2018-03-01 DSpace 7 Working Group Meeting notes

Date

01 Mar 2018

Discussion (combined Angular UI & REST meeting)

Attendees

- Andrea Bollini (4Science)
- Art Lowel (Atmire)
- Tim Donohue
- Giuseppe Digilio (4Science)
- Tom Desair
- Patrick Trottier

Location: DSpace Meeting Room

Notes

- General Updates
    - Planning Spreadsheet: https://docs.google.com/spreadsheets/d/18brPF7cZy_UKyj97Ta44UJg5Z8OwJGi7PLoPJVz-g3g/edit?usp=sharing
        - Latest status is still there
    - Tim notes recent discussions from DSpace Entities Working group (see 2018-02-23 DSpace Entities WG Meeting Notes)
        - Also discussed in DSpace DevMtg yesterday. Logs at: http://irclogs.duraspace.org/index.php?date=2018-02-28
        - More discussion on this will be occurring on dspace-devel list
- Angular UI Team updates (via Art)
    - Merged PRs:
        - https://github.com/DSpace/dspace-angular/pull/2297
    - Reviewed (feedback by Tim)
        - https://github.com/DSpace/dspace-angular/pull/228
            - Needs testing (Tim will look at)
            - **ACTION (Tim):** Update our README to make it clear how to change dependencies
            - **ACTION (Tim):** Update our README to also provide hints/tips on quickly testing PRs
    - Needs Review:
        - https://github.com/DSpace/dspace-angular/pull/221
            - Has been waiting for many weeks
            - Tim and/or Patrick will look at
    - Giuseppe's PR on Auth by Credentials: https://github.com/DSpace/dspace-angular/pull/230
        - Art will review
    - Lotte is looking at Date Widget
    - Lotte also starting to take over Search Backend code now (Hooking up Angular side to REST API.)
        - She contacted Giuseppe to ensure he hadn't started yet
        - Giuseppe & Lotte/Art should stay in touch about this code, so that two individuals aren't working on same code at same time.

- REST API Updates (via Andrea)
    - Workflow contract PR now available: https://github.com/DSpace/Rest7Contract/pull/20
        - Andrea feels this is a good start, but feels there are improvements that can be made in time for DSpace 7
    - 4Science plans for March
        - In March, working on a separate project to implement Workflow REST API in DSpace-CRIS. This work will align with the Contract as documented in PR#20 (see above)
        - In April, this work will come back to DSpace 7.
        - The two codebases (DSpace-CRIS and DSpace 7) are not too dissimilar at this time. So, the effort should be easily adaptable. However, the DSpace-CRIS codebase doesn't yet have the new Code Style rules, so there may be some cleanup to get the Workflow REST API moved into DSpace 7.
    - Question from Andrea: Should we work to enhance the REST Contract to be "more correct"? Or should we accept PR#20 for now, and merge the code based on that (in April)?
        - Tim notes that Contracts are rarely 100% correct at the beginning. You will likely learn a lot during the building of the code (for DSpace-CRIS) that can help enhance/improve the contract. Suggests iterating here...accept contract as-is, work on implementation code against that version, then iterate and improve contract & re-align code.
        - Tom agrees
        - Art agrees.
    - Tim notes PR to align Unit Tests with Code Style (needs reviewers): https://github.com/DSpace/DSpace/pull/1975
    - Tom notes PR to update commons-collections (needs reviewers): https://github.com/DSpace/DSpace/pull/1931
        - Tim will look at this again
    - Tom also notes Hibernate fixes in this PR (needs reviewers): https://github.com/DSpace/DSpace/pull/1964
        - NOTE that this changes many queries. So, it could use more eyes
        - We suggest merging it quickly though as any errors in query refactoring should then be caught by future REST API development. As of now, all Unit Tests pass though, so queries should be OK.
    - Tom also notes Authentication response PR: https://github.com/DSpace/DSpace/pull/1974
        - New idea here to move to using WWW-Authenticate header + custom challenge types

    - This header is not currently supported in browsers, and there's not really any standard types of responses (other than "Basic" for Basic Auth): https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/WWW-Authenticate
    - Tom asks is this ok to define our own authentication types (e.g. "password", "shibboleth", etc)? Anyone see issues?
    - Tim says it seems reasonable if we document this in the REST Contract.
    - Andrea notes there may be some risk here, if other libraries or tools assume that WWW-Authenticate will only ever return "Basic" as it's authentication type. We may end up having to revisit this in the future once browsers standardize on this or another such header (if they do so).
        - Should we consider just doing this in our own custom way?
    - Art doesn't think we should have issues on Angular side of things with WWW-Authenticate
    - All agree this seems OK to move forward, but should do the following:
        - First, do more testing with WWW-Authenticate. Make sure Angular is good, HAL Browser is good. See if other tools make improper assumptions here
        - Second, look around to see if any other tools similar support multiple Auth options
        - https://www.iana.org/assignments/http-authschemes/http-authschemes.xhtml
        - **ACTION (Tom):** Look into the above and
  - back on Log4j upgrade: Unable to locate Jira server for this macro. It may be due to Application Link configuration.
    - a PR. A work-in-progress PR is also welcome, if you want early feedback/support from others
    - Will ask questions on Slack or email (dspace-devel) as they come up.
  - Discussion of Submission Logging issues (from Tom)
    - Problem: Because we do some caching in the Angular layer, not every page click/hit will generate a hit to the REST API. So, for logging page hits to Statistics, we'll have to find another route to ensure any hits that go to the Angular cache also still get logged to Stats
    - Idea to use Web Sockets to log these to Stats. Any objection?
    - Art notes the other option would be to do something similar to Google Analytics...when the route changes, send info back to the server-side on the new location/hit to log it.
    - Web Sockets: Easier to track behaviors that are not route changes.
        - can also communicate changes on the backend (server) to the client as well...e.g. via a log file via web sockets
    - browser support looks good these days : https://caniuse.com/#feat=websockets
    - Andrea recalls that Texas A&M did work with Web Sockets in their UI Prototype
        - WebSockets + Spring Boot + Angular (v1) was in Texas A&M's prototype (See prototype #7 on this page, and there was a video on this prototype too): DSpace UI Prototype Challenge#PrototypeSign-Up
        - Code for that prototype still on GitHub (see above wiki page)
        - Tom will get in touch with Texas A&M folks
    - Tom has a branch he's started playing with websockets on: https://github.com/atmire/DSpace/commits/DS-3574_Log-stats-websockets-jms
    - More digging to be done here...but all agree that websockets will be useful in other areas of our application. So, we shoul learn more and figure out how best to start using them.
- **Next Meeting will be Thurs, March 8 at 15:00UTC in DSpace Meeting Room**