

Leveraging Bepress's New API for Metadata Transformations

What is the project about?

For years, our law school has maintained a Bepress Digital Commons repository - the [Texas A&M Law Scholarship](#) repository - to help publicize our faculty's work and other important institutional information. Getting content and metadata put into the repository was straightforward enough, but getting content out of the repository was limited and difficult. After discussing this issue with Bepress, they got to work on developing better solutions, and they recently announced a [Digital Commons API](#) for customers to use that aims to improve the ability of users to engage with and export metadata and other information about the repository. This project explores how I use the new API to extract metadata from the repository and transform it into clean metadata for import into the researcher information system used by our university.

What are the project goals?

- Transform metadata from Bepress Digital Commons into clean metadata required by my institution's researcher information system.
- Eliminate duplicative work - instead of manually entering in the metadata into each platform, we can enter into Digital Commons and then export it out to different platforms using APIs and custom scripts.
- Get familiar with the API to explore other potential uses and applications (marketing publications or extracting information for ad-hoc queries, for example).

What tools/skills/applications are needed to do this project?

Projects like this really only require a little technical skill (familiarity with scripting languages and interacting with APIs) and time. For this, you'll also need access to the Bepress API by requesting a token from customer service, and documentation for either [Version 1](#) or [Version 2](#) of the API. During discussion, I can expand on more specific tools and applications that were helpful in making the transformations.

Things to look out for over time

This project is implementing a bridge solution, which is subject to being broken over time as initial circumstances change. For example, if Bepress or the RIMS changes the metadata schema being used, then this solution will be outdated and will no longer work properly. As a result, it is helpful to keep note of the different custom scripts being implemented across your services and applications.

Discussion Notes

- Metadata template created from consultation with team maintaining university RIMS
- You have to set up an environment to query the API and receive and process results.
 - I have a AWS LAMP server through AWS Educate that is a perfect fit for this kind of project.
 - For more complicated or larger projects, I would make a Github repository to store the code.
- Use Postman to get familiar with how to query the API (as explained in documentation). Export a site snapshot to get familiar with the available metadata fields and the overall metadata schema and repository data structure. With this snapshot, you can build ideas of what is possible with the data set.
- For this project, I am only interested in discovering and transforming metadata for faculty that are actively publishing and affiliated with the law school, so that I can “push” scholarship metadata from Bepress to our RIMS. This will eliminate the need for faculty or librarians to manually update the RIMS faculty profile. The RIMS is also being linked up to another activity reporting system, so being able to send updated scholarship information through this daisy chain is important.
- Review of metadata schema revealed Bepress has an “author id” for each of the faculty authors. This is really helpful for finding all works for a specific author within your repository!
- Phases of project:
 - Connecting to API & learning queries and metadata structure
 - Using scripting languages to make simple & complex transformations
 - Debug & learn along the way, capture a print out of entire tree and “Save as PDF” for reference when scripting.
 - Build simple interface to make faculty selection & export routine and easy.
 - Conceptualize future projects

- Mapping -- Bepress Field => RIMS Field
 - context_key => pub_id
 - title => title
 - CUSTOM => people_id (script will create value based on external table data since not present in Bepress metadata)
 - configured_field_t_book_title => parent_title [for book chapters]
 - abstract => abstract
 - Need to strip HTML elements out of abstract text
 - doi => doi
 - configured_field_t_issn_num => issn
 - configured_field_t_isbn => isbn10
 - Strip out - and count for 10 digits
 - configured_field_t_isbn => isbn13
 - Strip out - and count for 13 digits
 - configured_field_t_source_publication => journal_title
 - document_type => publication_type
 - RIMS limited to specific subset: academic article, book, chapter, conference, internet publication, report, repository items
 - Bepress types: article, book_section,
 - publication_date [PARSE] => month
 - publication_date [PARSE] => year
 - => begin_page (not coming through API stream)
 - => end_page (not coming through API stream)
 - configured_field_t_volnum => volume
 - configured_field_t_issnum => issue
 - author [PARSE] ; author_display_lname ; author_display ; author_userid => author_list
 - RIMS required specific format: [Surname, Initials], [Surname, Initials], & [Surname, Initials]
 - configured_field_t_name_editor => editor_list
 - NULL => pubmed_link [NULL for all]
 - configured_field_t_publisher => publisher
 - url OR download_link => repository_link
 - 'url' does not contain https://, parse that in ; use over download_link in case of link outs?

- Questions for RIMS Team
 - Can I have \n for in abstract or remove that too?
 - Does editor_list have specific format?