2019-10-10 DSpace 7 Working Group Meeting

Date

10 Oct 2019 from 14:00-15:00 UTC

Location: https://lyrasis.zoom.us/my/dspace (Meeting ID: 502 527 3040).

- More connection options available at DSpace Meeting Room

Agenda

- (15 mins) Developer Stand Up - Developers give brief updates on their effort (or their team's effort).
    - Update/see "Current Work" section below based on your status. *Please feel free to update prior to meeting.*
    - Please highlight any new work (needing reviews/testing), any blockers (for you), and any discussion topics you may have.
- (30 mins) General Discussion Topics
    1. REST API discussion: Reminder to all (self included) which endpoints can support which methods
        a. Per our Contract README: https://github.com/DSpace/Rest7Contract/blob/master/README.md#use-of-the-http-verbs-and-http-response-code
        b. Multiple Resource (i.e. Collection) Endpoints (e.g. /items) may support only POST and GET
        c. Single Resource Endpoints (e.g. /items/:uuid:) may support GET, HEAD, PUT, PATCH and DELETE
        d. Sub-paths only act on *associations between resources* (e.g./items/:uuid:/mappedCollections), and only allow GET, PUT, POST, DELETE **of those associations**.
            i. We may need to revisit the decisions in these two recent PRs:
                1. https://github.com/DSpace/Rest7Contract/pull/79 (Community & Collection Logo). Should PUT/DELETE just use /bitstreams/<:logo_uuid>? (as Logos are Bitstreams!)
                2. https://github.com/DSpace/Rest7Contract/pull/80 (Collection Item Templates). Should PATCH/DELETE just use /items/<:itemtemplate_uuid>? (as Templates are Items!)
    2. Revisiting our Code Review processes for PRs
        a. Up until now, all PRs require two code reviewers
        b. As we've seen, this provides good feedback, but can cause review/implementation delays if only one reviewer can be identified in a timely fashion
        c. This issue was brought to Steering Group. Recommendation is to consider allowing for single reviewers for smaller / more obvious PRs, with option of that reviewer asking for an additional reviewer if they find something requiring additional feedback.
    3. Revisiting REST API Projections: https://jira.duraspace.org/browse/DS-3533
    4. Reviewing & scoping tasks flagged as "NEEDS MORE INFO" in Development Planning Spreadsheet
        a. This is necessary to prepare for the next phase of the DSpace 7 Estimation Process
- (15 mins) Planning for next week
    - Assigning PRs for Review
    - Next tasks from Development Planning Spreadsheet

Attendees

- Art Lowel (Atmire)
- Andrea Bollini (4Science)
- Tim Donohue
- Lieven Droogmans
- Mark H. Wood
- Giuseppe Digilio (4Science)
- Ben Bosman
- Chris Wilper
- Paulo Graça
- Dimitris Pierrakos
- Julius Gruber
- Laura Henze

Current Work

Legend for status icons
★ = Highest Priority tasks (please prioritize these reviews/tasks over others). These are tasks with lots of dependencies
✖ = review done (this week), changes were requested.
✔ = review done, approved.
⚠ = review done, merge conflict or other minor changes requests

Tickets / PRs In Progress

1. (Angular) Adding Accessibility via Travis CI https://github.com/DSpace/dspace-angular/pull/356 (work in progress) (*Lower priority*)
2. ⚠ (Angular Bug) https://github.com/DSpace/dspace-angular/issues/368 (Art Lowel (Atmire))
3. (REST Contract) Edit Homepage news: https://github.com/DSpace/Rest7Contract/pull/45 (Ben Bosman - has outstanding questions/comments) (*Lower priority*)
4. (REST) DS-4043: Revisit the security layer of the submission (work in progress) Andrea Bollini (4Science)
5. (REST) Pagination bug with withdrawn items: https://github.com/DSpace/DSpace/pull/2406 (Dimitris Pierrakos , Ben Bosman - Feedback provided)

PRs Needing Review

1. (REST Contract) Group and eperson management: https://github.com/DSpace/Rest7Contract/pull/41 (Tim Donohue - feedback provided, Andrea Bollini (4Science) - ⚠ feedback provided)
2. (REST) Issue when community has multiple dc.title values https://github.com/DSpace/DSpace/pull/2486 **UPDATED** (✔ Tim Donohue , ✔ Andrea Bollini (4Science) - merged)
3. (REST) Spring security for createAndReturn with parent id https://github.com/DSpace/DSpace/pull/2489 (✔ Tim Donohue , ✔ Chris Wilper)
4. (REST) Authority control bugfixes https://github.com/DSpace/DSpace/pull/2528 (✔Tim Donohue, Giuseppe Digilio (4Science), Andrea Bollini (4Science))
5. (REST) (Entities) CS n, ✔ Paulo Graça - also created)
6. (REST) Scripts and p
7. (REST) DS-4337 implement bitstream-bitstreamformat relation endpoints https://github.com/DSpace/DSpace/pull/2503 (Tim Donohue ⚠ Feedback provided, Andrea Bollini (4Science) ⚠ feedback provided, Ben Bosman)
8. **(NEW)** (REST) DS-4359 add null check to getAllRelationshipTypes https://github.com/DSpace/DSpace/pull/2545 (Ben Bosman)
9. **(NEW)** (REST) DS-4360 fix searchevents link https://github.com/DSpace/DSpace/pull/2546 (Ben Bosman)
10. **(NEW)** (REST) Ds 4317 bundles in rest https://github.com/DSpace/DSpace/pull/2548 (Ben Bosman, Tim Donohue, Chris Wilper)
11. **(NEW)** (REST) Proof of Concept REST Projections: https://github.com/DSpace/DSpace/pull/2547
12. (Angular) Item-Collection Mapper: https://github.com/DSpace/dspace-angular/pull/348 (Tim Donohue - ⚠ **REREVIEW**, ✔ Art Lowel (Atmire))
13. (Angular) Shibboleth integration support: https://github.com/DSpace/dspace-angular/pull/429 (Giuseppe Digilio (4Science) reviewed again **fixed error with yarn start**, Fernando FCT/FCCN, Paulo Graça - feedback provided)
14. (Angular) forceBypassCache should be removed from the RequestService: https://github.com/DSpace/dspace-angular/pull/468 (✔Art Lowel (Atmire), ✖ Giuseppe Digilio (4Science) issue found)
15. (Angular) Collection pages WIP: https://github.com/DSpace/dspace-angular/pull/472 (Art Lowel (Atmire) - feedback provided, Tim Donohue - REREVIEW AND MERGE)
16. (Angular) Routing by handle and uuid: https://github.com/DSpace/dspace-angular/pull/490 (Art Lowel (Atmire), Giuseppe Digilio (4Science), Tim Donohue , Andrea Bollini (4Science) - might be able to help with pid endpoint)
17. (Backend) dspace.bat file: https://github.com/DSpace/DSpace/pull/2544 (Tim Donohue, Alexander Sulfrian) - one approval.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

PRs Merged this week!

1. ✔ (REST) Fixing exception error when using UUID on Harvesting process https://github.com/DSpace/DSpace/pull/2536
2. ✔ (REST) Oai harvesting setup https://github.com/DSpace/DSpace/pull/2491

Blocked

1. (Blocked PRs go here)

Delayed / Needs Discussion

1. **Managing Authorization info in Angular UI**: How to pass Authorization rights (i.e. logged in user's access rights) from REST API to Angular? See for example: https://github.com/DSpace/dspace-angular/issues/393
    a. Can this be achieved via passed HAL "_links" (e.g. the existence of an "edit" link in REST response means you must have Edit rights)?
    b. In July 25 meeting, we noted this probably cannot be resolved with just one simple solution. May need to look at different options for different scenarios
        i. Also likely to need to store/cache a user's Groups in UI layer, as some areas (e.g. Administrative) require knowledge of user group
2. **REST API Projections:**
    a. Early work begu l, this approach seems like a good direction, need volunteers to move it forward.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

        i. https://github.com/DSpace/Rest7Contract/issues/2 (discussion resumed by Andrea Bollini (4Science) could be relevant for the projection)

3. Initial Performance Testing from Chris.
    a. https://cwilper.github.io/dspace-perftest/
4. (REST Contract) Edit Homepage News: https://github.com/DSpace/Rest7Contract/pull/45
    a. Delayed until after Preview release. General agreement (in meeting on March 21, 2019) that storing HTML in metadata fields is not really ideal behavior. Metadata (from a librarian standpoint) tends to be free of format-related markup (as that allows for easier sharing, understanding of metadata. Currently Community & Collection homepage information is HTML-based and *is stored in metadata* that is appropriate for a minor subset of information (like the title) but it is better to move large/rich text to bitstreams.
    b. Proposal here is to consider storing HTML-based markup (for Site, Community & Collection homepages) in Bitstream(s) associated with the object in question. May allow for more CMS-lite behavior in the future
    c. Timeline for this is uncertain. Possibly in 7 or 8. May depend on how/whether it can be scoped.
5. Concurrency in DSpace 7 (or 8). What do we want to do when multiple editors are editing the same object? Needs further analysis regarding implementation details
    a. We've decided (in meeting on March 7, 2019) to use ETags to implement concurrency. REST Contract notes on ETags: https://github.com/DSpace/Rest7Contract#etags--conditional-headers
    b. ETags only update of the two fields match. If someone edits first, your edit would fail and you would get a fail response (422?)
    c. ETags seems to have broader support in other REST APIs. Recommended also by both Art and Andrea.

Notes

- REST API discussion: Reminder to all (self included) which endpoints can support which methods
    - Reviewed notes in agenda above, Per our Contract README: https://github.com/DSpace/Rest7Contract/blob/master/README.md#use-of-the-http-verbs-and-http-response-code
    - We may need to revisit the decisions in these two recent PRs:
        - https://github.com/DSpace/Rest7Contract/pull/79 (Community & Collection Logo). Should PUT/DELETE just use /bitstreams/<:logo_uuid>? (as Logos are Bitstreams!)
        - https://github.com/DSpace/Rest7Contract/pull/80 (Collection Item Templates). Should PATCH/DELETE just use /items/<:itemtemplate_uuid>? (as Templates are Items!
    - For Item Templates, there are many Item features which don't apply (Item Templates cannot have bundles/bitstreams – they are just metadata – and they cannot be withdrawn or made private)
        - Andrea noted these are long standing flaws (since DSpace 1.x) in the Java API layer. Currently, there's no Item Template object, so at the Java layer it's possible to add bitstreams to Item Templates or withdraw them...however, in DSpace 6.x and below we just didn't enable those features in the User Interfaces.
        - After discussion, we feel it might be good to create a separate `/itemtemplates/` endpoint which is specific to these Item Templates. However, if that proves difficult/complex, we could just use the `/items/` endpoint (as described) and delay these improvements to DSpace 8.x (as they are existing flaws in 6.x anyways, and we could similarly hide them from the UI layer in 7.x).
    - For Logos, they seem similar enough to Bitstreams that we should be able to use the /bitstreams endpoint
    - **ACTION: Tim will create tickets to suggest the necessary changes to these REST Contracts. Discussion can move to those tickets & (if needed) can be brought back up in future meetings**
- Revisiting our Code Review processes for PRs
    - Up until now, all PRs require two approvers
    - As we've seen, this provides good feedback, but can cause review/implementation delays if only one reviewer can be identified in a timely fashion
    - This issue was brought to Steering Group. Recommendation is to consider allowing for single reviewers for smaller / more obvious PRs, with option of that reviewer asking for an additional reviewer if they find something requiring additional feedback.
    - Some brainstorms from the team:
        - Should we default to **one** approver required per PR, but optionally request a second approver for more complex PRs?
            - Tim notes that REST Contracts likely need to default to two approvers. We've already seen issues where we accidentally approved contracts that didn't align entirely with our own design (see previous discussion). More eyes is really better when it comes to Contracts
        - Should we default to **two** approvers (like currently), but optionally flag specific PRs as needing just one approval (if they are minor, obvious, or simply quick bug fixes with ITs)?
        - Team agrees we should try an approach and see how it works. Can always change the approach if something isn't working.
        - Tim prefers the latter approach for now (two approvers is default, but can make exceptions for specific PRs). We can try that out for a few weeks and see what it's impact is. If it's too much work or not working well, we could try the one approver approach in future.
    - **ACTION: Tim will figure out a way to flag PRs as "one approval" needed. That way we can visually see which ones are smaller & easier to move along. Developers who add PRs to our review list can optionally flag their own PRs as "one approval" to note to Tim (and others) that they feel it just needs one person.**
- Other topics were moved to next week. Ran out of time.