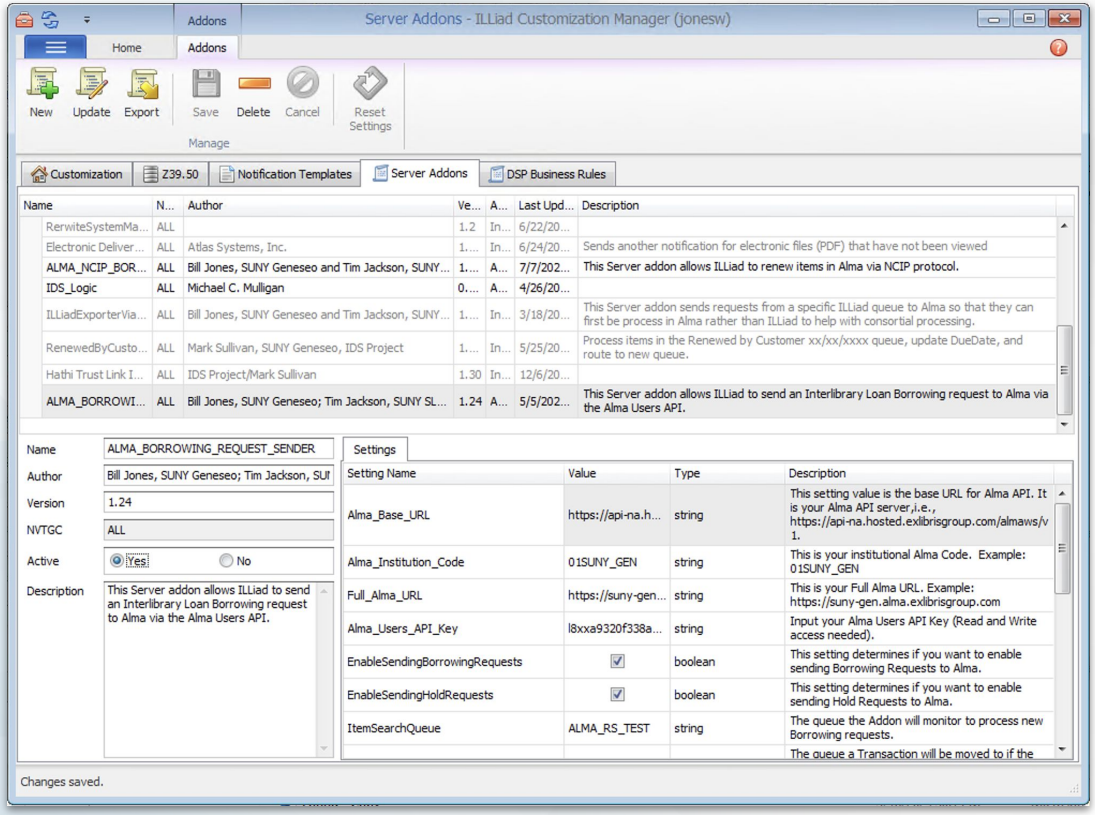
This Addon is distributed in a .zip folder and contains 6 files.

The folder contains the Addon code (in .lua), the configuration file (in .xml), and four .txt files. One for sublibraries crosswalk, one for listing excluded locations, one for the process_type_router, and one for custom error routing.

Screenshots of file contents (6 files)



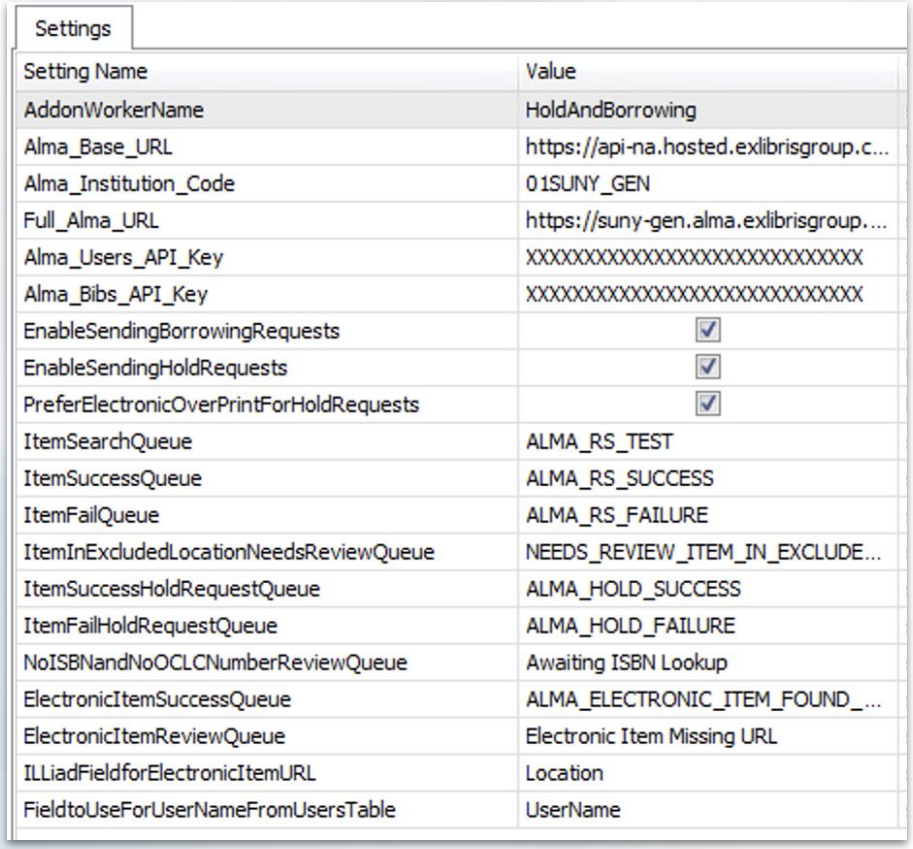
How do you install the Server Addon into ILLiad?



The Addon is installed through the ILLiad Customization Manager.

Click ‘New’, select the .zip folder, and click the ‘Yes’ radio button for the ‘Active’ setting.

How to configure the Server Addon for your library



Settings

Setting Name	Value
AddonWorkerName	HoldAndBorrowing
Alma_Base_URL	https://api-na.hosted.exlibrisgroup.c...
Alma_Institution_Code	01SUNY_GEN
Full_Alma_URL	https://suny-gen.alma.exlibrisgroup....
Alma_Users_API_Key	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Alma_Bibs_API_Key	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
EnableSendingBorrowingRequests	☑
EnableSendingHoldRequests	☑
PreferElectronicOverPrintForHoldRequests	☑
ItemSearchQueue	ALMA_RS_TEST
ItemSuccessQueue	ALMA_RS_SUCCESS
ItemFailQueue	ALMA_RS_FAILURE
ItemInExcludedLocationNeedsReviewQueue	NEEDS_REVIEW_ITEM_IN_EXCLUDE...
ItemSuccessHoldRequestQueue	ALMA_HOLD_SUCCESS
ItemFailHoldRequestQueue	ALMA_HOLD_FAILURE
NoISBNandNoOCLCNumberReviewQueue	Awaiting ISBN Lookup
ElectronicItemSuccessQueue	ALMA_ELECTRONIC_ITEM_FOUND_...
ElectronicItemReviewQueue	Electronic Item Missing URL
ILLiadFieldforElectronicItemURL	Location
FieldtoUseForUserNameFromUsersTable	UserName

The Addon is configured through the ILLiad Customization Manager by typing in values.

Settings can also be coded directly into the .xml file for redistribution.

There are currently 20 configuration fields.

Addon Workflow

for Interlibrary Loan transactions

Simplified Workflow:

From an ILLiad queue, Checks local availability via SRU Lookup using ISBN or OCLC Number for Borrowing Loan TNs

Not owned & Not available

Attempts to Submit Borrowing Request to Alma

vector icon

Route to failure queue

Route to success queue

Owned & Available

Checks for user's current loans to avoid duplicate requests

Becris

If electronic

Attempts to add URL to ILLiad transaction

ahmadwil

Route to failure queue

Route to success queue

If physical

Attempts to send Hold Request to Alma

Ariyanto Deni

Route to failure queue

Route to success queue

Step 1: Did the Addon run on this transaction already?

https://www.reddit.com/r/lotrmemes/comments/15xqcv8/were_going_in_circles/

The first step is to see if the Addon ran on the transaction already. If it has not, it will leave a note:

The ALMA BORROWING REQUEST SENDER Addon ran on this transaction.

Reason: There might be an issue with the transaction or (maybe) an issue with the Addon. If the Addon ran on the transaction already and it is still in the processing queue, the Addon will route the transaction to a failure queue so it doesn't go stale, and leave a note:

ERROR: The ALMA BORROWING REQUEST SENDER Addon already ran on this transaction and it has been sitting in the ALMA_RS_TEST processing queue. The TN is being routed to ALMA_HOLD_FAILURE. Please remove the note that says 'The ALMA BORROWING REQUEST SENDER Addon ran on this transaction.' and re-route the TN to the ALMA_RS_TEST queue in order to reprocess the TN.

ILLiad Note

Step 2: Is there an ISBN/OCLC Number, and is the ISBN valid?

https://www.instagram.com/p/Cx8OgzdvJhv/

Sometimes, users or systems leave erroneous information in the ISBN field. If there is an invalid ISBN sent to the SRU call, an error will be sent back.

If an ISBN is present in the request, the Addon will mathematically validate the ISBN.

If there is no ISBN and no OCLC Number available in the request, the Addon will route the request to a special review queue for missing identification numbers.

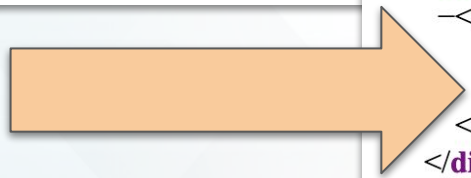
Our Case Study on ISBN Validation

On 4/29/24 10:23 AM, Persico, Angela wrote:

Hi Bill,

After talking with Tim on Friday I wondered what would happen if I put a “bad” ISBN in a request. I’ve attached the log, which shows TN 1161860, with an ISBN of “9781941250129, 19412". We have routing rules that try to catch the weird things that patrons put in this field, but wanted you to see the errors for this.

Thanks!
Angela



```
−<searchRetrieveResponse>
    <version>1.2</version>
  −<diagnostics>
    −<diag:diagnostic>
        <diag:uri>200812</diag:uri>
        <diag:message>Invalid query</diag:message>
      </diag:diagnostic>
    </diagnostics>
  </searchRetrieveResponse>
```

The Dawn of ISBNs

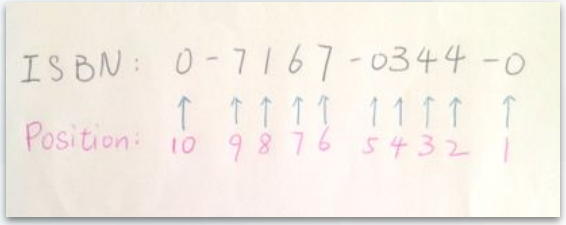
https://en.wikipedia.org/wiki/Alchemy#/media/File:Alchemist.png

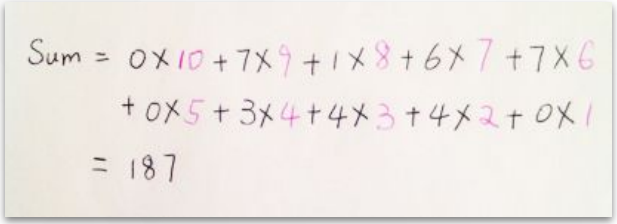
W. H. Smith (the largest single book retailer in Great Britain), the British Publishers Association's Distribution and Methods Committee and other experts in the U.K. book trade devised the Standard Book Numbering (SBN) system in 1966 and it was implemented in 1967. At the same time, the International Organization for Standardization (ISO) Technical Committee on Documentation (TC 46) set up a working party to investigate the possibility of adapting the British SBN for international use. As a result, the International Standard Book Number (ISBN) was approved as an ISO standard in 1970, and became ISO 2108 (ALA).

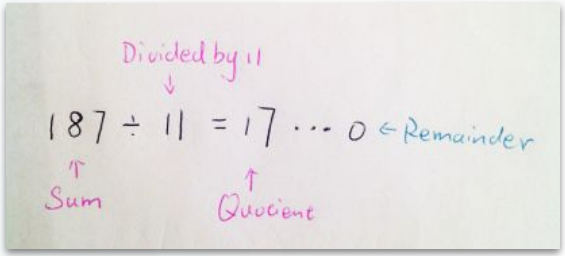
International ISBN Agency

An ISBN is an International Standard Book Number. ISBNs were 10 digits in length up to the end of December 2006, but since 1 January 2007 they now always consist of 13 digits. ISBNs are calculated using a specific mathematical formula and include a check digit to validate the number (International ISBN Agency)

What's the 10-digit math?







Starting from left to right, multiply each of the digits, starting with 10 at the first digit, and ending with 1 at the last digit. Sum these multiples, and divide by 11. If the remainder is 0, then the ISBN is valid!

https://www.instructables.com/How-to-verify-a-ISBN/

What's the 13-digit math?

```
POS =   1     2     3     4     5     6     7     8     9    10    11    12    13
--- - ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
SUM = 9×1 + 7×3 + 8×1 + 0×3 + 5×1 + 9×3 + 0×1 + 7×3 + 6×1 + 4×3 + 8×1 + 4×3 + 1x1
    =   9 +  21 +   8 +   0 +   5 +  27 +   0 +  21 +   6 +  12 +   8 +  12 +   1
    = 130
130 % 10 = 0 = VALID ISBN-13
```

Utah State University

Starting from left to right, multiply each of the digits, alternating between 1 and 3. First, start with the number 1 at the first position and continue alternating multiplying either 1 or 3 with each digit, ending with 1 at the last digit. Sum these multiples, and divide by 10. If the remainder is 0, then the ISBN is valid!

Okay so we have a good ISBN, what is next?

Having the right number is important.

```
ISBN Validator > The ISBN is 10 Digits long.
ISBN Validator > The 10ISBN sum is: 0+36+48+35+0+35+20+27+10+9=220. The remainder when 220/11 = 0
ISBN Validator > The 10 digit ISBN: 0465075959 is Valid.
```

Step 3: Is there local availability?

https://makeameme.org/meme/hows-your-availability

The Addon uses the ISBN or OCLC Number and checks for local availability with an **SRU Lookup**, with a preference over ISBN if available.

What does a URL for an SRU look like?

ISBN

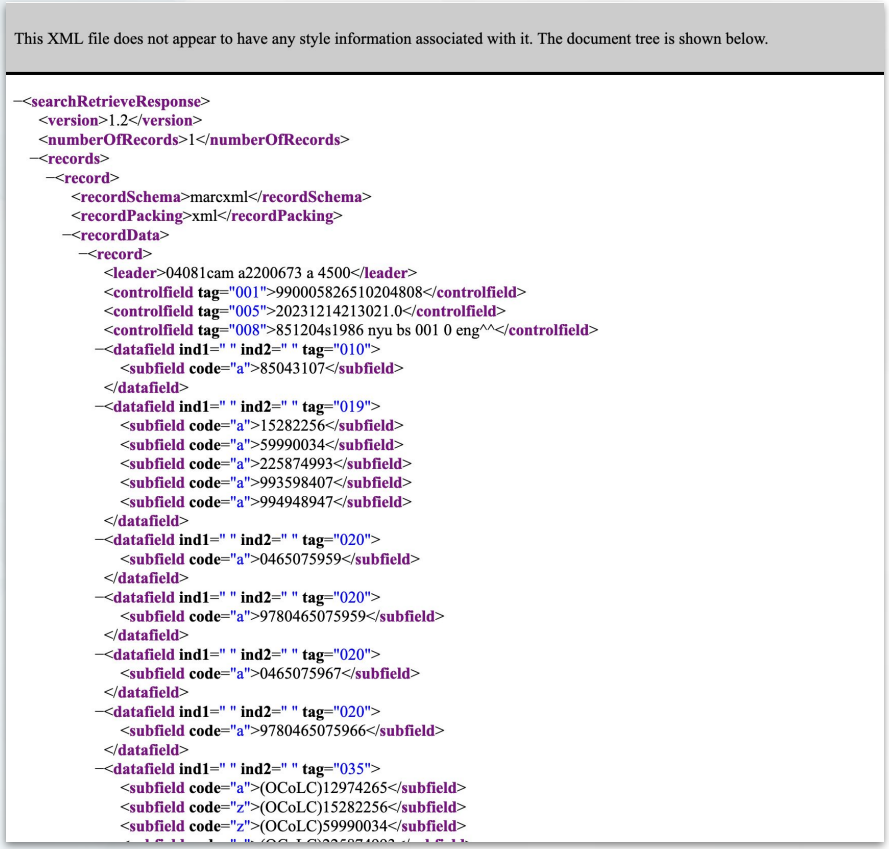
https://**suny-gen.alma.exlibrisgroup.com**/view/sru/**01SUNY_GEN**
?version=1.2&operation=searchRetrieve&recordSchema=marcxml&
query=alma.isbn=" .. **isbn** .. "&maximumRecords=3";

OCLC

https://**suny-gen.alma.exlibrisgroup.com**/view/sru/**01SUNY_GEN**
?version=1.2&operation=searchRetrieve&recordSchema=marcxml&
query=alma.oclc_control_number_035_a=" .. **oclc_number** ..
"&maximumRecords=3";

Variables: Full_Alma_URL, Alma_Institution_Code, ISBN/OCLC Number

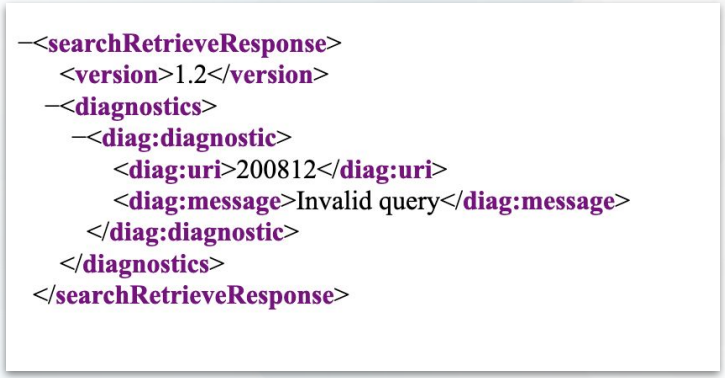
What does an SRU return look like?



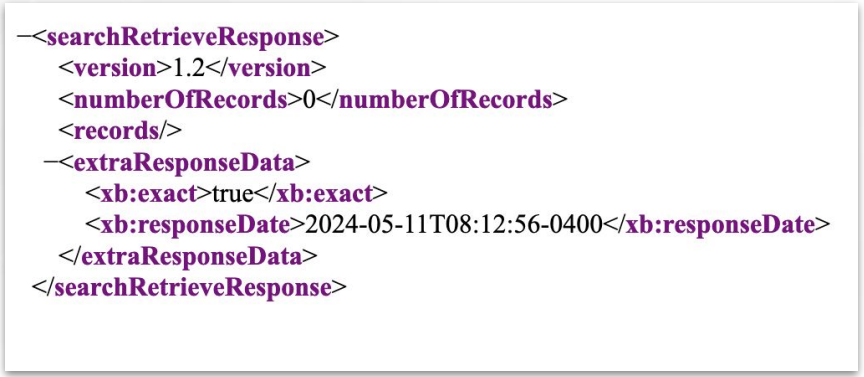
This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
−<searchRetrieveResponse>
   <version>1.2</version>
   <numberOfRecords>1</numberOfRecords>
  −<records>
   −<record>
      <recordSchema>marcxml</recordSchema>
      <recordPacking>xml</recordPacking>
     −<recordData>
      −<record>
         <leader>04081cam a2200673 a 4500</leader>
         <controlfield tag="001">990005826510204808</controlfield>
         <controlfield tag="005">20231214213021.0</controlfield>
         <controlfield tag="008">851204s1986 nyu bs 001 0 eng^^</controlfield>
        −<datafield ind1=" " ind2=" " tag="010">
            <subfield code="a">85043107</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="019">
            <subfield code="a">15282256</subfield>
            <subfield code="a">59990034</subfield>
            <subfield code="a">225874993</subfield>
            <subfield code="a">993598407</subfield>
            <subfield code="a">994948947</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="020">
            <subfield code="a">0465075959</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="020">
            <subfield code="a">9780465075959</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="020">
            <subfield code="a">0465075967</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="020">
            <subfield code="a">9780465075966</subfield>
         </datafield>
        −<datafield ind1=" " ind2=" " tag="035">
            <subfield code="a">(OCoLC)12974265</subfield>
            <subfield code="z">(OCoLC)15282256</subfield>
            <subfield code="z">(OCoLC)59990034</subfield>
```

What does it look like if an invalid ISBN is sent to an SRU lookup?



978-0744092652, 12345



0465075959abcdefghijklmnopqrstuvwxyz

978-0744092652,123451234512345123459780744092652,123451234512345123459780744092652,123451234512345123459780744092652,12345123451234512345978074409265212345123451234512345

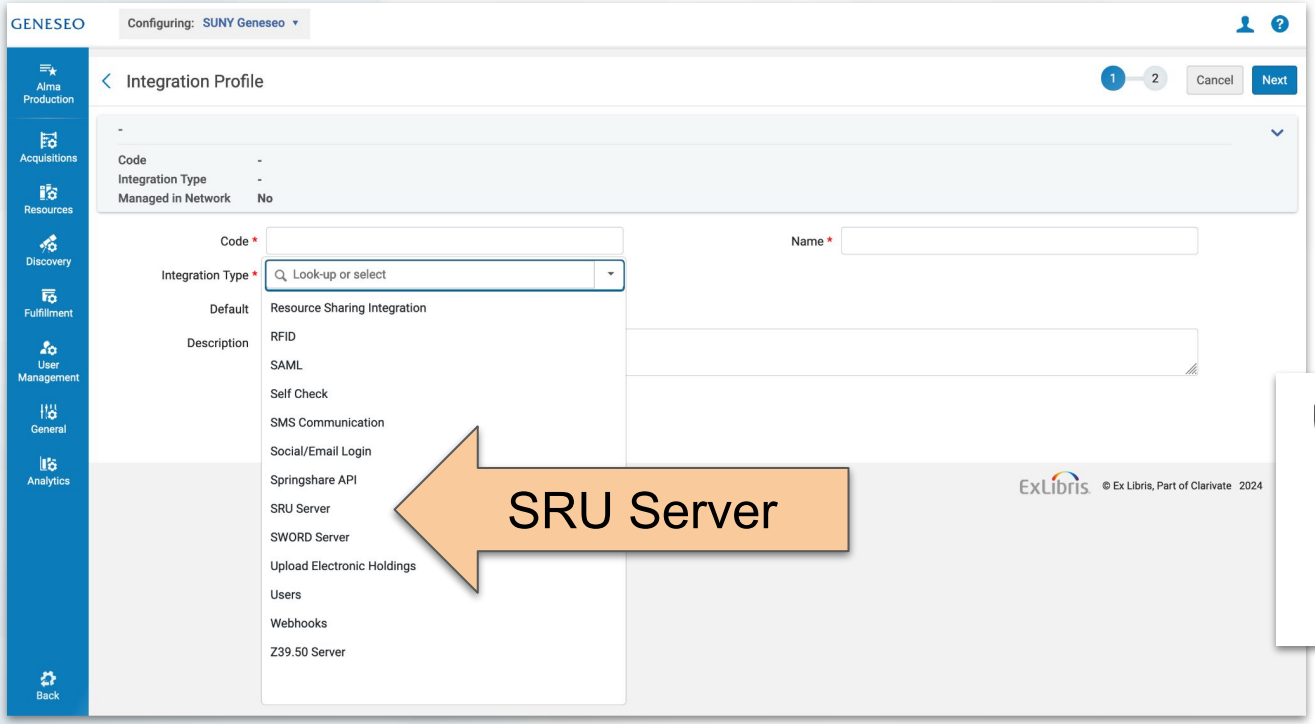
Not an error!

ISBN Tolerance: Alma don’t care (kind of)

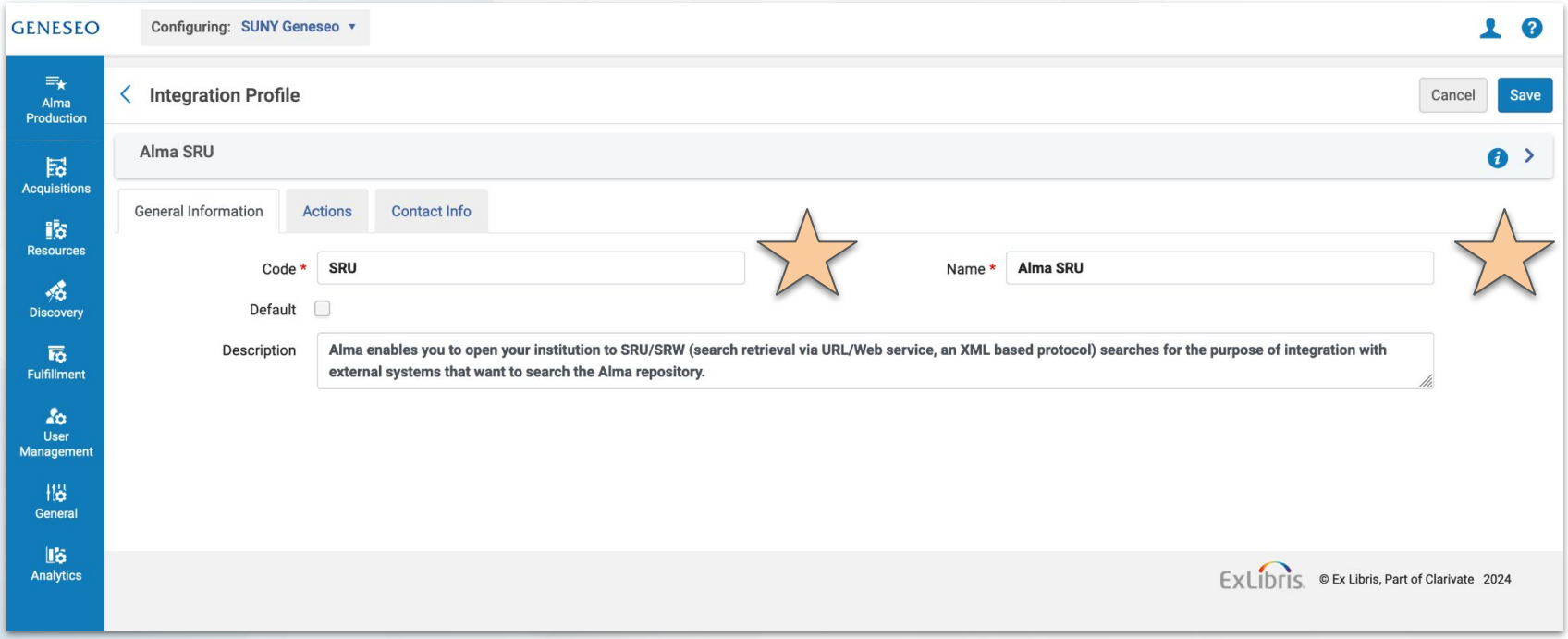
It appears that the Users API accepts almost all alphanumeric strings for an ISBN value.
The Users API will not accept malformed URLs or special characters in the ISBN value.

How **you** can enable SRU Lookup (super easy!)

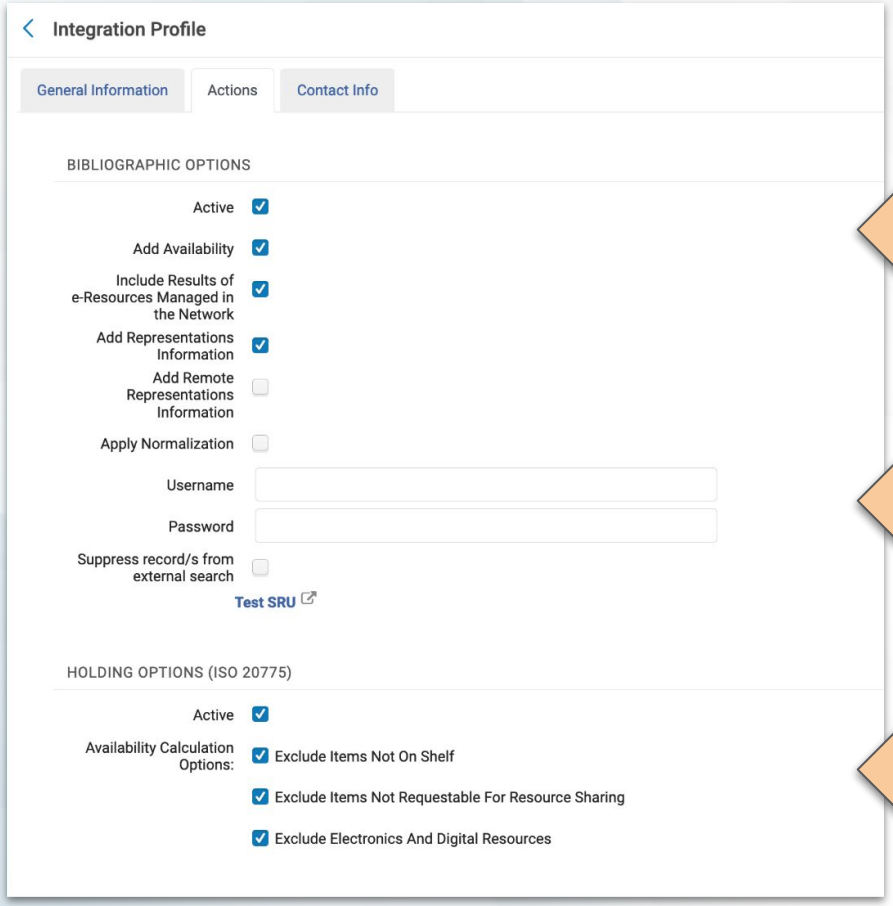
Go to Config > General > External Systems > Integration Profiles > Add Integration Profile



Give the Integration Profile a Code and a Name



Set your configuration for the SRU under ‘Actions’

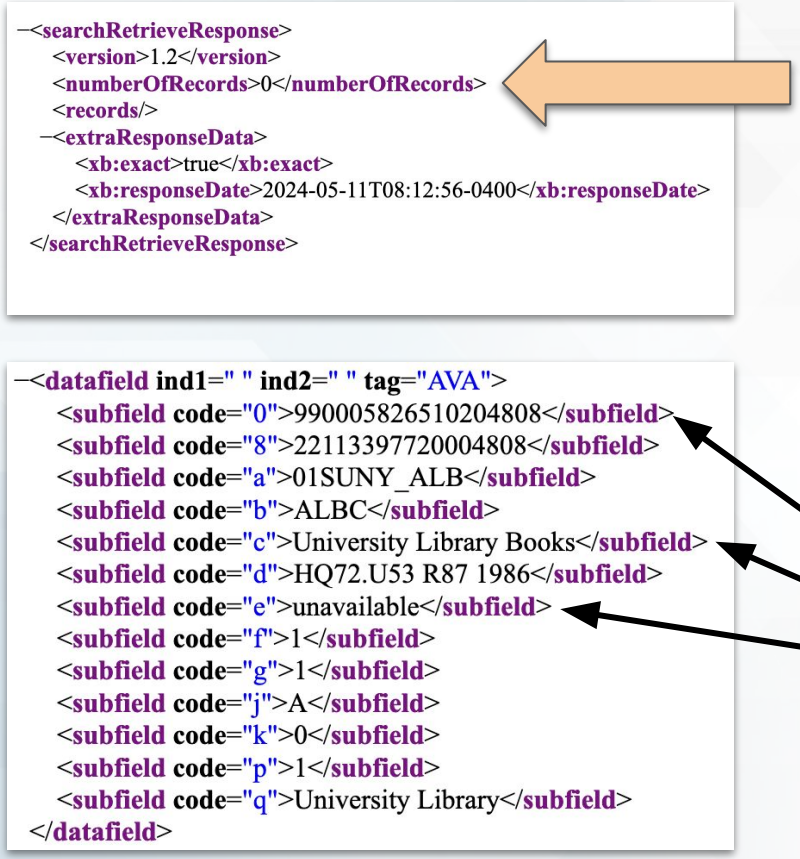


Set Bibliographic Options

Set Connection Password (optional)

Set Holdings Options

Analyzing the SRU return to determine local availability



If there is no record available, the SRU lookup will return a tag with Number of Records = 0. Otherwise, it will enter the number of records available.

There can be multiple records in a return, and a record can have more than one AVA or AVE tag with local availability information.

If the Number of Records is not equal to zero, then the Addon will gather:

- MMSID
- Shelving Location
- Availability

Determining Physical vs. Electronic Holdings from SRU

If the item is a physical item, the SRU lookup will return an “AVA” tag

If the item is an electronic item, the SRU lookup will return an “AVE” tag

AVA

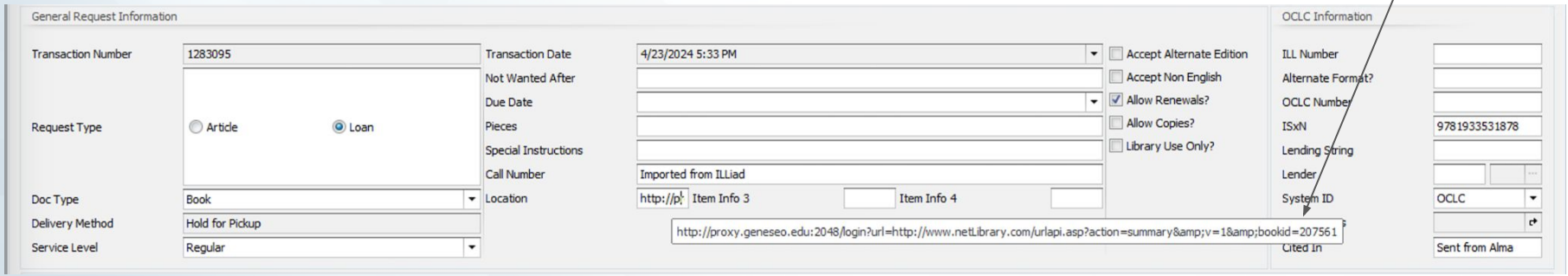
AVE

856u Magic: Adding the Full Text URL to the ILLiad TN

If there is an AVE tag in the record, the Addon will check if the item record contains an 856u tag with a URL, and add that URL to a configurable field in the ILLiad TN.

```
−<datafield ind1="4" ind2=" " tag="856">
    <subfield code="3">Bibliographic record display</subfield>
  −<subfield code="u">
      http://proxy.geneseo.edu:2048/login?url=http://www.netLibrary.com/urlapi.asp?action=summary&v=1&bookid=207561
    </subfield>
  −<subfield code="z">
      An electronic book accessible through the World Wide Web; click for information
    </subfield>
 </datafield>
```

URL



Smart Functionality: Choosing Electronic over Print

Which format do you want to deliver first?

In July 2024, we took the build_hold_request function and split out the AVA and AVE lookups into two separate functions. Then in the build_hold_request function, we created a check to see whether to run the AVE lookup first, or the AVA lookup first based on the configuration setting for PreferElectronicOverPrintForHoldRequests.

Tech-Set

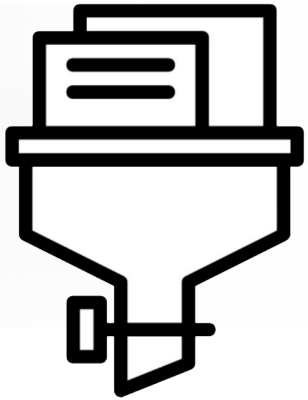
Luiz Carvalho

Step 4: Filtering through the results

For each record available in the SRU return, The Addon will look at AVA/AVE tag(s), and if the availability is equal to “unavailable” or “Unavailable”, then it will determine that record to be unusable.

For available items, the Addon will look at the Shelving Location of the item, and if the shelving location is listed on the ‘excluded_locations.txt’ document in the Addon, then that record will be unusable.

If the record contains an AVA tag that says the item is Available and the item is in a permitted location (according to the excluded locations list), the record will be determined usable and the MMSID will be used to send a Hold Request.

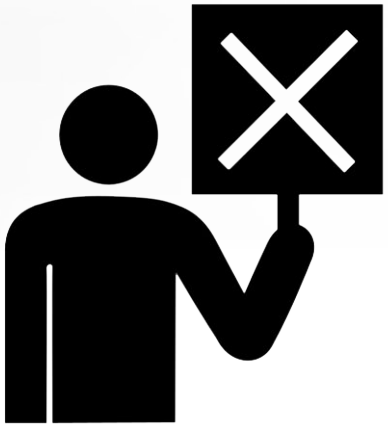


The excluded_locations.txt file

ibrandify

If the item location is on the excluded_locations list, a note will be left with the name of the excluded location, and the transaction will be routed to a review queue for items that belong in an excluded location. A note will be left in the transaction with the excluded location name.

This excluded_locations file works for both physical and electronic items that come back in an SRU Lookup.


Hayashi Fumihiro

Message from check_excluder function: The shelving location [Schrader Hall Storage] is on the Exclude list.
The location is on the exclude list. Routing to Review Queue.

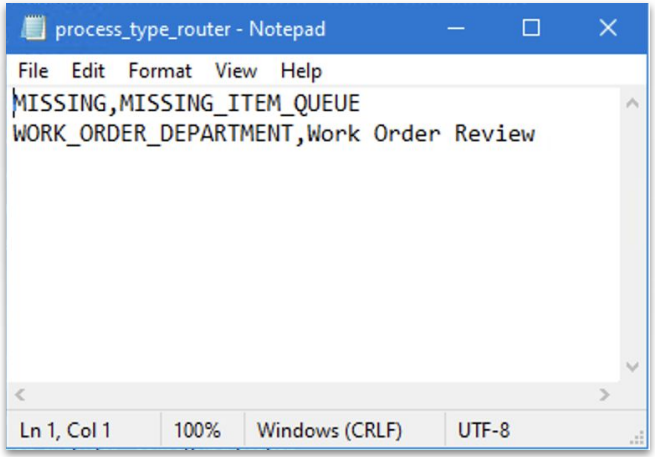
ILLiad Note

The process_type_router.txt file

If the item availability during the SRU Lookup comes back as owned and not available, a separate API call will be made using the Alma Bibs API to determine if the item is in a specific process_type. This information is held in the <process_type> tag in the API return.

If a <process_type> value exists, the Addon will check the process_type_router.txt to see if the value is listed, and the Addon will route the TN to the queue specified in the process_type_router.txt document.

If the <process_type> is equal to “ILL”, then the Addon will leave a note in the TN and attempt to send a Borrowing request to Alma:



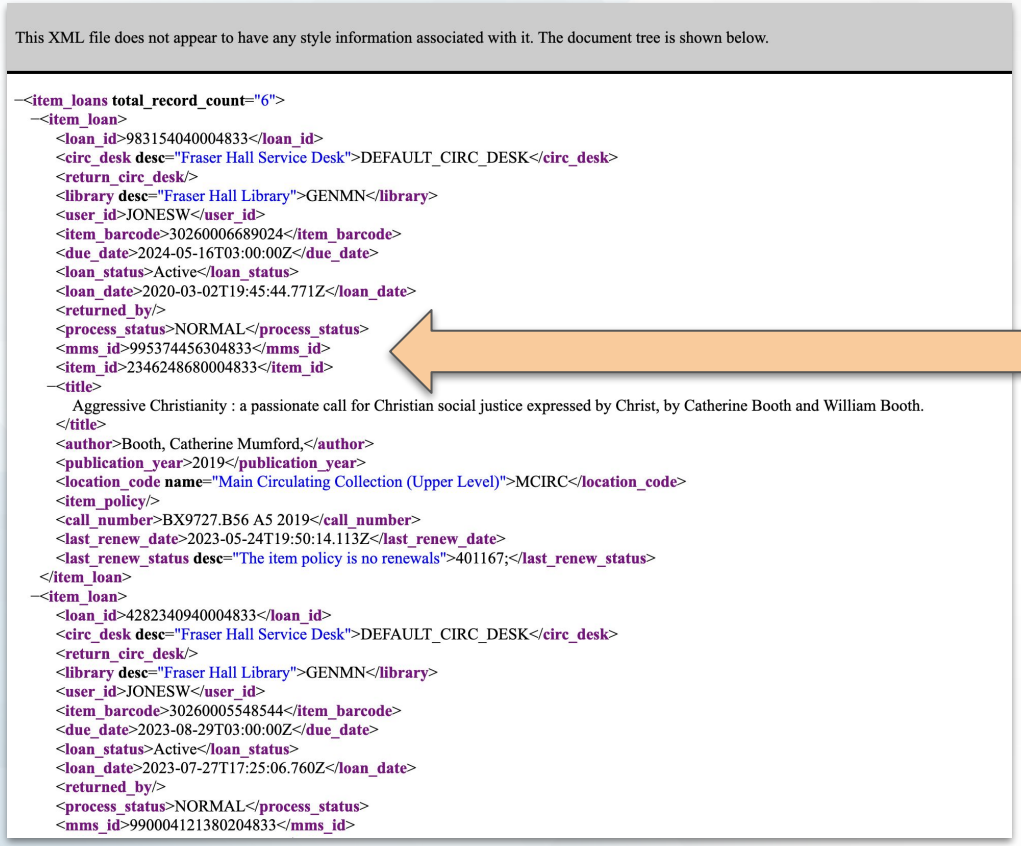
6/14/2024 12:43 PM	The ALMA BORROWING REQUEST SENDER Addon ran on this transaction.	System
6/14/2024 12:43 PM	From ALMA BORROWING REQUEST SENDER: Item is currently on Loan through Resource Sharing	System
6/14/2024 12:43 PM	Alma API Response for Alma Borrowing Request Sender received successfully	System

Repeat shopper? Catching duplicate requests

If the item is available and if the item is in a permissible location, the Addon will check to see if the user has active loans on the MMSID by using the Alma Users API ‘Retrieve user loans’ call:

https://**api-na.hosted.exlibrisgroup.com**/almaws/v1/users/**USERNAME**/loans?user_id_type=all_unique&limit=10&offset=0&order_by=id&direction=ASC&loan_status=Active&apikey=**YOUR_USERS_API_KEY**

Here is a sample Loans API return



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
−<item_loans total_record_count="6">
  −<item_loan>
      <loan_id>983154040004833</loan_id>
      <circ_desk desc="Fraser Hall Service Desk">DEFAULT_CIRC_DESK</circ_desk>
      <return_circ_desk/>
      <library desc="Fraser Hall Library">GENMN</library>
      <user_id>JONESW</user_id>
      <item_barcode>30260006689024</item_barcode>
      <due_date>2024-05-16T03:00:00Z</due_date>
      <loan_status>Active</loan_status>
      <loan_date>2020-03-02T19:45:44.771Z</loan_date>
      <returned_by/>
      <process_status>NORMAL</process_status>
      <mms_id>995374456304833</mms_id>
      <item_id>2346248680004833</item_id>
    −<title>
        Aggressive Christianity : a passionate call for Christian social justice expressed by Christ, by Catherine Booth and William Booth.
      </title>
      <author>Booth, Catherine Mumford,</author>
      <publication_year>2019</publication_year>
      <location_code name="Main Circulating Collection (Upper Level)">MCIRC</location_code>
      <item_policy/>
      <call_number>BX9727.B56 A5 2019</call_number>
      <last_renew_date>2023-05-24T19:50:14.113Z</last_renew_date>
      <last_renew_status desc="The item policy is no renewals">401167;</last_renew_status>
  </item_loan>
  −<item_loan>
      <loan_id>4282340940004833</loan_id>
      <circ_desk desc="Fraser Hall Service Desk">DEFAULT_CIRC_DESK</circ_desk>
      <return_circ_desk/>
      <library desc="Fraser Hall Library">GENMN</library>
      <user_id>JONESW</user_id>
      <item_barcode>30260005548544</item_barcode>
      <due_date>2023-08-29T03:00:00Z</due_date>
      <loan_status>Active</loan_status>
      <loan_date>2023-07-27T17:25:06.760Z</loan_date>
      <returned_by/>
      <process_status>NORMAL</process_status>
      <mms_id>99000412138020483</mms_id>
```

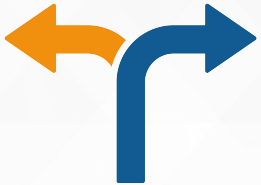
Here's the MMSID!

If an MMSID in the User's Loans matches the MMSID from the SRU call, the transaction will be routed to a Failure queue.

At this point, there are two options:

There's nobody home

If there isn't a local item available for the ISBN or OCLC Number provided in the request, the Addon will attempt to submit a **Borrowing Request** to Alma, if the Library has turned on this feature in the Addon.



Come and get me!

If the item is locally available for check out, AND the user does not have active loans on the MMSID, then the MMSID is gathered from the SRU and used to send a **Hold Request** for the user to Alma.

Placing Hold Requests through the Alma Users API

Let's Draw

How do you place a Hold Request in Alma using a web API call?

You use the Alma Users API and send a 'Create User Request' call

Ex Libris Developer Network
https://developers.exlibrisgroup.com › ... › REST APIs

Users and Fulfillment

The **Users API** allows access to **users** related information, as described in the map below. Alma provides a set of Web services for handling **user** information, ...

https://developers.exlibrisgroup.com/alma/apis/users/

Structure of the Users API ‘Create Request’

You establish a connection to the API with the URL structure:

https://**api-na.hosted.exlibrisgroup.com/almaws/v1**/users/ **USERNAME** /requests?user_id_type=all_unique&mms_id=**THE_ITEM_MMSID**&allow_same_request=false&apikey=**YOUR_USERS_API_KEY**

And then you send a small XML message to the URL:

```xml
<user_request>
    <request_type>HOLD</request_type>
    <description></description>
    <pickup_location_type>LIBRARY</pickup_location_type>
    <pickup_location_library>GENMN</pickup_location_library>
    <pickup_location_circulation_desk></pickup_location_circulation_desk>
    <pickup_location_institution>01SUNY_GEN</pickup_location_institution>
    <target_destination></target_destination>
    <material_type></material_type>
</user_request>
```

Building the XML Message for the Hold Request

```xml
<user_request>
  <request_type>HOLD</request_type>
  <description></description>
  <pickup_location_type>LIBRARY</pickup_location_type>
  <pickup_location_library>GENMN</pickup_location_library>
  <pickup_location_circulation_desk></pickup_location_circulation_desk>
  <pickup_location_institution>01SUNY_GEN</pickup_location_institution>
  <target_destination></target_destination>
  <material_type></material_type>
</user_request>
```

Pickup Location Type will be ‘LIBRARY’ unless the Pickup Location is ‘Home Delivery’ or ‘Office Delivery’.

Pickup Location Library: sublibraries.txt crosswalk

Pickup Location Institution: Addon config for Alma_Institution_Code

How does the Addon send the Hold Request?

The Addon creates the URL by getting the username from the transaction, and the MMSID from the SRU call, and the Users API Key and Alma URL base stored in the Addon's config.

https://**api-na.hosted.exlibrisgroup.com/almaws/v1**/users/**USERNAME**/requests?user_id_type=all_unique&mms_id=**THE_ITEM_MMSID**&allow_same_request=false&apikey=**YOUR_USERS_API_KEY**

The Addon uses a WebClient to establish a connection to the Users API, and then it sends the assembled XML message from the last slide.

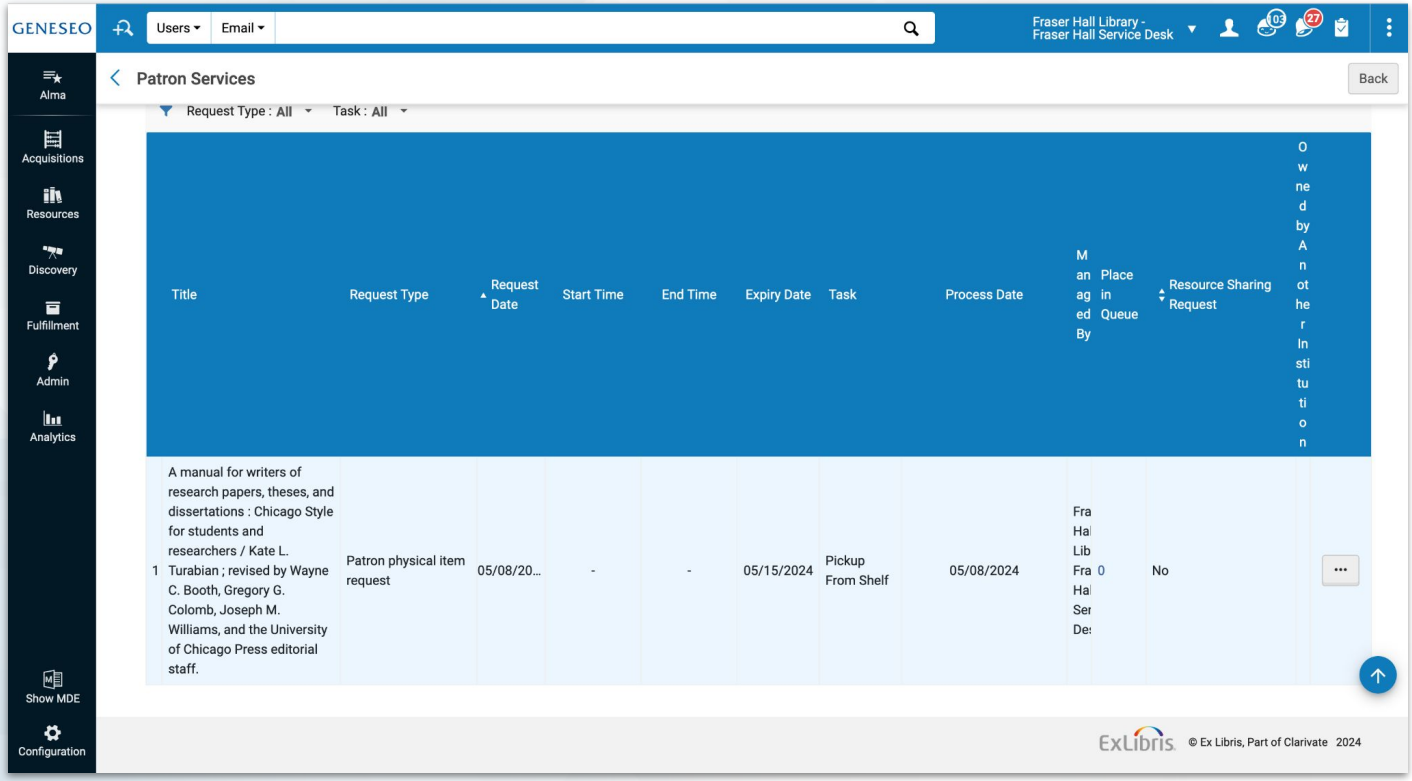
```
LogDebug("Creating web client for Alma HOLD message.");
local webClient = Types["WebClient"]();
webClient.Headers:Clear();
webClient.Headers:Add("Content-Type", "application/xml; charset=UTF-8");
webClient.Headers:Add("accept", "application/xml; charset=UTF-8");
LogDebug("Sending Hold Message to Alma Users API.");

local responseString = webClient:UploadString(alma_url, hold_message);
```

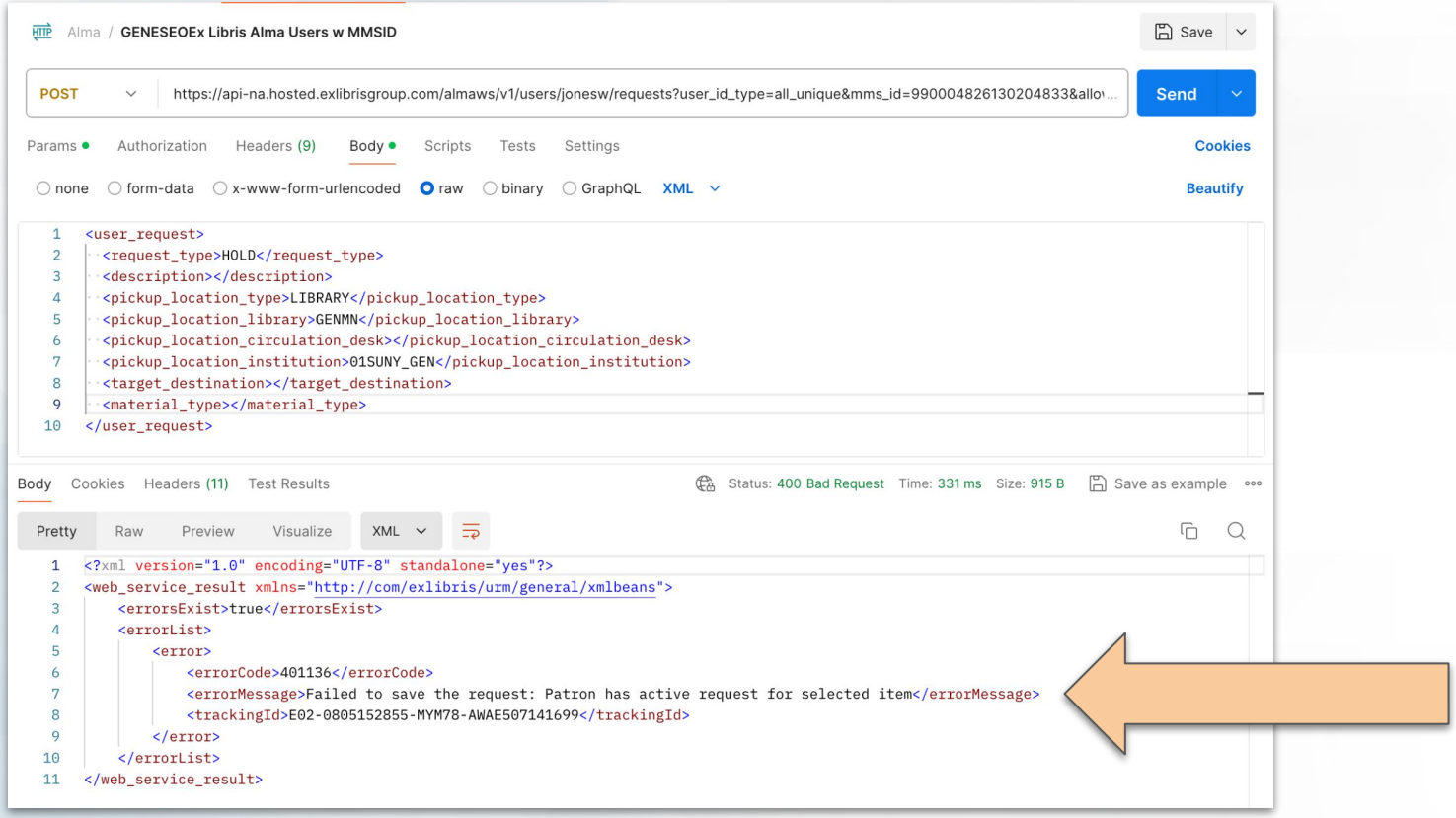
What does a successful Hold Request look like from the API?

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<user_request>
    <user_primary_id>JONESW</user_primary_id>
    <request_id>4811222300004833</request_id>
    <additional_id>481-122-230</additional_id>
    <request_type>HOLD</request_type>
    <request_sub_type desc="Patron physical item request">PATRON_PHYSICAL</request_sub_type>
    <mms_id>990004826130204833</mms_id>
    <title>A manual for writers of research papers, theses, and dissertations : Chicago Style for students and researchers / Kate L. Turabian ; revised by
        Wayne C. Booth, Gregory G. Colomb, Joseph M. Williams, and the University of Chicago Press editorial staff.</title>
    <author>Turabian, Kate L., author.</author>
    <description></description>
    <holding_id>2287506190004833</holding_id>
    <pickup_location>Fraser Hall Library</pickup_location>
    <pickup_location_type>LIBRARY</pickup_location_type>
    <pickup_location_library>GENMN</pickup_location_library>
    <managed_by_library>Fraser Hall Library</managed_by_library>
    <managed_by_circulation_desk>Fraser Hall Service Desk</managed_by_circulation_desk>
    <managed_by_library_code>GENMN</managed_by_library_code>
    <managed_by_circulation_desk_code>DEFAULT_CIRC_DESK</managed_by_circulation_desk_code>
    <material_type/>
    <volume></volume>
    <issue></issue>
    <part></part>
    <date_of_publication></date_of_publication>
    <request_status>In Process</request_status>
    <request_date>2024-05-08Z</request_date>
    <request_time>2024-05-08T15:26:15.178Z</request_time>
    <task_name>Pickup From Shelf</task_name>
    <expiry_date>2024-05-16Z</expiry_date>
</user_request>
```

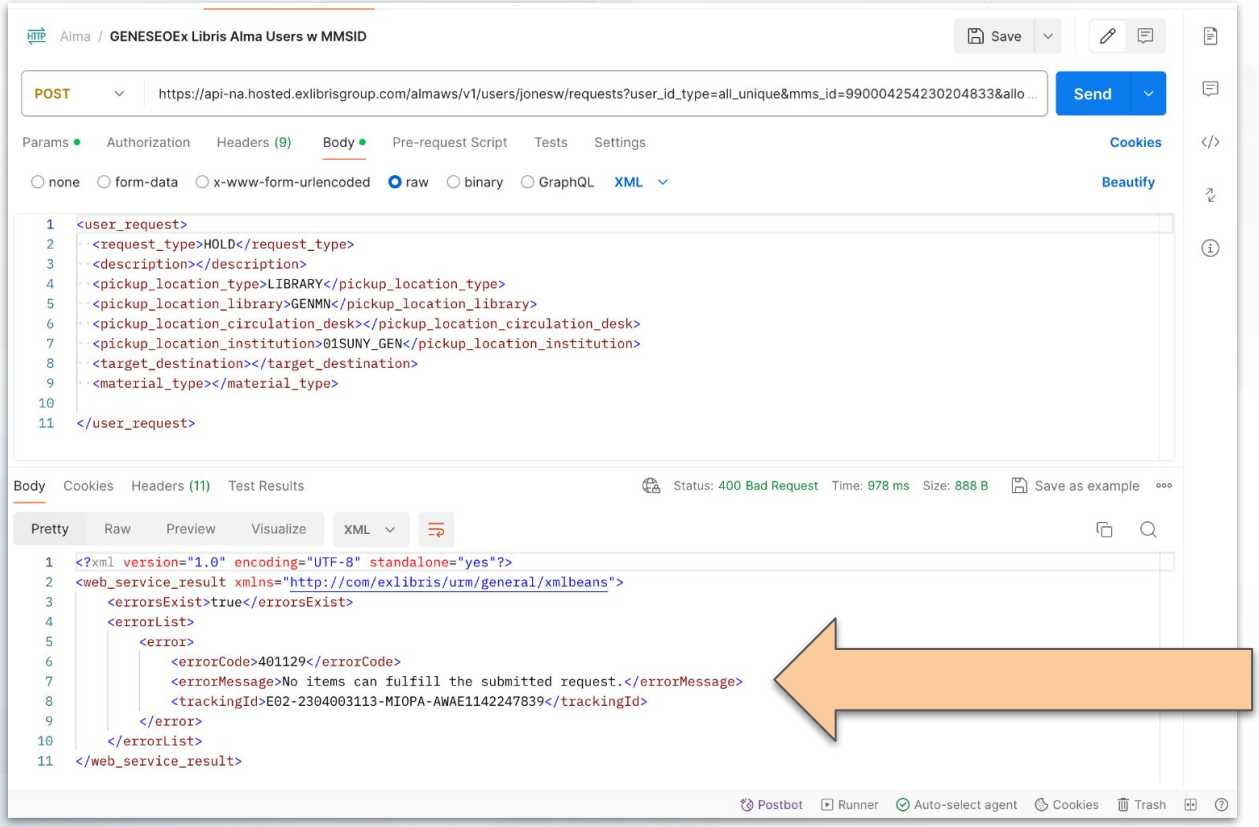
What does a successful Hold Request look like in Alma?



What does a failed Hold Request look like (in Postman)?



What happens if you send a **hold request** for an electronic item?



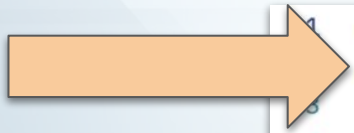
No items can fulfill the submitted request

Handling a Successful Hold Request

If the Hold Request in Alma is successful, the API will return an XML message, and inside that message will be the tag ‘<user_request>’.

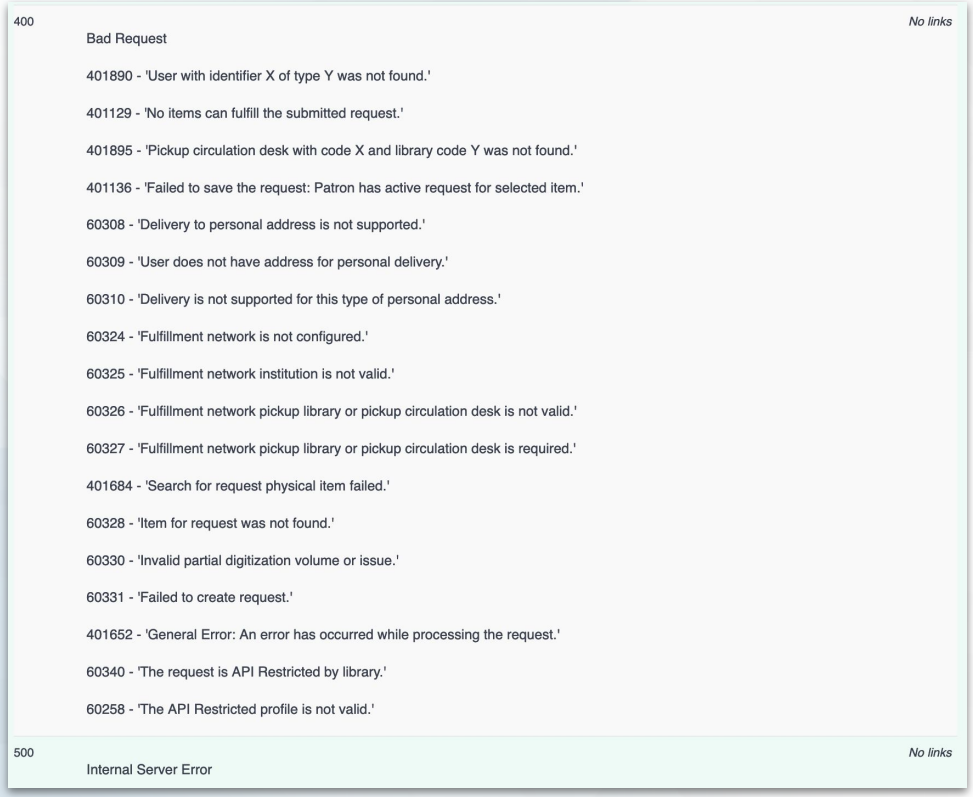
If that tag is present in the API return, the Addon will route the transaction to a configurable ‘Item Hold Success’ queue.

If that ‘<user_request>’ tag is not present in the API return, that means that there was some type of error and the error needs to be handled appropriately.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<user_request>
    <user_primary_id>JONESW</user_primary_id>
    <request_id>4811222300004833</request_id>
    <additional_id>481-122-230</additional_id>
    <request_type>HOLD</request_type>
    <request_sub_type desc="Patron physical item request">PATRON_PHYSICAL</request_sub_type>
    <mms_id>990004826130204833</mms_id>
```

Handling a Failed Hold Request



There are multiple reasons that a failure can occur in a Hold Request.

If there is an API error returned, the Addon treats it as an error and it uses an error handler to catch the specific error.

You need to know the behavior to make effective adjustments

It’s really important to know what the error is in order to handle it appropriately!

In order to get the error, you need to first capture the InnerException.message

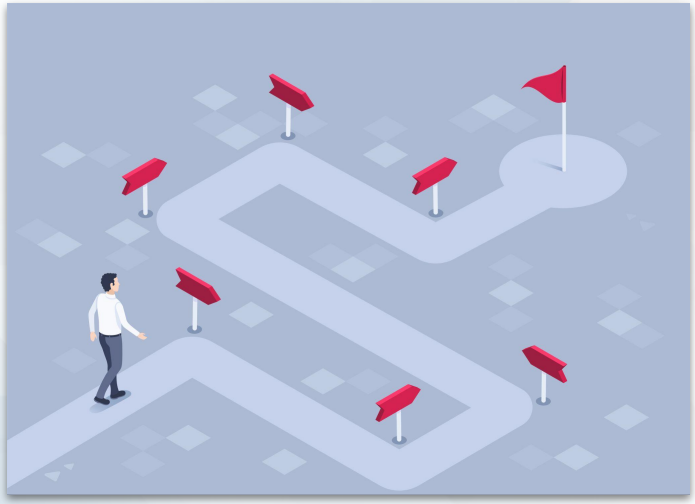
Next you need to create a streamreader and read the error message in entirety.

In the Alma API error message are the XML tags:

- <errorCode>
- <errorMessage>

These are extremely helpful in determining why things don’t go as anticipated!

Behavior Management for Hold Request API Errors



The Addon automatically routes a TN to a configurable ItemFailHoldRequestQueue if the error code is 401136:

Failed to save the request: Patron has active request for selected item.

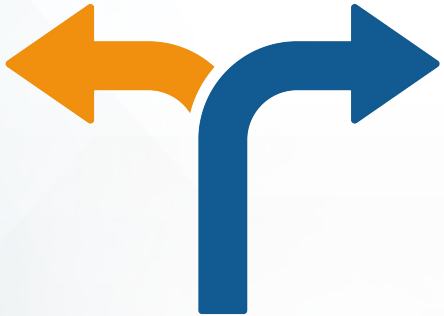
A special file is included in the Addon titled ‘error_routing’ that allows staff to create a comma separated list of error codes and ILLiad queue names in order to route to specific queues for specific API error codes

So what happens next?

If the Hold Request fails, there are two possible routes:

If Submitting Borrowing Requests is **enabled**, the Addon will attempt to submit a Borrowing Request

If Submitting Borrowing Requests is **disabled**, the transaction will be routed to the configurable Hold Failure queue



Placing Borrowing Requests through the Alma Users API

Let's Draw

How do you place a Borrowing Request in Alma using a web API call?

Use the Alma Users API and send a 'Create user request for resource sharing' call

Ex Libris Developer Network
https://developers.exlibrisgroup.com › ... › REST APIs

Users and Fulfillment

The **Users API** allows access to **users** related information, as described in the map below. Alma provides a set of Web services for handling **user** information, ...

https://developers.exlibrisgroup.com/alma/apis/users/

Structure of the Users API ‘Create user request for resource sharing’

You establish a connection to the API with the URL structure:

https://**api-na.hosted.exlibrisgroup.com**/almaws/v1/users/ **USERNAME** /resource-sharing-requests?user_id_type=all_unique&override_blocks=true &apikey=**YOUR_USERS_API_KEY**

And then you send an XML message to the URL:

```xml
<user_resource_sharing_request>
<format desc="string">
<xml_value>PHYSICAL</xml_value>
</format>
<pickup_location_type>LIBRARY</pickup_location_type>
<pickup_location desc="string">
<xml_value>GENMN</xml_value>
</pickup_location>
<citation_type desc="string">
<xml_value>BK</xml_value>
</citation_type>
<title>The best title you've ever seen</title>
</user_resource_sharing_request>
```

This is the most basic XML message that we could send to the API and receive a successful response.

```
local ml = '<?xml version="1.0" encoding="ISO-8859-1"?><user_resource_sharing_request>';
    ml = ml .. '<format desc="string">';
    ml = ml .. '<xml_value>PHYSICAL</xml_value>';
    ml = ml .. '</format>';
    ml = ml .. '<pickup_location_type>' .. pickup_location_type ..
    '</pickup_location_type>';
    ml = ml .. '<pickup_location desc="string">';
    ml = ml .. '<xml_value>' .. pickup_location_library .. '</xml_value>';
    ml = ml .. '</pickup_location>';
    ml = ml .. '<citation_type desc="string">';
    ml = ml .. '<xml_value>BK</xml_value>';
    ml = ml .. '</citation_type>';
    if usernotes == "" then
    ml = ml .. '<note>Request created from ILLiad TN: ' .. transactionNumber_int ..
    '</note>';
    end
    if usernotes ~= "" then
    ml = ml .. '<note>Request created from ILLiad TN: ' .. transactionNumber_int .. '.
    Note from Patron: ' .. usernotes .. '</note>';
    end
    if loan_title ~= nil then
    loan_title = cleanup_field(loan_title);
    ml = ml .. '<title>' .. loan_title .. '</title>';
    end
    if isbn ~= nil then
    ml = ml .. '<isbn>' .. isbn .. '</isbn>';
    end
    if loan_author ~= nil then
    loan_author = cleanup_field(loan_author);
    ml = ml .. '<author>' .. loan_author .. '</author>';
    end
    if loan_date ~= nil then
    ml = ml .. '<year>' .. loan_date .. '</year>';
    end
    if loan_publisher ~= nil then
    loan_publisher = cleanup_field(loan_publisher);
    ml = ml .. '<publisher>' .. loan_publisher .. '</publisher>';
    end
    if loan_place ~= nil then
    loan_place = cleanup_field(loan_place);
    ml = ml .. '<place_of_publication>' .. loan_place .. '</place_of_publication>';
    end
    if loan_edition ~= nil then
    ml = ml .. '<edition>' .. loan_edition .. '</edition>';
    end
    ml = ml .. '<call_number>Imported from ILLiad</call_number>';
    if oclc_number ~= nil then
    ml = ml .. '<oclc_number>' .. oclc_number .. '</oclc_number>';
    end
    ml = ml .. '</user_resource_sharing_request>';
```

Building the XML Message for the Borrowing Request

- Pickup Location Type will be ‘LIBRARY’ unless the Pickup Location is ‘Home Delivery’ or ‘Office Delivery’.
- Pickup Location Library: sublibraries.txt crosswalk
- Notes are sent from the original ILLiad transaction
- Bibliographic information is cleaned for troublesome characters before sent

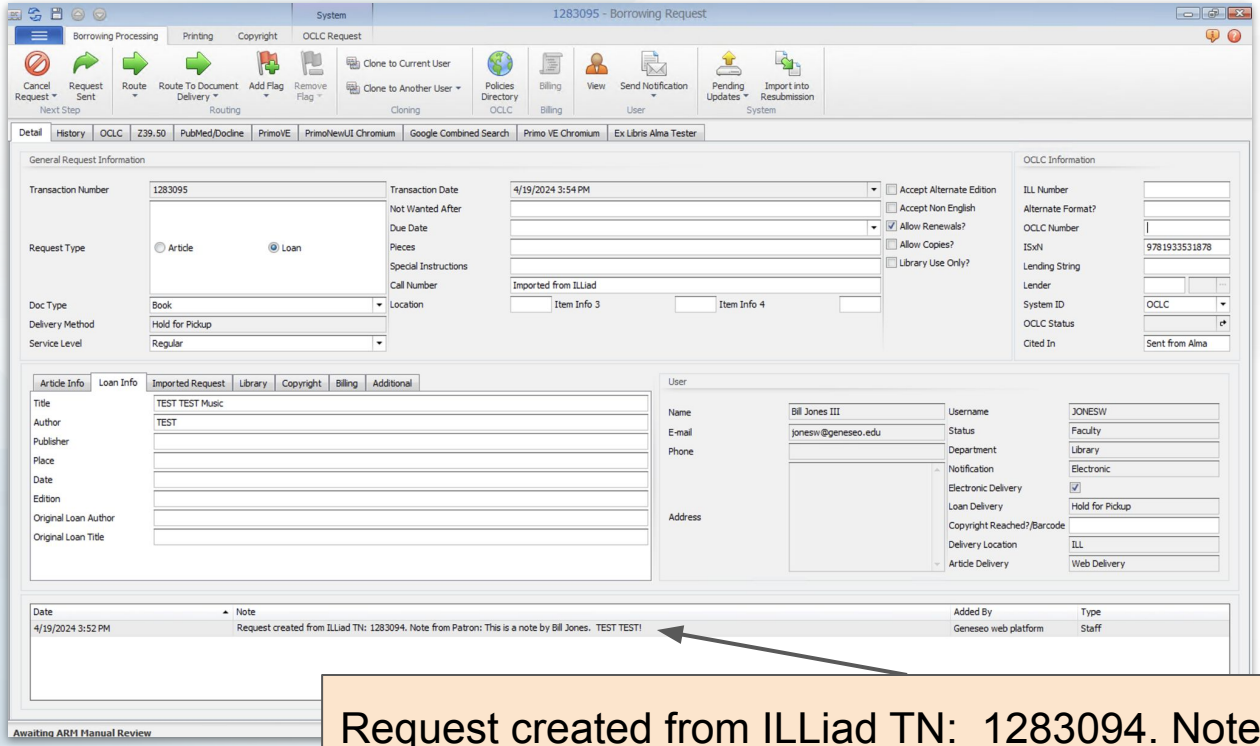
Leaving love notes behind

What if your patron left behind a note in the ILLiad request?

Let's copy that note to the Alma Borrowing request.

Is the request coming back to ILLiad through Alma GES? So is the note!

Note with a Note!

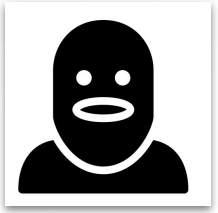


Request created from ILLiad TN: 1283094. Note from Patron: This is a note by Bill Jones. TEST TEST!

The Note has a note left from the patron, and a note left by ILLiad when the borrowing request was sent to Alma, with the original ILLiad Transaction Number.

When the request was not able to be filled in the Alma rotas, and went to back to ILLiad (as GES from Alma), it carried both the note from the patron and the note from ILLiad.

Look out for those shifty characters!



NounProject

XML escape characters

There are only five:

"	"
'	'
<	<
>	>
&	&

https://stackoverflow.com/questions/1091945/what-characters-do-i-need-to-escape-in-xml-documents

If these characters are sent in the XML message, it will return an error. A special function was built to escape these characters for any text field that might be sent in the XML message:

```
function cleanup_field(title)

-- " = &quot;
-- ' = &apos;
-- < = &lt;
-- > = &gt;
-- & = &amp;

local cleaned_string = title;

cleaned_string = cleaned_string:gsub('&', '&amp;'):gsub('"', '&quot;'):gsub("'", '&apos;'):gsub('<', '&lt;'):gsub('>', '&gt;');

return cleaned_string;
end
```

The Great Escape: Handling Unicode Characters

At one point during our testing, Angela came across titles with diacritics. These produced a failure in the Borrowing Request API call.

The solution to handling this issue was placing a tag at the beginning of the XML message: ‘<?xml version="1.0" encoding="ISO-8859-1"?>’

Adding these headers for the Addon’s webClient was also necessary:

>> webClient.Headers:Add("Content-Type", "application/xml; **charset=UTF-8**");

>> webClient.Headers:Add("accept", "application/xml; **charset=UTF-8"**);

Handling a Successful Borrowing Request

If the Borrowing Request in Alma is successful, the API will return an XML message, and inside that message will be the tag ‘<user_resource_sharing_request>’.

If that tag is present in the API return, the Addon will route the transaction to a configurable ‘Item Success’ queue.

If that ‘<user_resource_sharing_request>’ tag is not present in the API return, that means that there was some type of error and the error needs to be handled appropriately.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<user_resource_sharing_request>
    <request_id>4630602910004833</request_id>
    <external_id>01SUNYGEN0022168</external_id>
    <created_date>2024-02-23Z</created_date>
    <last_modified_date>2024-02-23Z</last_modified_date>
    <created_time>2024-02-23T18:09:13.781Z</created_time>
    <last_modified_time>2024-02-23T18:09:15.633Z</last_modified_time>
    <status desc="Locate in process">LOCATE_IN_PROCESS</status>
    <owner>GENMN</owner>
    <partner desc="Nazareth University" link="https://api-na.hosted.exlibrisgroup.com/almaws/v1/partners/Nazareth College">Nazareth
```

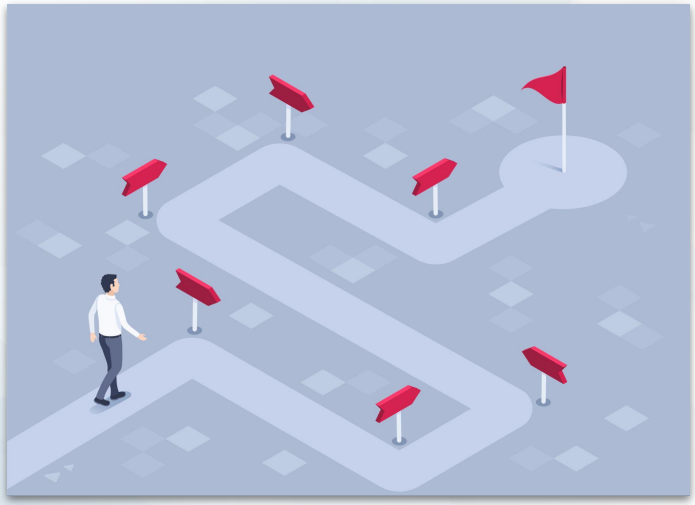
Handling a Failed Borrowing Request

400	Bad Request 401890 - 'User with identifier X of type Y was not found.' 401604 - 'Warning: The institutional inventory has services for the requested title.' 401652 - 'General Error: An error has occurred while processing the request.' 402039 - 'Could not create request, default item location is not defined for the resource sharing library' 401768 - 'Patron is not affiliated with a resource sharing library' 402362 - 'Failed to save the request: Patron has duplicate request' 401607 - 'Resource sharing library (owner) is missing' 401608 - 'The given resource sharing library (owner) is not defined in the patron record' 60340 - 'The request is API Restricted by library.' 60258 - 'The API Restricted profile is not valid.'
500	Internal Server Error

There are multiple reasons that a failure can occur in a Borrowing Request.

If there is an API error returned, the Addon treats it as an error and it uses an error handler.

Behavior Management for Borrowing Request API Errors



The Addon automatically routes a TN to a configurable ItemFailQueue if the error code is 401873:

Failed {0}

We wrote a custom note for this error that says:

Patron has duplicate Borrowing Request in Alma

A special file is included in the Addon titled ‘error_routing’ that allows staff to create a comma separated list of error codes and ILLiad queue names in order to route to specific queues for specific API error codes. If the error is not caught by the error_routing document, the transaction will be routed to the configurable ItemFailQueue status.

Future Plans

Continue Testing the Addon at:

University at Albany

Begin Testing the Addon at:

Binghamton University

University at Buffalo

Upstate Medical University

Tools

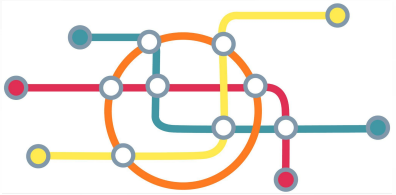
[Postman]

[Notepad++]

[Baretail]

Impact at the University at Albany

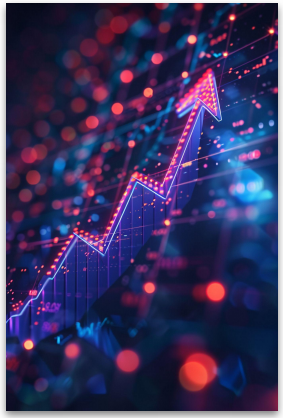
Workflow



Using 2 instances of the addon

- Holds enabled/Borrowing disabled
  - All loan requests with ISBN/OCLC number sent to Awaiting Request Processing
  - “Holds” addon monitors queue and checks for local availability and active requests
  - If owned and available: hold request placed
  - If active request exists: routed to review
  - If unavailable/not owned: request sent to ARM (aka OCLC Direct Request)
- Borrowing enabled/Holds disabled
  - ARM automation checks for match in ‘Alma Partners’ Custom Holdings Group and routes to Awaiting Arm Review
  - Custom routing rule routes request to queue monitored by addon
  - “Borrowing” addon sends request out to Alma Borrowing
- If unfilled in Alma, request is recreated in ILLiad and sent to ARM
  - IDS Logic rule bypasses ARM Alma Partners automation

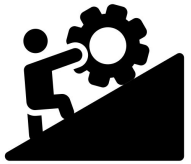
Impact at UAlbany



- Between 3/1/24 and 7/17/24
  - 581 loan requests received in ILLiad
    - 251 filled by lenders in OCLC (i.e., not cancelled, placed on hold, owned as ebook, purchased, etc.)
  - Addon sent 181 requests (31.2%) to Alma
    - 125 borrowing requests
      - 104 requests (83.2%) filled in Alma
    - 56 hold requests (92 since 12/1/23)

- Greatly reduces staff processing time
  - Hold creation process is fully automatic if patron provides a valid ISBN/OCLC number
  - Alma Borrowing processing is easier/faster/fewer steps than ILLiad

Challenges


Maryati

- Can’t reactivate original ILLiad request if Alma request goes unfilled
- Need two versions of the addon to minimize request duplication in ILLiad
- Addon cannot create personal delivery borrowing requests because the method the addon uses to create borrowing requests does not accommodate personal delivery
- Holds can be placed on items that users can’t request via Primo (Alma config issue)
- Requests for multivolume works are a challenge
- Need to update custom holdings group for new Alma partners as we expand our Alma partnerships

Sharing the Addon

- Addon files available in GitHub repository
  - https://github.com/Hypolymer/Alma_Borrowing_Request_Sender/

- Plan to add to ILLiad addons directory once a few more institutions have tested the addon

- SUNY Libraries: submit a ticket to SUNY Library Services if you'd like to implement this addon at your library
  - info@sunyolis.libanswers.com

IDS PROJECT Conference 2024

Libraries Without Borders: Connecting Collections

Questions?

Thank you!

Timothy Jackson, Timothy.Jackson@suny.edu

Bill Jones, jonesw@geneseo.edu

Angela Persico, apersico@albany.edu