

# 2019-08-22 DSpace 7 Working Group Meeting

- [Date](#)
- [Agenda](#)
- [Attendees](#)
- [Current Work](#)
  - [Tickets / PRs In Progress](#)
  - [PRs Needing Review](#)
  - [PRs Merged this week!](#)
  - [Blocked](#)
  - [Delayed / Needs Discussion](#)
- [Notes](#)

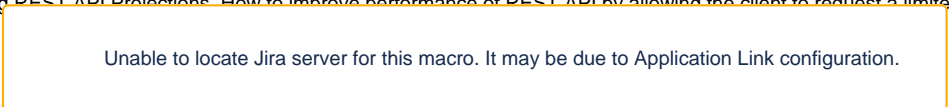
## Date

22 Aug 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. Revisiting [PEST API Projections](#). [How to improve performance of PEST API by allowing the client to request a limited response](#) (only returning
      - a. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
      - b.
2. Improved Estimation Strategy for Beta release. (Estimates should include time for discussion / code reviews, also should be done by multiple developers)
  - a. [Development Planning Spreadsheet](#) needs to be updated, so that we can determine what (outstanding) tasks need estimation.
  - b. Will be using [Wideband Delphi estimation technique](#)
- (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](#)
- [Heather Greer Klein](#)
- [Lieven Droogmans](#) - Out until Aug 29 meeting
- [Mark H. Wood](#)
- [Giuseppe Digilio \(4Science\)](#)
- [Ben Bosman](#)
- [Pascal-Nicolas Becker](#)
- [Chris Wilper](#)
- [Terrence W Brady](#)
- [Paulo Graça](#)
- [Dimitris Pierrakos](#)
- [Julius Gruber](#)
- [Laura Henze](#)

## Current Work

## Legend for status icons

★ = Highest Priority tasks (please prioritize these reviews/tasks over others). Currently this lists tasks/features that need to be completed for Preview Release.

★ = Priority for OR2019 (Preview #2 Release).

blocked URL = review done (this week), changes were requested.

blocked URL = 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. blocked URL (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. blocked URL (REST Contract) Group and eperson management: <https://github.com/DSpace/Rest7Contract/pull/41> (Waiting on updates from Ben Bosman )

## PRs Needing Review

1. (REST Contract) Collecting statistics - <https://github.com/DSpace/Rest7Contract/pull/63> (Dimitris Pierrakos, Andrea Bollini (4Science) blocked URL Feedback provided, Ben Bosman , ✓ Tim Donohue )
2. (REST Contract) Update bitstream properties and format <https://github.com/DSpace/Rest7Contract/pull/68> (✓ Tim Donohue, Andrea Bollini (4Science) blocked URL Feedback provided)
3. (NEW) (REST Contract) DS-4317 bundles in REST <https://github.com/DSpace/Rest7Contract/pull/72> (✓ Tim Donohue - minor tweaks suggested, Andrea Bollini (4Science) )
4. (NEW) (REST Contract) upload bitstream to archived item - update <https://github.com/DSpace/Rest7Contract/pull/71> (✓ Tim Donohue, Andrea Bollini (4Science) )
5. (REST) Pagination bug with withdrawn items: <https://github.com/DSpace/DSpace/pull/2406> (Dimitris Pierrakos , Ben Bosman - Feedback provided)
6. (REST) implement upload bitstream to archived item <https://github.com/DSpace/DSpace/pull/2473> (✓ Tim Donohue , Andrea Bollini (4Science)- blocked URL Feedback provided)
  - a. Questions on Bitstream Sequence ID. **MAKE READ-ONLY**
7. (NEW) (REST) Issue when community has multiple dc.title values <https://github.com/DSpace/DSpace/pull/2486> (blocked URL Tim Donohue , Andrea Bollini (4Science) )
8. (NEW) (REST) Oai harvesting setup <https://github.com/DSpace/DSpace/pull/2491> (Tim Donohue, Andrea Bollini (4Science))
9. (Angular) (Entities) Deleting relationships: <https://github.com/DSpace/dspace-angular/pull/402> (Paulo Graça - will test again, Tim Donohue )
10. (Angular) Move Item Component: <https://github.com/DSpace/dspace-angular/pull/335> (Giuseppe Digilio (4Science) - REREVIEW, Tim Donohue - REREVIEW)
11. (Angular) Item-Collection Mapper: <https://github.com/DSpace/dspace-angular/pull/348> ( Tim Donohue - REREVIEW, NEEDS SECOND REVIEWER)
12. (Angular) Shibboleth integration support: <https://github.com/DSpace/dspace-angular/pull/429> (Julius running into an error with 'yarn start' only) (Giuseppe Digilio (4Science) feedback provided, Fernando FCT/FCCN, Paulo Graça )
13. (Angular) Submission Miscellaneous fixes: <https://github.com/DSpace/dspace-angular/pull/432> (blocked URL Art Lowel (Atmire), Julius Gruber )
14. (Angular) UI Language Cookie <https://github.com/DSpace/dspace-angular/pull/443> (✓ Tim Donohue, Paulo Graça)
15. (Angular) Convert i18n files to JSON5 format <https://github.com/DSpace/dspace-angular/pull/439> (Tim Donohue, **NEEDS SECOND REVIEWER**)
16. (Angular) Search Performance optimizations #2 <https://github.com/DSpace/dspace-angular/pull/437> (Tim Donohue, Giuseppe Digilio (4Science))
17. (Angular) bitstream format registries <https://github.com/DSpace/dspace-angular/pull/455> (Tim Donohue , blocked URL Art Lowel (Atmire))
18. (New) (Angular) Fix the labels on edit collection and community pages <https://github.com/DSpace/dspace-angular/pull/459> (Giuseppe Digilio (4Science) , Tim Donohue )
19. Indirect entity refs during CSV import <https://github.com/DSpace/DSpace/pull/2471> (Ben Bosman - ⚠ reviewed and provided feedback, Tim Donohue )
20. (Backend) Solr 7 fixes for upgrading to DSpace 7 <https://github.com/DSpace/DSpace/pull/2393> (Chris Wilper? , **NEEDS SECOND REVIEWER**)

## PRs Merged this week!

1. (First PR merged goes here)

## Blocked



1. (Blocked PRs go here)

## Delayed / Needs Discussion

1. (REST) **Scripts & Processes endpoint**: <https://github.com/DSpace/Rest7Contract/pull/17>
  - a. How do we move this forward? In our Jan 24 meeting, Tim noted a risk of "scope creep" with this feature suggestion. This idea has been tabled since then.

- b. In [July 25 meeting](#), Atmire said they'd come back with notes on the proposed backend implementation.
2. **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 membership
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 if 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.
6. Improve/Re-enable End To End (e2e) Testing. Could there be opportunities to use Travis CI + Docker Compose for testing of Angular?? <https://github.com/DSpace/dspace-angular/issues/453#issuecomment-519672141>
  - Docs: <https://docs.travis-ci.com/user/docker/#using-docker-compose>
  - Blogpost on how to do it: <http://elliott.land/post/using-docker-compose-on-travis-ci>
  - An example travis.yml file: <https://github.com/Ortus-Solutions/docker-buildfiles/blob/master/.travis.example.yml>

## Notes

- Lots of individuals just returning from holidays. Welcome back!
- Quick update on Improved Estimation Strategy for Beta release.
  - [Heather Greer Klein](#) has been working on this while Tim was out on holiday. She has some initial spreadsheets nearly ready, and Tim will be reviewing today/tomorrow
  - Anticipate that estimation spreadsheets will be ready to share either on Aug 29 or Sept 5....once ready, we'll go through them in detail and get some estimators lined up, etc.
- Discussion: Revisiting REST API Projections
  - Relevant
    - 
    - 
    - Unable to locate Jira server for this macro. It may be due to Application Link configuration.
  - Main point
    - Performance will improve because REST APIs gathering less information (via DB & Hibernate), and UIs retrieving/parsing less information (smaller JSON)
  - Old PR (from Tom Desair, previously of Atmire) began some Projections work based on [Spring Data REST Projections](#)
    - <https://github.com/DSpace/DSpace/pull/1847>
    - Basic idea is that Spring Data REST provides a `@Projections` annotation that can be used to define named projections. In requests to the REST API, by default you still get all data back, but you can pass in a "projections=" parameter to limit the information you need based on one (or more) defined projections.
  - Will require changes **both** to REST API and Angular UI
    - [Art Lowel \(Atmire\)](#) notes that UI's current caching system may need some rethinking. All objects currently cached in UI layer. If some requests are only returning partial objects (e.g. metadata only), then the UI may need to "know" when it only has a partial object – so that it can make a new request for the full object when necessary. Ideally though, we obviously should figure out the REST API implementation first, then brainstorm how this affects the UI cache.
    - [Andrea Bollini \(4Science\)](#) notes that Tom's initial PR had a good approach of building off Spring Data REST Projections... **however**, as noted in [this PR comment](#), Andrea feels that the Projection itself should be using the "\*\*Rest" classes, like ItemRest.class, (instead of the Hibernate classes, like Item.class), as Projections are a REST API level concept...and should therefore be implemented at that level. Currently, it seems odd that they are bypassing the "\*\*Rest" classes to define projections on the Hibernate classes themselves
      - Tim notes point well taken. However, it sounds like this needs further investigation / thinking on how to change the implementation to use "\*\*Rest" classes. Perhaps we can do this in two stages? Start with Tom's current strategy just to get an initial implementation (that the UI can start to use & implement). Then, we can refactor to use the "\*\*Rest" Classes once we figure out a way to do so.
      - Andrea agrees. We should do the REST Contract first, then an initial implementation to build from. Will need to scope the idea of Projections to make this easier to approach.
  - Scoping Projections: What types of Projections do we need?
    - Andrea notes that Projections should be based on use cases, e.g. a "list" of objects (Items, Collections, Communities) should return less information about objects in that list than an Object page would return.
    - Tim suggests we start small. One initial projection.
    - **ACTION: Start with a "list" projection that ONLY returns: all metadata and one thumbnail (if exists)**
      - Others agree. Andrea notes this also seems like it would resolve our two outstanding bugs related to projections (DS-4306 and DS-4307, see above for links)
  - Next Steps:

- Need a REST Contract for Projections. This is already somewhat detailed in Tom's initial PR's description...just needs to be formalized into a REST Contract
    - Initial implementation will be limited to a projection(s) for the "list" use case (all metadata + thumbnail). This might be one "list" projection, or maybe two projections: "metadata only" projection and "thumbnail" projection.
    - Keep an eye out for other possible Projection use cases / needs...this "list" projection though seems to be the biggest need (and may provide massive performance improvements)
    - May then need to investigate using the "\*\*Rest" classes & refactor that initial implementation (see comments from Andrea above)
  - Who will start this work?
    - Ben notes that [Chris Wilper](#) might be interested. Will check with him though
    - **ACTION:** [Tim Donohue](#) will update JIRA tickets with what we decided today & then we can look to assign the ticket(s).
- Assigning PR Reviews
  - One discussion topic came up
  - *Questions on Bitstream Sequence ID in two PRs:* <https://github.com/DSPACE/Rest7Contract/pull/68> and <https://github.com/DSPACE/DSPACE/pull/2473>
  - Should Sequence ID still exist in DSpace 7? It seemed to be used as an alternative URL for bitstream downloads (in DSpace 6). Bitstreams could be downloaded via sequence ID (unique in an Item) or name, or both.
  - Sequence ID is misnamed. It's not defining a sequence (ordering) in terms of Bitstreams...it's also not managed by the database layer (not a database sequence). It's an integer defined (by the UI in DSpace 6) for each bitstream, and it must be unique within an Item.
  - Andrea notes that Sequence ID had another purpose in DSpace 6....it was a "persistent" Identifier for Bitstreams, especially since a Bitstream's name could be changed
  - Ben notes that in DSpace 7, the Bitstream's "persistent" Identifier is its UUID
  - **We need to keep Sequence IDs around in DSpace 7 to allow for redirects from old (DSpace 6 style) URLs to new URLs**
  - However, the way they were managed in DSpace 6 was odd. **In DSpace 7, we'll make Sequence IDs READ-ONLY at the REST API layer.**
  - Need a ticket to discuss whether Sequence IDs should remain a key feature (in which case they probably should be auto-assigned by the Database or Java API layer, they currently are not), or if they are only being kept for backwards compatible URLs (redirecting pre-DSpace 7 URLs to new URLs). [Tim Donohue](#) will create this ticket
  - **POST-MEETING UPDATE:** After further analysis, Tim realized that Sequence IDs are being managed by ItemService.update(). Therefore they should **not** be deprecated. Instead, Tim created a small PR to better document sequence IDs in the code / javadocs: <https://github.com/DSPACE/DSPACE/pull/2492/>