Optimizing Digital Resource Sharing Workflows with OCLC APIs
Matt Niehoff
Applications Developer, Minitex

Stacy Brunner
Senior Product Analyst, OCLC
Digital Resource Sharing and OCLC APIs

Matt Niehoff - Applications Developer
nieho003@umn.edu

Minitex
What is Minitex?

- A state-funded library organization located at the University of Minnesota Twin Cities.
- Programs:
  - Database access to all Minnesotans (https://elibrarymn.org/)
  - Online Reference for all Minnesotans
  - Cooperative Purchasing
  - Support for Digital Collections
  - Training & Education
  - and...
Resource Sharing
Minitex Borrowing/Lending in OCLC

Borrowing

- 8th on OCLC with 20,845 requests (MII).

Lending

- 2nd on OCLC with 34,882 requests as Minitex (MII).
- Additional 21,190 requests as the University of Minnesota (MNU).

External Integrations

Additional (Current) External Systems

- Alma (ExLibris)
- MNLINK (VDX)
- SHAREit (Auto-Graphics)
- ILLiad
- Direct Requests via Minitex Website
Situation

Minitex 2015

- Large Volume of ILL Requests across multiple platforms
- A paper-based process
Situation

- [https://xkcd.com/927/](https://xkcd.com/927/)
Mira - A Digital Request Management System

- Standardize request format
- Automatic request sorting
- Point-of-need printing
- Audit trail for all requests
- Automatic workflow statistics
- Fully (almost) electronic workflow
- Opportunities for integration
Mira - A Digital Request Management System

- WPF Desktop Application
  - Click-Once Deployment
  - Permissions with Active Directory
  - DevExpress UI and MVVM Libraries
- SQL Server backend
  - Entity Framework
  - Managed by University OIT
- Windows Server
  - Scheduled .NET Console Applications for Auto-Download/Auto-Verification
  - Document OCR with pypdfocr ([https://pypi.org/project/pypdfocr/](https://pypi.org/project/pypdfocr/))
Digital Minitex Requests

August 4th, 2015 - May 25th, 2021 14:57

- 1,739,923 requests* across 7 platforms and 13 symbols
- 2019 (last “normal” year)
  - 319,647 requests
  - ~ 1,076 per working day

* Not all ILLiad requests are imported to Mira
Mira - A Digital Request Management System

- Standardize request format
- Automatic request sorting
- Point-of-need printing
- Audit trail for all requests
- Automatic workflow statistics
- Fully (almost) electronic workflow
- Opportunities for integration
Lifecycle of a Minitex Request
Minitex Resource Sharing Workflow Overview

- Incoming Requests
- Verification
- Fulfillment
- Delivery
Optimize with OCLC APIs
3 Ways Mira Uses OCLC APIs

- WorldCat Search
  - Enhancing incoming request metadata
  - Automatically locating owning partners
- Article Exchange
  - Integrated delivery of electronic resources
Enhancing Incoming Metadata
Limited Bib Information

Title: Are you afraid of the dark?

Bibliographic Information:

- Title: Are you afraid of the dark?
- Author: 
- Description: 
- Publisher: 
- Edition: 
- Series: 
- Standard Numbers: OCLC - 1048049972;
Limited Bib Information

Borrower Notes

Bibliographic Information

Title: Are you afraid of the dark?

Standard Numbers: OCLC - 1048049972;
WorldCat Search API - Bib Read

https://platform.worldcat.org/api-explorer/apis/wcapi/Bib/Read
<record xmlns="http://www.loc.gov/MARC21/slim">
  <leader>00000cgm a2200000Ma 4500</leader>
  <controlfield tag="001">1048049972</controlfield>
  <controlfield tag="008">140820s2014 nyu323 vleng</controlfield>
  <datafield ind1="0" ind2="0" tag="245">
    <subfield code="a">Are you afraid of the dark?</subfield>
    <subfield code="n">Season 1</subfield>
    <subfield code="c">Nickelodeon ; directed by D.J. MacHale [and others].</subfield>
  </datafield>
  <datafield ind1="3" ind2="" tag="246">
    <subfield code="a">Are you afraid of the dark?</subfield>
    <subfield code="n">The first year</subfield>
  </datafield>
  <datafield ind1="" ind2="1" tag="264">
    <subfield code="a">[United States] :</subfield>
    <subfield code="b">Viacom,</subfield>
    <subfield code="c">©2014.</subfield>
  </datafield>
  <datafield ind1="" ind2="" tag="300">
    <subfield code="a">2 videodiscs (approximately 323 min.) :</subfield>
    <subfield code="b">sound, color ;</subfield>
    <subfield code="c">4 3/4 in.</subfield>
  </datafield>
</record>
Title: Are you afraid of the dark?

Description: 2 videodiscs (approximately 323 min.) : sound, color ; 4 3/4 in.


Standard Numbers: OCLC - 1048049972;
• Extract more information without overriding original data and getting correct formats

• On-demand in-app querying
Auto-Verification
WorldCat Search - Auto-Verification

- Use ISBN, ISSN, UPC to get an OCLC Number

- Query locations from record with most matching holdings

- Filter against partner libraries
StringBuilder url = new StringBuilder();
url.Append("http://www.worldcat.org/webservices/catalog/content/libraries/");
switch (QueryType)
{
    case WCSearchLocationsQueryType.Isbn:
        url.Append("isbn/");
        break;
    case WCSearchLocationsQueryType.Issn:
        url.Append("issn/");
        break;
    case WCSearchLocationsQueryType.Oclc:
        url.Append("/");
        break;
    case WCSearchLocationsQueryType.StandardNumber:
        url.Append("sn/");
        break;
}
url.Append(QueryIdentifyingNumber);
url.Append("?wskey=");
url.Append(wskey);

if (Json)
{
    url.Append("&format=json");
}
if (ServiceLevel == "full")
{
    url.Append("&servicelevel=full");
}
if (FrbrResultSet == false)
{
    url.Append("&frbrGrouping=off");
}
if (!String.IsNullOrEmpty(this.State))
{
    url.Append("&location="); url.Append(this.State);
}
return url.ToString();
StringBuilder url = new StringBuilder();
url.Append("http://www.worldcat.org/webservices/catalog/content/libraries/");
switch (QueryType)
{
    case WCSearchLocationsQueryType.Isbn:
        url.Append("isbn/");
        break;
    case WCSearchLocationsQueryType.Issn:
        url.Append("issn/");
        break;
    case WCSearchLocationsQueryType.Oclc:
        url.Append("/");
        break;
    case WCSearchLocationsQueryType.StandardNumber:
        url.Append("sn/");
        break;
}
url.Append(QueryIdentifyingNumber);
url.Append("?wskey=");
url.Append(wskey);

if (Json)
{
    url.Append("&format=json");
}
if (ServiceLevel == "full")
{
    url.Append("&servicelevel=full");
}
if (FrbrResultSet == false)
{
    url.Append("&frbrGrouping=off");
}
if (!String.IsNullOrEmpty(this.State))
{
    url.Append("&location="); url.Append(this.State);
}
return url.ToString();
http://www.worldcat.org/webservices/catalog/content/libraries/isbn/9791220046886?wskey={wskey}&format=json&servicelevel=full&frbrGrouping=off
http://www.worldcat.org/webservices/catalog/content/libraries/isbn/9791220046886?wskey={wskey}&format=json&servicelevel=full&frbrGrouping=off

{
    "title": "Interesting state",
    "author": "Wunderlich, Elsie. Manzo, Marco. Pieralice, Stefania, editor.",
    "publisher": "Start",
    "date": "[2019]",
    "ISBN": ["9791220046886"],
    "OCLCnumber": "1108918806"
}
Build Query URL

http://www.worldcat.org/webservices/catalog/content/libraries/isbn/9791220046886
?wskey={wskey}&format=json&servicelevel=full&frbrGrouping=off

http://www.worldcat.org/webservices/catalog/content/libraries/1108918806?wskey=wskey&format=json&startLibrary=1&maximumLibraries=50&servicelevel=full&frbrGrouping=off

{
  "title": "Interesting state",
  "author": "Wunderlich, Elsie. Manzo, Marco. Pieralice, Stefania, editor.",
  "publisher": "Start",
  "date": "[2019]",
  "ISBN": ["9791220046886"],
  "OCLCnumber": "1108918806"
}
http://www.worldcat.org/webservices/catalog/content/libraries/isbn/9791220046886?wskey={wskey}&format=json&servicelevel=full&frbrGrouping=off

{
   "title": "Interesting state",
   "author": "Wunderlich, Elsie. Manzo, Marco. Pieralice, Stefania, editor. ",
   "publisher": "Start",
   "date": "[2019]",
   "ISBN": ["9791220046886"],
   "OCLCnumber": "1108918806",
   "totalLibCount": 11,
   "library": [
      {
         "institutionName": "K.O. Lee Aberdeen Public Library",
         "streetAddress1": "215 4th Ave SE",
         "streetAddress2": "",
         "city": "Aberdeen",
         "state": "SD",
         "oclcSymbol": "AML"
      }
   ]
}

http://www.worldcat.org/webservices/catalog/content/libraries/1108918806?wskey=wskey&format=json&startLibrary=1&maximumLibraries=50&servicelevel=full&frbrGrouping=off
Locations in Mira
Article Exchange
Article Exchange

- Directly post to Article Exchange from Scanning Application
- Ability to retrieve files by URL/Password
Social media and vaccine hesitancy: new updates for the era of COVID-19 and globalized infectious diseases
Article Exchange - API

Documentation

https://www.oclc.org/developer/develop/web-services/article-exchange-api.html

Upload uses a Multipart HTTP POST method

https://ill.sd00.worldcat.org/articleexchange/?autho=[ID]&password=[password]
internal string UploadFile(string uploadFile) // C:/..../request1234.pdf
{
    // Get content type from file extension
    string contentType = base.GetContentType(Path.GetExtension(uploadFile));

    // Define the multipart boundary string
    string boundaryDef = String.Format("---HTTPCLIENT-{0:x}", DateTime.Now.Ticks);
    string boundaryItem = String.Format("--{0}", boundaryDef);
    string finalBoundary = String.Format("{0}--", boundaryItem);
    byte[] finalBoundaryBytes = Encoding.UTF8.GetBytes(String.Format("\r\n{0}\r\n", GetFinalBoundary));

    // Set request headers
    HttpWebRequest webrequest = (HttpWebRequest)WebRequest.Create(Endpoint);
    webrequest.ContentType = String.Format("multipart/form-data; boundary={0}", boundaryDef);
    AttachWSKeyV2(webrequest, wskey);

    // Build post header and convert to a byte array to write into the request stream
    StringBuilder postHeader = BuildPostHeader();
    byte[] postHeaderBytes = Encoding.UTF8.GetBytes(postHeader.ToString());
// Create FileStream
FileStream fileStream = new FileStream(uploadFilePath, FileMode.Open, FileAccess.Read);

// Set content length and open request stream
Stream requestStream = webrequest.GetRequestStream();
// Write post header
requestStream.Write(postHeaderBytes, 0, postHeaderBytes.Length);

// Write file to the request stream
byte[] buffer = new Byte[8192];
int bytesRead = 0;
while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) != 0)
{
    requestStream.Write(buffer, 0, bytesRead);
}

// Write trailing boundary
requestStream.Write(finalBoundaryBytes, 0, finalBoundaryBytes.Length);

// Close open streams
fileStream.Close();
requestStream.Close();
// Post the File
WebResponse response = webrequest.GetResponse();

// Parse the response
Stream s = response.GetResponseStream();
StreamReader sr = new StreamReader(s, Encoding.UTF8);
string output = sr.ReadToEnd();
response.Close();
sr.Close();
return output;
private void Process()
{
    string temp = this.rawResponse;
    temp = temp.Replace(" xmlns="", ", ");

    MemoryStream xmlStream = new MemoryStream(Encoding.UTF8.GetBytes(temp));

    XmlSerializer xmlSerializer = new XmlSerializer(typeof(AEResponseXml), "http://www.w3.org/2005/Atom");

    AEResponseXml x = (AEResponseXml)xmlSerializer.Deserialize(xmlStream);

    this._Url = x.content.uploadResponse.accessInformationResponse.url;
    this._Password = x.content.uploadResponse.accessInformationResponse.password;
}
Article Exchange - Delivery

- Post Scanned or Saved File via Article Exchange API

- Use URL/Password combination to update in external system or send e-mail notification directly from Mira

- Manage electronic delivery pipeline without use of external scanning applications/websites
In Review

- Minitex has built a digital ILL request management system.

- Digital Requests allow better tracking and external integration opportunities.

- OCLC APIs help automate and enhance Resource Sharing workflows.
Thank You!

Matt Niehoff - Applications Developer
nieho003@umn.edu
Optimizing Digital Resource Sharing Workflows with OCLC APIs
Ways to optimize resource sharing

**ISO 18626**
Standard protocol for communicating ILL requests.

**Resource Sharing**
APIs specifically for taking actions on requests in the OCLC resource sharing network.

**Article Exchange**
Deliver documents securely between patron and supplying and requesting libraries.

**Policies Directory**
Information regarding OCLC suppliers, group memberships, and individual library polices related to ILL.

**Smart Fulfillment**
Aggregate service to determine fulfillment.
ISO 18626 Protocol

- Actually not an API.
- While requests may look different from system to system, the protocol provides a way of communicating request details in a standard way.
- Pilot testing between RelaisILL and WorldShare ILL/Tipasa wraps up this month.
- Register your WorldShare ILL symbol to send/receive ISO 18626 requests this summer. More information coming soon, or email oclcresourcesharing@oclc.org
ISO 18626 Protocol: Build, connect

• The Circ/ILL Toolkit is a library for marshalling, unmarshalling, and exchanging messages conforming to ISO ILL (ISO 18626) or NCIP (NISO Z39.83).
  – Use the toolkit (oc.lc/circill-toolkit) to build your own ISO requesting. Submit requests to or receive requests from the OCLC network.

• Most partners that previously implemented the Resource Sharing SOAP service will now use ISO 18626 instead.

• **Ideal for** managing requests between disparate networks or ILL applications and transferring requests between systems.
Resource Sharing APIs

- Built to support the WMS/Tipasa My Account interface.
  - Create Request
  - Read Request
  - Search Requests
- Coming soon to OCLC Developer’s Network – release date TBD
- Will replace the former SOAP-based API described in the Minitex workflow.
Resource Sharing APIs: Build

Ideal for...

• Patron request forms or systematically creating requests directly on the OCLC network.
• Getting request status for use outside ILL applications.
• Showing patrons their circ and ILL requests in one place.
Planning to add more functionality in future, like support for a mobile ILL application. Would implement:

- Get Queue
- Update Request
Article Exchange API

- Upload and retrieve shared documents.
- Extended recently when scanner integration added for Tipasa. Offers the option to mark a request shipped upon adding document to Article Exchange.
- Minitex use described in Matt’s presentation
- **Ideal for** securely delivering an article that can be shared with a library or patron, especially when document provided from a non-ILL system.
Policies Directory API

• Provides institution data such as fees, hours, closures, contact information, supplier status
• Used today by ILLiad.
• Ideal for assessing available suppliers for a request.
Smart Fulfillment API

- User Research happening now. API coming soon.
- Determine fulfillment based on
  - Who’s asking (patron, faculty, consortia member, ILL)
  - What’s the item (location, format)
  - What’s the policy (loan duration, in library use)
  - What’s the delivery expectation (turnaround time)
- Ideal for providing a fulfillment option wherever it’s needed.
Resource Sharing SOAP API

- Older API, not available on OCLC Developer Network. New access NOT available.
- Replace with
  - ISO 18626 protocol
  - Resource Sharing APIs
Ideas for resource sharing APIs

• Submit ideas and enhancements to WorldShare ILL or Tipasa OCLC Community Center
• Request a one-on-one meeting to discuss your library’s development needs: oclcresourcesharing@oclc.org
Because what is known must be shared.
thank you