2022-07-28 DSpace 7 Working Group Meeting

Date

28 Jul 2022 from 14:00-15:00 UTC

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

- More connection options available at DSpace Meeting Room

7.4 Release Plan

Release Schedule:

The 7.4 Release will concentrate on maintenance (bug fixes, usability fixes and improvements), but will accept donated features from anyone in the community. Features which missed the 7.3 release will also be accepted.

- *Monday, August 1 (Donated Feature Notification Deadline):* Any community members who wish to donate a feature to this release must notify Tim Donohue by this date (either via email, Slack or GitHub). The DSpace 7 team will then provide feedback on whether it will be possible to include this feature in the release (based on team member availability and the size of the feature). Early notifications are more likely to get included in the release.
- *Friday, August 26 (Feature PR Creation Deadline):* PRs should be created by this date if they are to be reviewed in time for the release. Please note there is no guarantee that a PR will be included if it is created by this date. Larger PRs are recommended to be created earlier, as that makes it more likely they can be reviewed in time for inclusion. *(Smaller bug fixes are welcome anytime)*
- *Friday, September 9 (Donated Feature PR Merge Deadline):* All new *donated & approved* feature PRs should be fully reviewed & merged by this date. Donated features are those which anyone donates back to the DSpace project. *(Donated PRs which are improvements or bug fixes are accepted after this deadline)*
- *Friday, September 16 (Final PR Review/Test Deadline):* All code reviewers or testers should submit their feedback by this date. Code reviews must be constructive in nature, with resolution suggestions. Any code reviews submitted AFTER this date will be considered non-blocking reviews. **NOTE: Larger PRs or donated PRs will have their own deadlines established for PR creation, review and merger.** This deadline only applies for PRs with no other deadline established.
- *Friday, September 30 (PR Merge Deadline):* All PRs should be merged by this date. (Note: bug fixes can still get in after this deadline, as long as they are small or important)
- *Week of October 3:* Internal / Early release goal. If possible, we'd like to release 7.3 this week.
- *Monday, October 10:* Public Release Deadline. 7.4 will be announced/released by this date.

Ongoing/Weekly tasks:

- Tackle/Claim issues on 7.4 board (starting with "high priority")
- Review/test all PRs assigned to you for review/testing: https://github.com/pulls/review-requested (*Prioritize reviews of "high priority" PRs first)*

Upcoming holidays for team members (Please add your holidays to this list)

- Tim : On Holiday Aug 10-16
- 4Science team (main version 7 developers)
    - Andrea: On Holiday July 25 to Aug 5
    - Corrado: no holidays over the summer
    - Giuseppe: On Holiday Aug 1- 19
    - Davide: On Holiday July 25 to Aug 5
    - Luca: On Holiday Aug 8 - 22
    - Misha: On Holiday July 22 to Aug 5
    - the rest of the dev team (> 10) have some weeks across July/Aug but there is always presence of REST, Angular and Architecture specialist
- Atmire team
    - Lieven: July 25 - Aug 12
    - Art: July 29 - Aug 5
    - Ben: July 25-29 and Aug 8-12
- The Library Code team
    - Pascal: Aug 01 - Aug 23
- Paulo Graça (FCT|FCCN)
    - Aug 1st-5th
    - Aug 18th-31st

Agenda

- (30 mins) General Discussion Topics
    1. (Please suggest a topic! If no topics are suggested, we will primarily go through the project board to ensure PRs are moving along.)
- (30 mins) Planning for next week
    - Review the Backlog Board - Are there any tickets here stuck in the "Triage" column? We'd like to keep this column as small as possible.
    - Review the 7.4 Project Board - Assign tickets to developers & assign PRs to reviewers.
        - Paid (by DSpace project) developers must keep in mind priority. If new "high" or "medium" priority tickets come in, developers should move effort off of "low" priority tasks.
        - Volunteer developers are allowed to work on tickets regardless of priority, but ideally will review code in priority order

Attendees

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

Current Work

Project Board

DSpace 7.4 Project Board: https://github.com/orgs/DSpace/projects/18

To quickly find PRs assigned to you for review, visit https://github.com/pulls/review-requested (This is also available in the GitHub header under "Pull Requests Review Requests")

Maintenance tasks we feel should be included in 7.4

(When selecting PRs to work on, please consider these higher level categories)

- Curation Tasks are not very usable from the Admin UI: their results are only sent to dspace.log https://github.com/DSpace/dspace-angular/issues/1322 and they cannot restore deleted objects https://github.com/DSpace/DSpace/issues/7956 and cannot run on items https://github.com/DSpace/DSpace/issues/8384
- Resource Policy edit form is confusing & have usability issues. See for example https://github.com/DSpace/dspace-angular/issues/1704 and https://github.com/DSpace/dspace-angular/issues/644 (which is not policy specific but is reproducible when editing a policy)
- Improve password requirements/management. See for example these tickets https://github.com/orgs/DSpace/projects/18?card_filter_query=label%3A%22authentication%3A+password%22
- Provide a way to manage/cleanup Processes, e.g. https://github.com/DSpace/DSpace/issues/7974 and https://github.com/DSpace/dspace-angular/issues/1343 and https://github.com/DSpace/dspace-angular/issues/1660
- Issues with file download from submission/workflow, see https://github.com/DSpace/dspace-angular/issues/1644 and https://github.com/DSpace/DSpace/issues/8378
- General caching issues, e.g. https://github.com/DSpace/dspace-angular/issues/644 and https://github.com/DSpace/dspace-angular/issues/742 and https://github.com/orgs/DSpace/projects/18?card_filter_query=label%3A%22performance+%2F+caching%22
- Statistics display issues, e.g. https://github.com/orgs/DSpace/projects/18?card_filter_query=label%3A%22component%3A+statistics%22
- (Possibly) Restrict system managed metadata fields from editing: https://github.com/DSpace/DSpace/issues/8299

New Feature development process for 7.4

- **For brand new UI features**, at a minimum, the UI ticket should contain a description of how the feature will be implemented
    - If the UI feature involves entirely new User Interface interactions or components, *we recommend mockups or links to examples elsewhere on the web*. (If it's useful, you can create a Wiki page and use the Balsamiq wireframes plugin in our wiki)
    - Feature design should be made publicly known (i.e. in a meeting) to other Developers. **Comments/suggestions must be accepted for TWO WEEKS, or until consensus is achieved** *(whichever comes first)***.** After that, silence is assumed to be consent to move forward with development as designed. (The team may decide to extend this two week deadline on a case by case basis, but only before the two week period has passed. After two weeks, the design will move forward as-is.)
    - This does mean that if a UI feature is later found to have design/usability flaws, those flaws will need to be noted in a bug ticket (to ensure we don't repeat them in other features) and fixed in follow-up work.
- **For brand new REST features (i.e. new endpoints or major changes to endpoints)**, at a minimum we need a REST Contract prior to development.
    - REST Contract should be made publicly known (i.e. in a meeting) to other Developers. **Comments/suggestions must be accepted for TWO WEEKS, or until consensus is achieved** *(whichever comes first)***.** After that, silence is assumed to be consent to move forward with development. (The team may decide to extend this two week deadline on a case by case basis, but only before the two week period has passed. After two weeks, the design will move forward as-is.)
    - This does mean that some REST features may need future improvement if the initial design is found to later have RESTful design flaws. Such flaws will need to be noted in a bug ticket (to ensure we don't repeat them in other features) and fixed in follow-up work.
    - *REST API Backwards Compatibility support*
        - During 7.x development, we REQUIRE backwards compatibility in the REST API layer between any sequential 7.x releases. This means that the 7.1 REST API must be backwards compatible with 7.0, and 7.2 must be compatible with 7.1, etc.
            - However, deprecation of endpoints is allowed, and multi-step 7.x releases may involve breaking changes (but those breaking changes must be deprecated first & documented in Release Notes). This means that it's allowable for the

7.2 release to have changes which are incompatible with the 7.0 release, provided they were first deprecated in 7.1. Similarly, 7.3 might have breaking changes from either 7.1 or 7.0, provided they were deprecated first.

- After 7.x development, no breaking changes are allowed in minor releases. They can only appear in major releases (e.g. 7.x8.0 or 8.x9.0 may include breaking changes).

Issue Triage process for 7.4

- **Overview of our Triage process:**
  1. *Initial Analysis*: Tim Donohue will do a quick analysis of all issue tickets coming into our Backlog Board (this is where newly reported issues will automatically appear).
  2. *Prioritization/Assignment*: If the ticket should be considered for this release, Tim Donohue will categorize/label it (high/medium/low priority) and immediately assign to a developer to further analysis. Assignment will be based on who worked on that feature in the past.
     a. "high priority" label = A feature is badly broken or missing/not working. These tickets must be implemented first, as ideally they *should be resolved* in the next release. (Keep in mind however that priorities may change as the release date approaches. So, it is possible that a "high priority" ticket may be rescheduled if it is a new feature that cannot fit into release timelines.)
     b. "medium priority" label = A feature is difficult to use, but mostly works.. These tickets *might* be resolved prior to the next release (but the release will not be delayed to fix these issues).
     c. "low priority" label = A feature has usability issues or other smaller inconveniences or a non-required feature is not working as expected. These tickets are simply "nice to have" in the next release. We'll attempt to fix them as time allows, but no guarantees are made.
  3. *Detailed Analysis*: Developers should immediately analyze assigned tickets and respond back within 1-2 days. The developer is expected to respond to Tim Donohue with the following:
     a. Is the bug reproducible? (If the developer did not understand the bug report they may respond saying they need more information to proceed.)
     b. Does the developer agree with the initial prioritization (high/medium/low), or do they recommend another priority?
     c. Does the bug appear to be on the frontend/UI or backend/REST API?
     d. Does the developer have an idea of how difficult it would be to fix? Either a rough estimate, or feel free to create an immediate PR (if the bug is tiny & you have time to do so).
     e. Are you (or your team) interested in being assigned this work?
  4. *Final Analysis:* Tim Donohue will look at the feedback from the developer, fix ticket labels & move it to the appropriate work Board. If it is moved to the Project Board, then the ticket may be immediately assigned back to the developer (if they expressed an interest) to begin working on it.
     a. If the ticket needs more info, Tim Donohue will send it back to the reporter and/or attempt to reproduce the bug himself. Once more info is provided, it may be sent back to the developer for a new "Detailed Analysis".

Notes