Blacklight Summit 2024 Agenda and Notes

This Doc: https://bit.ly/BlacklightSummit2024
YouTube Playlist of session recordings

Logistics

- Date: 10/16/2024 9:00am-4:30pm
- Location: Fine Visualization Hub, Engineering Library, Princeton
  Zoom: https://princeton.zoom.us/j/99871258732
- More logistics info in PUL Confluence

Agenda

- 9:00 - 10:00 am: Coffee
- 10:00 am - 12:00 pm: Morning session
  - 10:00 - 10:15 am: Introduction to Blacklight
  - 10:15 - 10:45 am: Community Updates
  - 10:45 - 11:00 am: Participant Introductions
  - 11:00 am - 12:00 pm: Demos
- 12:00 - 1:30 pm: Lunch (on your own)
- 1:30 - 4:30 pm: Afternoon session (subject to change)
  - 1:30 - 2:30 pm: **Lightning Talks** (Please list here)
    - (Jonathan) Blacklight_range_limit rehab proposal, jrochkind. https://github.com/projectblacklight/blacklight_range_limit/pull/266 .
    - (Brittny) Mirador & Blacklight UX/Accessibility Testing
      - https://crkn-rcdr.gitbook.io/user-centered-design
      - https://docs.google.com/document/d/1XXBKycGUG5YIquvnk5L7PABgIC-GvW8h8pW3QPdBKP4/edit?usp=sharing
    - (Eben) blacklight-allmaps (https://github.com/bplmaps/blacklight-allmaps)
    - (Bess) Blacklight-lando: https://github.com/bess/blacklight_lando
    - (Justin) Blacklight 9 https://docs.google.com/presentation/d/1vMrK4_69K6S9GrmvP26dSajzjJs6uaAil8Id6gImUrI/edit#slide=id.p
  - 2:30 - 2:45 pm: Break
  - 2:45 - 4:30 pm: **Unconference sessions**
    - **Unconference session (1): Upgrade Paths (2:55 -…**
    - **Unconference session (2): JS/asset pipelines - preferred patterns…**
    - Additional topics to be voted on at the summit. Please add your topics below.

Introduction to Blacklight

1. Introduction to community slides: Introduction to Blacklight
2. Communities (ArcLight, GeoBlacklight, Spotlight) slides: Community Updates

Participant Introductions

- Sean Aery (Duke University Libraries, Digital Projects Developer)
- Jesi Buell (Cornell)
- Tricia Clayton (Emory, Discovery Systems Program Manager)
- Karen L. Majewicz
- Matthew Phillips
- Phil Robinson (Cornell)
- Michael Kanning (Penn)
- Genia Kazymova (TRLN)
- Max Kadel (Princeton University Library, Library Software Engineer for Princeton, Discovery and Access Services Team)
- Eben English
- Cory Lown
- David Elyea
- Steven Ng (Temple University Libraries)
- Mary Zheng
- Lisa Croucher
- Megan May (Indiana, Enterprise Library Services Team)
- Emily Toner (Cornell)
- Jane Sandberg (Princeton, Discovery and Access Services Team)
- Beth German
- Christina Cortland (Cornell)
- Justin Coyne
- Brittny Lapierre
- Melissa Wallace (Cornell)
- Amrey Mathurin (Penn)
- Ferran Jorba (Universitat Autònoma de Barcelona)
- Sarah Chintomby (Cornell)
- Niqui O'Neill
- Huda Khan (Software developer, Stanford University)
- Dean Farrell (University of North Carolina at Chapel Hill Libraries)
- Christina Chortaria (Princeton, Discovery and Access Services Team)
- Esme Cowell (in person)
- Tracy McCormick (West Virginia University Senior Developer)
- Natalie MacDonald
- Isabela Souza
- (If in person, please put in your name)

- cat æscling (Utah, Software Developer for the Digital Infrastructure Development team at the Library)
- Bess Sadler (Princeton, research data team)
- James Griffin (Princeton, research data team)
- Sean Warren (Princeton, digital library team)
- Chris Colvard (Indiana University, developer/devops for Avalon Media System)
- Daniel Pierce (Indiana University/Samvera Hyrax Tech Lead)
- Ryan Laddusaw ((Princeton, Discovery and Access Services Team))
- Zeke Graves (Duke Libraries)
- Eric Griffis (NYU)
- Esmé Cowles (Princeton, Asst. Director, Library Software Engineering)

Demos

- Duke University Libraries (Sean Aery)
- Cornell University Libraries (Christina Cortland)
- Stanford University Libraries (Niqui O'Neill, Huda Khan, Cory Lown)
- University of Pennsylvania Libraries (In-person staff)
- Boston Public Library (Eben English)
- Triangle Research Libraries Network (TRLN) (Genia Kazymova)
- Indiana University Libraries (David Elyea, Chris Colvard, Daniel Pierce)
- Princeton University Library (Esmé Cowles)
- West Virginia University Libraries (Tracy McCormick)
- Canadian Research Knowledge Network (Brittny Lapierre)
- Temple University Libraries (Jennifer Anton, David Kinzer, Chris Doyle, Steven Ng)

Lightning Talks

- Blacklight range limit plugin (Jonathan) - date limit plugin
    - https://github.com/projectblacklight/blacklight_range_limit/pull/266
    - Digital collections for science history institute museum and library
    - Date facet for limiting search: Histogram graph. Some people have found a way to do limit plugin without the histogram.
    - Does not work with Blacklight 8, because of JavaScript issues
    - Written 10 years ago
    - Stakeholders wanted Jonathan to work on updating that
    - Experimenting with seeing how this might work - community may or may not decide to keep it, and may or may not want to use the histogram approach.
    - Rewrote:
        - ChartJS - only dependency. No JQuery
        - Original alpha: histogram, no sliding bar, not interactive
        - Close to being accessible - screen reader only list of bars, text-only version
        - Would still be rescued to work with Blacklight 8 or 9

        - Not going to support legacy sprockets. Will use Modular JS/ESM. Will support importmap or propshaft with sprockets. Supports any JS bundling with Vite/Ruby with importmap or propshaft. Passing CI tests.
        - Week on actual JS and two weeks on the CI - interfacing with Blacklight and all the different JS mechanisms.
        - Would be a new major version of Blacklight range limit.
        - Experimental work in one branch/one PR. Difficult to do this complex work and breaking it out.
        - Would like it to be Blacklight range limit 9.
        - Questions:
            - Q: BRL or V1 of a new thing?
                - Jonathan's good either way
                - No further version of BRL if no one continues to work on that
                - Benjamin: Not suggesting shouldn't be a community supported plugin. Given that it is a total rewrite/different features/APIs are different - wonder about value saying it is BRL 9 or instead histogram plugin
                - Backend at controller level is actually the same. Differences are in the JavaScript section.
            - From slack: thanks for all the work
            - Q: What features are not included
                - [there was another feature here :)]
                - Hovering over histogram for getting count - not in alpha
                - Drag/click to define range: not in alpha
            - Automatic installer may not work for every situation, but tool will
            - Want to get it to work with BL7 - and then get it to work with 8. As well as rails 7.
            - Would love alpha testers
- Mirador & Blacklight UX/Accessibility Testing
    - https://crkn-rcdr.gitbook.io/user-centered-design
    - https://docs.google.com/document/d/1XXBKycGUG5YIquvnk5L7PABgIC-GvW8h8pW3QPdBKP4/edit?usp=sharing
    - Working on replacing old Perl codebase with new work
    - Putting out demos for test users. Over and over will be running UX and accessibility studies.
    - Process for each
        - Recruitment
        - Accessibility study
        - Analyze in design sprint
        - Create prototype
        - Conduct usability test on prototype
        - Gather more feedback on design

        - Publish finalized designs: Discuss with open source community members to see if proposals will make sense in the context of the application/system
    - IIIF Viewer and accessibility
        - Created generalized recommendations with some feedback from community. Digerati and NYU also helped.
        - Asked IIIF community: does anyone have any more ideas, etc.
        - Recommendations:
            - Avoiding deeply nested options in displays
                - One of the issues encountered in Mirador - image viewer. Many icons. Click to get menu item. Then click to get feature. Appeared to be pain point for both UX and accessibility reasons.
            - Displaying OCR
                - Text based recommendations of images being displayed, e.g. videos, scanned book or magazine content with OCR. Make sure these are displayed to end users in a convenient way
                    - Most users prefer a left side pane to the viewer - shows up first and is easier to access
                - Allows people to use screen reader to read through content
            - Providing clear navigational aids
                - Table of contents or Google doc TOC content UI
                    - E.g. Big books, big newspapers, big magazines.
                    - Community CA portal - need to go through one by one to see content
                    - Feature would increase accessibility
                    - Can use IIIF ranges for TOC and IIIF annotations for outline like capabilities. Outlines are more concrete/discrete in what they do. Annotations are better mechanisms for providing outlines.
        - Can also add suggestions
        - Supporting research
            - Data from accessibility testers
            - What software they use
            - E.g. Clustering exercise: Raw data from accessibility testers. Found themes to see what problems were being discussed in clusters.
            - Usability testing:
                - Breakdown of the changes that were made based on the observations
                - Screenshot of what step of the workflow looked like at the time

                - After UX/accessiblity testing: will add OCR content as side pane, simplified view.
        - Doing the same steps here for IIIF with Blacklight. Specifically around facets
            - Could put out a survey - between usability study and design sprint
            - Could work with community to come out of the solutions, and not just an after the fact situation
        - Questions
            - Maze: Have not seen that before
                - app.maze.co/projects. User testing software. Like Google analytics. Add tag to your website. Will support users going through steps that you setup. Can also do with Figma diagrams.
                - Build: Live website test. Add “blocks” and steps - what website. What you want people to do at that step. Tell software what a successful path looks like.
                - Publish and share link to test users. Can watch the results come in.
                - Need to pay for each seat/who has edit access. Share report out to folks so they can see the results.
                - For one test: not automatically reporting if users have direct/indirect. Watching directly.
            - Love user-centered design GitBook. Term: Design Ops
                - Love how it is baked into the process
                - Beth German: Article about DesignOps:
                    - Blakiston, R., Liu, Y. (Bob), & Miller, L. (2021). DesignOps in an Academic Library: Reducing Barriers, Improving Efficiency, and Scaling Impact of Design and Research. Journal of Library Administration, 62(1), 31–46. https://doi.org/10.1080/01930826.2021.2006984
            - Handling situations with multiple ways to get to success
                - Indirect success: one way to categorize either less efficient flows - or other paths. Gave
- (Eben) blacklight-allmaps (https://github.com/bplmaps/blacklight-allmaps)
    - Maps that are georeferenced
    - Georeferencing: Taking a raster image like a scanned map, usually a historical map. Correlating pixels in the image with geographical layer. “Warping the map” against the base layer, so you can view the historical map over the base layer.
    - Allmaps: open source platform for georeferencing maps. Software as a service platform. Could run allmaps server if you wanted to. IIIF manifest for map image object. Interface to start georeferencing process. Picking points on historical map and corresponding to point on map base layer.
        - Saves data as IIIF annotation. Can thuse use IIIF
        - Could download annotations and store
    - New gem for blacklight released this spring. Some early adoption. Getting more people aware of it. Similar installation process to other blacklight plugins

        - Solr fields to correspond to different pieces of information
        - BIG 10 alliance geoportal:
            - Aggregating maps and GIS data from various BIG 10 institutions
            - GeoBlacklight instance
            - Blank search
            - Facet: Georeferenced via allmaps plugin
            - Map: plugin provides this viewer window based on open sea dragon and uses IIIF image API to render the content. Zoomable viewer.
                - Georeferenced tab: Shows map content overlaid on google maps basemap
            - Can drag the opacity to view more or less of base layer
                - E.g. can see an area that used to be all water now has a highway. May be of interest for urban planning/historical info etc.
        - Sidecar database to share annotations. So not fetching annotations in real time.
    - Questions: Data doesn't really sit in the index, but in the sidecar, which means Solr is technically untouched
        - Everyone has their own way of updating their Solr data, so don't want to put that into the index
        - But there is a rake task to check Solr index against what is in all maps to update the sidecar/records in database so you could see if there are new maps now in allmaps. Could then write a script to update your Solr index based on that.
- (Bess) Blacklight-lando: https://github.com/bess/blacklight_lando
    - Wrote this because teach a lot of getting started with blacklight workshops
    - Not the best of luck with the solr wrapper gem on the Blacklight wiki
    - Use Lando to run docker containers for local development
    - Didn't want to walk a classroom of people through involved configuration process
    - Put that configuration into a gem called blacklight lando
    - Install:
        - Setups configuration for Solr using Blacklight defaults. Both development and test. Configures to run against the right version of Docker.
        - Then run "lando start"
    - Want to add PostGres in - future step
    - Q:
        - Way to configure Solr version.
            - Yes in config file.
- (Justin) Blacklight 9
    - Timeline: Sometimes towards quarter 1
    - Changes:
        - Bootstrap 4 to be removed. *Bootstrap 5 only*.
            - BL7 and 8 support both Bootstrap 4 and 5.

        - Rails:
            - Different asset pipelines for JS:
                - Rails 8 has propshaft as default, does not have sprockets anyway.
                - Importmap, js-bundling-rails, cssbundling-rails
                - Moving away from Sprockets
            - Moving away from build pipelines to “no build” world
        - “SASS” optional in 9, because of this “no build” philosophy
        - Minor UI changes
            - Avoid many UI changes unless necessary. Stacked up some changes to target accessibility and putting in major release.
    - Can add issues/feature requests to get them into the new version
    - Do need help: Review PRs for this milestone so they can be merged in. Issue: when you merge one, need to rebase the others. Getting them in regularly would be very helpful
        - https://github.com/projectblacklight/blacklight/milestone/54
    - Q:
        - Rails 8 is looking at SQLite as production ready. Blacklight tends to be pretty light on database - would we consider recommending SQLite as part of our trainings/tutorials for setting it up?
            - Don’t necessarily want to dictate databases. Not sure if SQLite meets our own production needs b/c need multiple fail over nodes. Would depend on needs for application.
        - BL9 still using view components for front end views?
            - That part is the same.
        - Q: Upgrade from 8 to 9 maybe less painful from 7 to 8 if already using Bootstrap 5?
            - Depends on experience in the past.
        - Q: Clarify propshaft comment?
            - Rails 8 has propshaft as default and no longer sprockets.

Unconference Sessions

Add topics you would like to discuss below by 10/16 at noon Eastern.

- Upgrade Paths
- JS/asset pipelines - preferred patterns
    - See some related agenda items added for non-existing community meeting by jrochkind here: https://docs.google.com/document/d/1wzgqPM5srMOt_20yyZCtWjFEU3TOIw5cL5zm22X2Wys/edit?tab=t.0#heading=h.tg4oc21g1zjx

- I'd like to hear about some of the differences between 7 and 8 - is this something that would be covered in the Upgrade Paths discussion or would it be separate? (What do we get out of upgrading?)

Unconference session (1): Upgrade Paths (2:55 - 3:40 ET)

Topic/objective:

- Differences between Blacklight 7 and 8. What is involved in upgrading?
    - What can be done in portions?
    - What needs to be done altogether?
- Experiences with upgrades that people may already have done at their institutions (whether in progress for a particular application or planning out details)
- Specific challenges/sticky points

- Facilitators: Emily Toner (Cornell)
- Note takers: Jesi Buell (Cornell)
- Gatekeeper: Huda (Stanford)

Notes:

- There is an "Upgrade to Blacklight 8 guide" in the wiki https://github.com/projectblacklight/blacklight/wiki/Upgrading-to-Blacklight-8
- Being on the latest version of Rails and Bootstrap that you can be on will make Blacklight upgrades go easier.
- Princeton has started but stopped because the date range functionality isn't there but do have a plan
- Isn't possible to have customizations that work in both 7 and 8 (science history institute in philadelphia)
- UPenn has updated Blacklight to 8, got rid of range limit plugin that was hindering update (using view components) - took a year - for their archival discovery blacklight
- Note from chat
    - "Arclight dumped blacklight_range_limit mostly because nobody wanted to re-write it. https://github.com/projectblacklight/arclight/pull/113"
- Iiif-search and (part of) blacklight-advanced search folded into 8
- Stanford (as part of/with these communities) has done work in updates in spotlight and geoblacklight, things have changed with view components; also javascript bundling has changed
- Diagram of how view components layout - https://github.com/projectblacklight/blacklight/wiki/Results-views-in-Blacklight-
- Indiana U - xray rails gem showed how partials overlaid each other, is there any analogous with view components? "No, but you can turn on annotate-render-view with file names, it will render those in HTML"
    - config.action_view.annotate_rendered_view_with_filenames = true

    - (see example: https://github.com/sul-dlss/SearchWorks/blob/main/config/environments/development.rb#L73)
- Also see: https://blog.corsego.com/annotate-views
- UPenn replaced app that used Blacklight 6 - going forward, they are creating apps with less dependencies/plugins - able to do that bc they have the team to customize
- BPL - how to have conversation with stakeholders about dropping functionalities?
    - Realities of support needed, and the caveat of the change being potentially temporary
    - Priority is really having a maintainable app
- UPenn - blacklight marc gem, overrode large portions, so they templated out librarian view // also citations, almost got overridden - they let go of dependencies
- Stanford notes in chat that range limit facet isn't accessible
- UPenn's project took a little over a year and 3-6 devs (ramped up towards end)
- Comment about possibilities for working on upgrades:
    - Spending more time working with main branch directly to see if app can work with updates as they continue
    - Comments from Zoom:
        - Do not usually work on main, because cannot keep up with all the changes
    - How to approach keeping in sync with minor releases
        - Would say cannot work on enhancements when working on maintenance
        - Break up some part of sprint for enhancement and also for maintenance
        - Quarterly sprints
    - Esme from slack:
        - one thing we did was have our Library leadership team read the British Library's post-incident report (https://www.bl.uk/cyber-incident/) and talk about the consequences of not keeping up with technical debt.
        - Zoom: Talking about your own institution's situation - with which applications may not be able to come up after some kind of issue (restoration/vulnerabilities, etc.) - would also clarify the need for maintenance.
    - Michael: Setting things up so that can move on to a new release within 6 months. Need administrative buy-in for this kind of thing.
    - Bess (from slack): https://blog.drinkbird.com/twenty-percent-rule/

Also from slack:
Useful in the rails upgrade process: https://github.com/Betterment/uncruft, https://rubygems.org/gems/next_rails

Terms:
View component: https://viewcomponent.org/, not to be confused with web components

View components are a particular method of encapsulating display and functionality. Earlier Blacklight version were fine with the partial view of updating content but with 8, there has been a complete move to the use of view components. This means the code has been rewritten to use this new path, and any upgrade path - especially where there were customizations to views in folks' own Blacklight installations - requires rewriting the customizations to follow the new view component method

Sidekiq: https://sidekiq.org/ . Background jobs that can run, triggered by the app, etc. Can be used, for example, for indexing in response to getting new data or an update to existing data.

Unconference session (2): JS/asset pipelines - preferred patterns (~3:50 - 4:30 ET)

Topic/objective:

- Facilitators: Eben English (Boston Public Library)
- Note takers: Chris Colvard (Indiana University), Esmé Cowles (Princeton University Library)
- Gatekeeper?

Notes:

- Rails JS bundling solution churn in past few releases
- Is there a community consensus on which JS bundling approach to take?
    - Best practice for least amount of pain in the future when upgrading?
- Sprockets was the old rails way that bundled assets and provided fingerprints for the files. Sprockets could not use npm for libraries and it wasn't very good at modern javascript modules. Sprockets did not follow the javascript community, so it wasn't well liked.
- What is the current state of Rails JS bundling options (importmaps/propshaft/etc.)?
    - Jonathan - jsbundling-rails (don't pick the webpacker option)
        - Vite - not officially supported by rails but great! (From slack: vite++ We do indeed use vite at Princeton and we're very happy with it. https://vite.dev/)
        - Importmaps: not bundling
            - Importmap is part of your browser: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script/type/importmap
            - https://github.com/rails/importmap-rails
            - This is one of the best ways for caching. Only changed files get loaded. In a bundled approach changing a single character means downloading the whole bundle again.

        - Sprockets on the way out
        - JS should look the same no matter which approach taken
        - PR build for esbuild/jsbundling-rails that passes without code changes
            - https://github.com/projectblacklight/blacklight/pull/3360
- Rails dropping sprockets/asset pipeline so need a new solution to handle JS/CSS
- How to include JS packages outside of rails gems? Sprockets couldn't handle it.
    - Rails tried to do it in the way JS community was….which was evolving
    - At a minimum BL JS needs to be included in your app
    - Options: importmaps (supported by BL right now) or jsbundling-rails
        - Jonathan: complex dependencies can be difficult
    - Sprockets bundled and fingerprinted JS allowing for caching
        - Propshaft handles fingerprinting piece
- Importmaps doesn't handle CSS bundling
    - Justin - need cssbundling-rails to do this bundling
        - Only need if doing SASS but if you precompile SASS then don't need to bundle anymore
    - (from slack): In the spirit of simplicity, we went with importmaps in our new catalog and it mostly just works. We don't have heavy JS dependencies. https://github.com/upenn-libraries/Find/blob/main/rails_app/config/importmap.rb
- How to handle libraries that have JS AND CSS?
    - It depends
        - Example: reference Bootstrap from CDN and have local CSS to set variables; more complex need more tools
    - Vite - not officially supported by rails
        - Does both CSS and JS using JS dependency techniques: yarn
    - From chat:
        - https://vite-ruby.netlify.app/
        - Here's an example of not needing SASS anymore with Bootstrap: https://sul-dlss.github.io/component-library/header/
        - We just import https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css And https://sul-dlss.github.io/component-library/styles/sul.css for customization
- What tools/approaches are people using?
    -
- What is the Blacklight approach?
    - Justin - Blacklight doesn't depend on any particular approach
        - Support importmap-rails because that is the Rails default
        - Could support jsbundling-rails since it shouldn't be hard
        - Similar for vite
        - Continuing to support sprockets would be hard
    - Jonathan agrees with Justin
        - Simplest might be importmap-rails

        - Prefers vite
    - Rails 7+ new app can be run with options to setup importmaps/jsbundling-rails and then Blacklight will use it fine
    - Blacklight 8 new app without any options would be sprockets right now (yikes)
- Don't start with webpacker/shakapacker either…it's also at the end of the road
- Lots of JS bundling options because they are converging and switching is very easy
- Tree shaking - new bundling tools only include JS that is actually used on the page
    - Importmaps includes all code
- Is there any dev/deploy tasks that need to be run with these tools?
    - Yes, rake assets:precompile still needed
- Huda - seems like it has been a lot of trial and error to figure out what works
    - Maybe start small and do a little first
    - Issues in the interplay between all of these tools and layers of tools
- Eben - it's all in your mind; prepare to be patient
- Jonathan can move over little by little - mix of sprockets and webpacker/vite but need to have those two JS pieces independent
- Bess (Princeton) - pretty much all moved over to vite
- JS bundling has maybe reached a more stable point so don't need to wait
    - Safe to switch to jsbundling-rails
    - Justin likes certain features of importmaps-rails but not as well supported
- We'll revisit next year and see if everything has changed...

(from slack): https://gist.github.com/taylor-steve/113df2dc64ceaab874fed0d1fdaed4ca is a useful resource explaining the different aspects of bundling, etc. and what changes from the "old ways" we are trying to address

(from slack): Mike: we've been able to avoid writing a lot of JS by using turbo (https://github.com/hotwired/turbo-rails). turbo-frames and view components can work very nicely together. here's an example: https://github.com/upenn-libraries/Find/blob/main/rails_app/app/components/library/info_component.html.erb

Translations - Type a term and I'll try to provide a definition (Ryan)

- Importmap
    - Importmap is part of your browser: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script/type/importmap
    - Rails has https://github.com/rails/importmap-rails that delivers your assets via importmap
    - https://www.honeybadger.io/blog/import-maps/
- Vite
    - JavaScript build tool
    - https://vite.dev/

    - Also see https://vite-ruby.netlify.app/
- SASS
    - Syntactically awesome stylesheets. A stylesheet language that compiles into css and offers a lot of features, some of which are now supported by most browsers
    - https://sass-lang.com/
- NPM:
    - Node package manager. JavaScript package registry and command line tool for installing JavaScript dependencies.
    - https://www.npmjs.com/
- ESM: https://webpack.js.org/guides/ecma-script-modules/
- Jsbundling-rails allows you to use rollup, esbuild or webpack. It's very easy to use.

Notes