'This Just In' - Alma Analytics API, Acquisitions, and the Quest for Visibility

Giulio Menna | g.menna@library.leidenuniv.nl | IGeLU 2022

Universiteit
Leiden
The Netherlands

Who am I?

- Giulio Menna, Developer / Functional Application Manager at Leiden University Libraries

- In practice: "I make things, and make sure things work"

Who are we?

- Six full-time developers in three different departments

- Six other colleagues that are free to develop solutions for their issues

- More than last year!

- Proudly sharing our finished products as Open Source on GitHub:
  https://github.com/LeidenUniversityLibrary

What are we talking about today?

- "Acquisitions - New books at the library" - https://acquisitions.library.universiteitleiden.nl/

- Open Source version: https://github.com/LeidenUniversityLibrary/Alma-Acquisitions-Open-Source

- Google 'Acquisitions Leiden University Libraries' to follow along!

- We are NOT going deep into the code

- We are going to give a look at the development process, the possibilities offered by ExLibris' APIs, issues, and workarounds

- Please feel free to contact me after the presentation!

Context: The problems we tried to solve

The app should answer the question:

"How do we let users know what the library has recently acquired?"

Problems we needed to solve:

1. Difficult-to-maintain codebase (Perl) of the existing application

2. Convoluted way of adding new acquisitions lists

   a) Create and RSS in Alma

   b) Manually add a TXT file on the server to let the application know there is a new acquisitions list

3. Application mostly unknown to the library users

The solutions we applied

1. Difficult-to-maintain codebase (Perl)

   - Convert the application to Laravel (PHP)

2. Convoluted way of adding new acquisitions lists

   - Add an admin panel to add new acquisitions lists (no interacting with the server needed)

3. Application mostly unknown to the library users

   - Let search engines and users know this application exists

   - More linking to the application itself

   - Newsletters

   - Trainings for students

How does it work?

A well-known approach in Alma:

1. Create an report with recent acquisitions in Alma Analytics

2. Fetch and display the results in the application via the Alma Analytics API

3. Links will lead to Primo (VE)

Sources:

[A webpage to retrieve new acquisitions using Alma Analytics API - Ex Libris Developer Network (exlibrisgroup.com)](#)

[How to use an API to retrieve an Alma Analytics report in 5 easy steps - Ex Libris Developer Network (exlibrisgroup.com)](#)

Create a report Alma Analytics

- Alma -> Analytics -> Design Analytics

- Create new analysis

- Bibliographic details:

  - Title

  - Author

  - Publisher

  - Publication Date

  - MMS Id (required to generate a link to Primo in the application)

  - Resource Type (allows users to find electronic book, journals, paper books, etc.)

  - Creation Date (The date an item was created in Alma. It allows to sort by 'latest acquisitions')

  - Subjects (allows for searching by subject if the user desires so)

  - LC classifications (Library of Congress Classification as requested by colleagues)

Create report in Alma Analytics

- Filters:
  - "Bibliographic Details"."Creation Date" >= TIMESTAMPADD(SQL_TSI_MONTH, -3, CURRENT_DATE)
  - AND MMS Id is not equal to any MMS Id in Acquisitions Process Types
  - AND Resource Type is equal to / is in Book - Electronic; Book - Physical; Book chapter - Electronic; Journal - Electronic; Journal – Physical
  - AND Bibliographic Lifecycle is equal to / is in In Repository
  - AND Availability is equal to / is in Available

  Check the results

  Save in a dedicated 'Acquisitions' folder in Alma Analytics

Create report in Alma Analytics
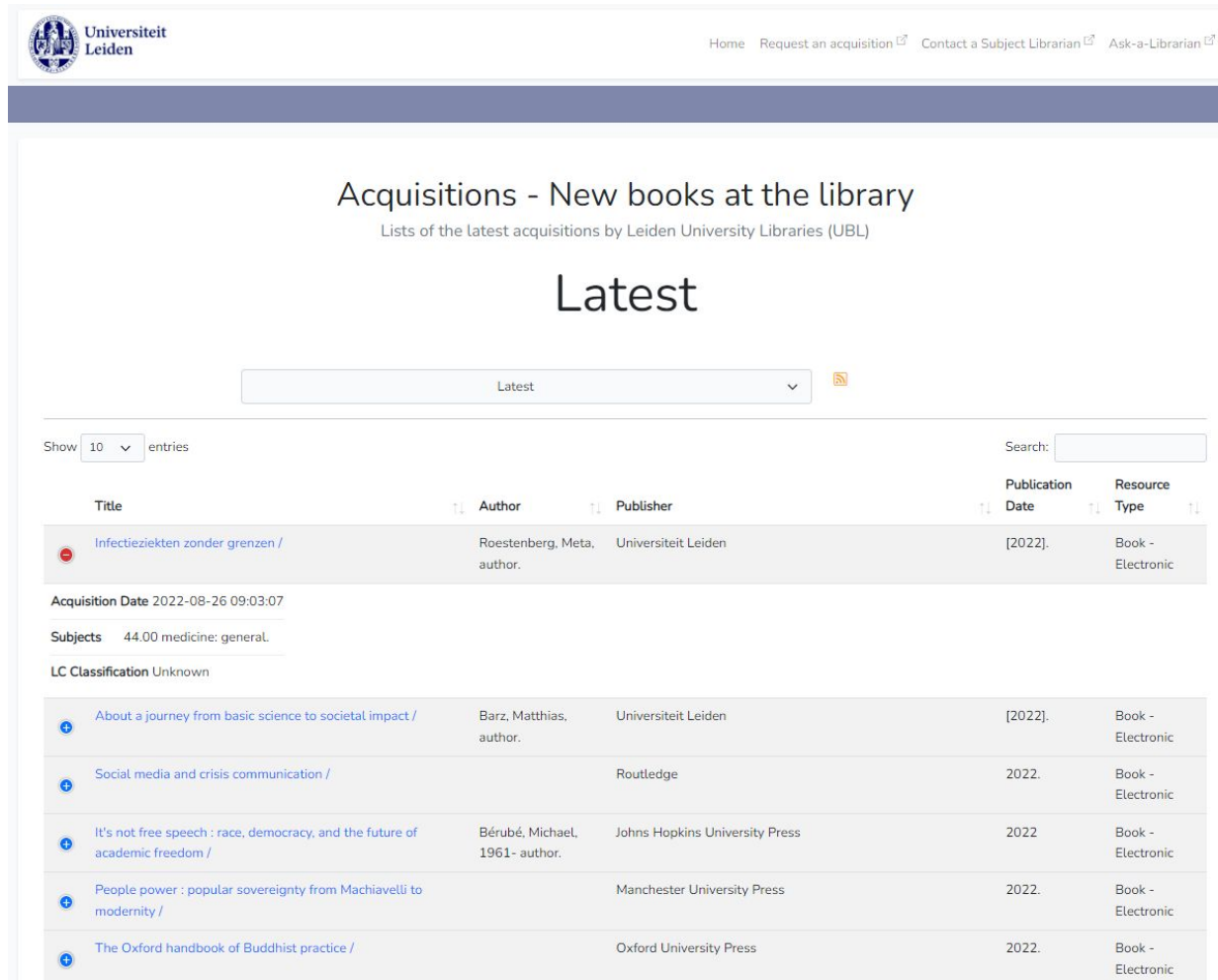
Display the results in the application

The application makes two Alma Analytics API queries:

1. "Give me a list of all the acquisitions reports we have in the 'Acquisitions' folder in Alma Analytics"
   - Used to display the list of acquisitions reports that the app's admin can generate and display to the users
   - Executes every time the admin requests it

2. "Give me the results from this specific acquisitions report in Alma Analytics and put them in a database table"
   - Used by the app to create and update each acquisitions list displayed to the user
   - Executes once a day per acquisitions report (or on demand, if needed)

Analytics - Ex Libris Developer Network (exlibrisgroup.com)

Display the results in the application

Programming challenges

- Fighting scope creep, managing feature requests

- Alma Analytics API returns results in XML

  - How do we put XML into MySQL?

- If an acquisitions report is deleted in Alma Analytics, what happens in the app?

- What if the Alma Analytics API takes too long to respond while creating / updating an acquisitions list?

- Main challenge: Working around the Alma Analytics API response time

The Alma Analytics API

An oversimplified view of how Acquisitions, Alma API, and Oracle work together:

1. Ask the Alma Analytics API: "What results are there for this acquisitions report?"

2. Alma Analytics API asks Oracle

3. Oracle replies to Alma Analytics API

4. Alma Analytics API sends the result to Acquisitions

The Alma Analytics API

- The issue is response time:

  - The time it takes for the Alma Analytics API to get a response is the same time it takes for the Alma Analytics website to display the results in the results tab when creating an analysis.

  - In other words: if it takes 5 seconds to display the results on the Alma Analytics website, it will take 5 seconds for the Alma Analytics API to get the results.

- We cannot query the API directly every time a user visits the site:

  - We consume calls towards the ExLibris API thresholds

  - It's too slow

The Alma Analytics API

- Possible solutions we thought of:

  - Cache the responses and serve them (See: [Reducing the number of API calls - Ex Libris Developer Network (exlibrisgroup.com)](exlibrisgroup.com))

  - Store the responses in a database and serve them

- We chose the database solution:

  - Better integration with our server setup (PHP + MySQL)

  - Less resource-intensive

  - Easier to maintain

Alma Analytics refreshes once per day, so:

We add the acquisitions to our database once per day and serve the results from there

The Admin Panel

Allows an admin to create, update, and delete acquisitions lists in the app (NOT in Alma Analytics)

- The admin of this application cannot make changes to Alma and Alma Analytics

- The data displayed in the Acquisitions app is the data from Alma

  - If the data is faulty (i.e. "wrong author name"), call the cataloguers, not the developer

  - If an acquisitions list contains books that are not part of the subject (i.e. "Russian ballads" in the "Applied math" acquisitions list), call the subject librarian that proposed the filters for Alma Analytics

The Quest for Visibility

- Search Engine Optimization (SEO)
  - Search Engine-friendly design
  - Automatically generated sitemaps
  - New acquisitions list pages submitted to search engines
- Interoperability
  - Exporting acquisitions lists to the University website to be displayed
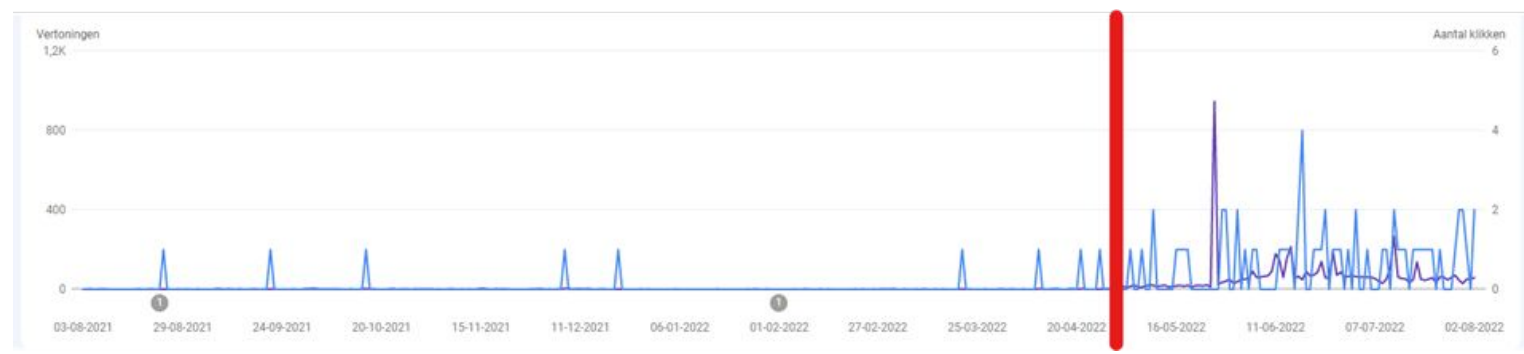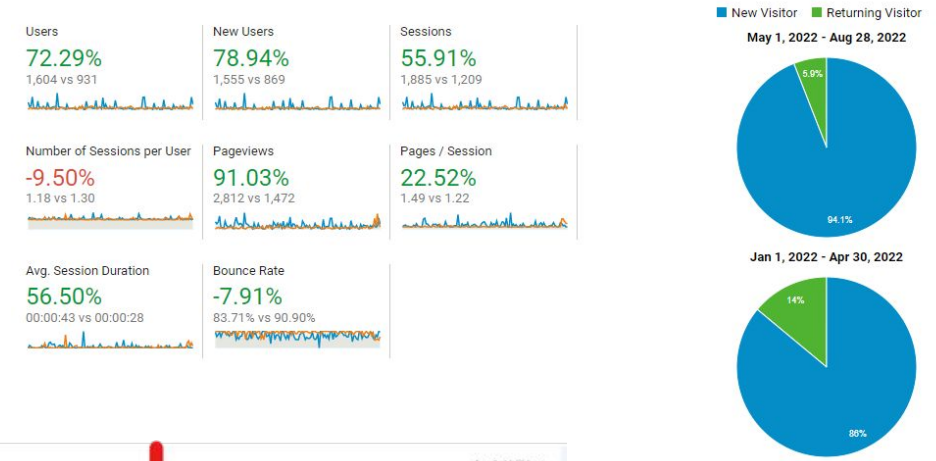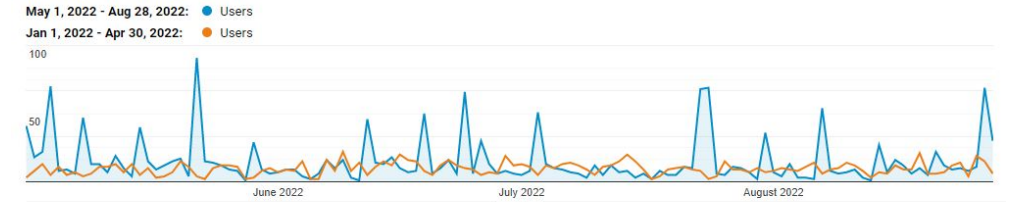 - RSS (for end-users and XSLT transformations)
  - Linking

The Quest for Visibility

Was all of this effective?

Too soon to tell (launched in May 2022, fewer users during summer)

Signs are positive:

- +91% pageviews

- +70% users

- +56% Avg. Session Duration

- + 55% sessions

- +20% pages per session

The future

- Code review

- Book covers

- Weekly overview of acquisitions per subject
  - Sent via email to users

- Continuous improvement!

Thank you!