2017-08-03 DSpace 7 Working Group Meeting notes

Date
03 Aug 2017

Angular updates

Attendees

- Art Lowel (Atmire)
- Giuseppe Digilio (4Science)
- Tim Donohue
- Matheus Silva
- Nelson Torres

Notes

- Waffle Board: https://waffle.io/DSpace/dspace-angular
- Giuseppe’s PRs waiting for final approval/merger:
  - https://github.com/DSpace/dspace-angular/pull/131
  - https://github.com/DSpace/dspace-angular/pull/132
- Art’s ngrx-platform upgrade is backlogged. It wasn’t working cause of a known bug in ngrx-platform. Will pick back up when bug is fixed
  - See https://github.com/DSpace/dspace-angular/issues/86 for more info
- Art will be working on browse by date / alphabet selector for collection browse options
- Also created tasks for upcoming Search work
  - First task is SearchService with mock data (which can be done even prior or in parallel to Search REST API work): https://github.com/DSpace/dspace-angular/issues/134 (Ready to be claimed)
  - Simple Search component (depends on 134): https://github.com/DSpace/dspace-angular/issues/136
  - Once REST API is ready, we’ll replace mock data with real data. Ticket for that at https://github.com/DSpace/dspace-angular/issues/135
  - Feedback on this selection is welcome! (from anyone)
  - In meantime, Giuseppe is creating a basic proof of concept to ensure this will meet our needs.
- Noted that our ts-lint settings are currently a bit strict (throwing warnings for this new module). We may need to look into possibly loosening the ts-lint requirements, or finding a way to disable on third-party modules. Will bring to Slack as necessary.

REST updates

Attendees

- Terrence W Brady joined
- Mark H. Wood joined

Notes

- Andrea still out on holiday
- Tim gave an update on DSpace 6 efforts. It’s very important to note that we’ve all realized that DSpace 6’s behavior with regards to database connections is significantly different from how DSpace 5 behaved. The behavior change came with the move to Hibernate and our current Hibernate configs.
  - The DSpace 7 REST API team needs to be aware of this difference in behavior as they develop new Java code.
  - Behavior is described at DevMtg 2017-08-02 (see bullet #2), and also discussed in great detail in the logs of that day’s meeting. Copied description/notes below:
    - In DSpace 5, each "Context" established a new DB connection. Context then committed or aborted the connection after it was done (based on results of that request). Context could also be shared between methods if a single transaction needed to perform actions across multiple methods.
    - In DSpace 6, Hibernate manages the DB connection pool. Each thread grabs a Connection from the pool. This means two Context objects could use the same Connection (if they are in the same thread). In other words, code can no longer assume each new Context() is treated as a new database transaction.
    - Currently, the DSpace 6 team is stabilizing the codebase based on this behavior (a 6.2 release coming very soon with fixes to Context & Hibernate)
    - In discussion is what we want to do for DSpace 7 API. Already a few ideas proposed. No formal plans yet.
      - Change "Context" to work how it used to (create a new connection per Context object), reconfiguring Hibernate as needed
      - Refactor "new Context()" into something like "Context.getCurrentContext" and "Context.createNewContext" (where former is an existing connection, and latter is a new one)
      - Or, potentially pull apart the Context object altogether, and split it into several objects based on the various jobs/tasks (e.g. a session-based object for things in HTTPSession, a request-based object for current HTTPRequest, etc etc)
    - Andrea looking for more feedback on Authentication for REST API: https://github.com/DSpace/Rest7Contract/issues/10
    - Tom looking for feedback/final decision on Search endpoint design: https://github.com/DSpace/Rest7Contract/pull/9