2020-02-06 DSpace 7 Working Group Meeting

Date
06 Feb 2020 from 15:00-16:00 UTC

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

More connection options available at DSpace Meeting Room

Beta 1 & 2 Sprint : Feb 3-14

- Work towards finalizing the Beta1 tasks in DSpace 7 Release Plan spreadsheet. Goal is to have Beta1 released in mid-to-late Feb.
- Beginning work on Beta2 tasks (especially those with many dependencies)

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. DSpace 7 (Fund) Development Sprints. Establishing 2 Week Sprint Processes based on DSpace 7 Community Sprints.
  2. Scripts & Processes endpoint PR#2648; Tim has concerns about usage of Bitstreams for temporary files (like uploaded CSVs, SAF Zips, AIPs, etc)
  3. [DS-4427 - Getting issue details...](https://www.bitbucket.org) *STATUS*
  4. (Please add a topic)
- (15 mins) Planning for next week
  - Assigning PRs for Review

Attendees

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

7.0 Release Goals

These resources define the prioritization and general schedule we are working towards

- DSpace 7 Release Goals: overview of goals/timelines & beta release process
- DSpace 7 Release Plan spreadsheet: our planning spreadsheet which details which features are scheduled for each Beta release.

Current Work
Claim a Ticket!

If you do not have access in JIRA or GitHub to officially claim the ticket you wish to work on, contact Tim Donohue

- Available Angular UI Tickets
  - Open issues in 7.0beta1 milestone
  - Open issues in 7.0beta2 milestone
  - Open issues flagged as "Difficulty: low". These may be a good starting point for newer developers.
- Available REST API Tickets
  - Unassigned tickets scheduled for 7.0beta1
  - Unassigned tickets scheduled for 7.0beta2
  - General REST API tickets flagged as "low hanging fruit". These may be a good starting point for newer developers.
  - General REST API unassigned bug tickets. Again, might be a good starting point for newer developers.

PRs Needing Review

1. (REST Contract) Group and eperson management: https://github.com/DSpace/Rest7Contract/pull/41
2. (REST Contract) Versioning https://github.com/DSpace/Rest7Contract/pull/97
3. (REST) DS-4399 improving patch system framework Part 1 https://github.com/DSpace/DSpace/pull/2591
4. (REST) Initial Implementation of Resource Policies endpoint: https://github.com/DSpace/DSpace/pull/2604
5. (REST) Projections continued: https://github.com/DSpace/DSpace/pull/2625
6. (REST) Workflow step definitions https://github.com/DSpace/DSpace/pull/2646
7. (REST) Scripts & processes: importing and exporting csv's https://github.com/DSpace/DSpace/pull/2648
8. (REST) Shibboleth authentication https://github.com/DSpace/DSpace/pull/2651
9. (NEW) (REST) Collection item template updates https://github.com/DSpace/DSpace/pull/2656
11. (Angular) Shibboleth authentication (merge with REST PR #2651)
12. (Angular) Generic uri metadata https://github.com/DSpace/dspace-angular/pull/568
13. (Backend) Java 11 upgrade and Replace JMockit with Mockito (MERGE on/around Feb 13)
14. (Backend) URL configuration refactor (new "dspace.server.url" and "dspace.ui.url") (MERGE on/around Feb 13)

PRs Merged this week!

1. (REST Contract) Support for a trail by retrieving parent objects https://github.com/DSpace/Rest7Contract/pull/100
2. (REST) Default webui.submit.upload.required https://github.com/DSpace/DSpace/pull/2650
3. (Angular) Verify whether file uploads are mandatory https://github.com/DSpace/dspace-angular/pull/552
4. (Backend) DS-4425 Indexing Performance is very slow since refactor. ITs take twice as long https://github.com/DSpace/DSpace/pull/2658
5. (Backend) dspace.bat file: https://github.com/DSpace/DSpace/pull/2544

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
1. 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.

2. Work is ongoing, but has been started in these areas:
   a. Summary of ideas: REST Authorization

2. REST API Projections: **DS-3533** - Getting issue details... Status
   a. Work begun in https://github.com/DSpace/DSpace/pull/2547 (by Chris Wilper)
   b. Early work begun at https://github.com/DSpace/DSpace/pull/1847. Discussed in more detail in our Aug 22 meeting. Overall, this approach seems like a good direction, need volunteers to move it forward.
   c. 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. Delayed. 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.


6. 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

- Sprint schedule discussion. Decided on the following:
  - Two **required** check-ins per week (for all paid developers). One on Monday (via #dev-sprint on Slack) and one on Thursday (in weekly meeting or via #dev-sprint on Slack)
  - Daily, informal updates to #dev-sprint are also recommended whenever a developer hits a blocker, realizes work may take longer than expected, or requires a review/feedback to proceed.
  - In weekly meeting on Thursday, we will assign tasks for the following week. Atmire & 4Science Teams should come with a list of tasks they propose.
- Discussion of Scripts & Processes endpoint PR#2648
  - Tim has concerns about usage of Bitstreams for temporary files (like uploaded CSVs, SAF Zips, AIPs, etc)
  - Kevin will be responding to Tim’s feedback in PR
  - Andrea notes that they rarely do cleanup of bitstreams in their production systems. We’d need a way to cleanup temporary files **separate** from normal Bitstreams
  - Tim notes we’d also need temporary Bitstreams to be exempt from things like Event listeners. Do we need to subclass Bitstreams to something like “TemporaryBitstream” so these can be treated differently?
  - Will revisit this next week as needed and based on Kevin’s feedback.