2019-01-17 DSpace 7 Working Group Meeting

Date
17 Jan 2019 from 15:00-16:00 UTC

Location: https://duraspace.zoom.us/my/dspace (Meeting ID: 502 527 3040).
  ▪ More connection options available at DSpace Meeting Room

Agenda

▪ (5 mins) General development / planning updates (Tim)
  ▪ DSpace 7 at OR2019 - Our proposals are in! Thanks all!
  ▪ Preview Release schedule discussion with Steering Group
    ▪ Tim will schedule a meeting of core contributors (DuraSpace, Atmire, 4Science) to discuss upcoming release schedule,
      brainstorm ways to speed up development/review processes and better stay on schedule.
▪ (20 mins) Unblocking our REST Contract PRs
  ▪ Spring Data REST "text-uri/list" to manage links between objects. Comments from Andrea Bollini (4Science) on #rest-api Slack Channel
    (yesterday).
    ▪ https://github.com/DSpace/Rest7Contract/pull/34 (alternative approach: https://github.com/DSpace/Rest7Contract/pull/42)
    ▪ https://github.com/DSpace/Rest7Contract/pull/35 (alternative approach: https://github.com/DSpace/Rest7Contract/pull/44)
▪ (10 mins) Quick updates on Angular UI tickets and/or PRs (Art)
▪ (10 mins) Quick updates on REST API tickets and/or PRs (Andrea)
▪ (10 mins) General Discussion Topics
  ▪ Planning for 7.0 Preview Release (end of January)
  ▪ Development Planning Spreadsheet
  ▪ Discussion of deletion of EPeople for GDPR compliance (from Pascal)
  ▪ Pending merger of: https://github.com/DSpace/DSpace/pull/2229. NOTE: item.getSubmitter() will return null if EPerson deleted.
  ▪ DSpace 7 installation/customization process. Documenting our plan/goal for 7.0
    ▪ Updates on "One Webapp Backend" PR: https://github.com/DSpace/DSpace/pull/2265 First example of a webapp (SWORDv1)
      that fully integrated & configurable.
    ▪ Also updated docs at: DSpace Backend as One Webapp
▪ (5 mins) Agenda items for next meeting?

Attendees

▪ Art Lowel (Atmire)
▪ Andrea Bollini (4Science)
▪ Tim Donohue
▪ Lieven Droogmans
▪ Mark H. Wood
▪ Giuseppe Digilio (4Science)
▪ Ben Bosman
▪ Pascal-Nicolas Becker
▪ Chris Wilper
▪ Terrence W Brady
▪ Pablo Prieto

Unavailable

Notes

(Notes below copied from last meeting. Details will be updated during this meeting.)

▪ General Updates (Tim)
  ▪ Tim thanks all for efforts on getting OR2019 presentations in. Full list at DSpace 7 at OR2019
  ▪ Notes discussion at Steering Group (yesterday) that we won't make current Preview Release goal (end of Jan). Suggestion out Steering
    was to have a Planning meeting of core contributors (DuraSpace, Atmire, 4Science) to discuss how to update the schedule,
    and brainstorm ways to speed up development/review processes and better stay on schedule.
    ▪ Tim will send out a Doodle poll this week to hopefully schedule meeting sometime next week
▪ Unblocking REST API PRs
  ▪ Tim summarized discussion from last meeting & feedback from Andrea in #rest-api Slack channel.
  ▪ Lots of back and forth for discussion & clarification occurred. This cannot be accurately documented, so I've summarized the decision / main
    points below.
  ▪ Final decision on managing relationships/links via PUT/POST (this is a summary of several back & forth discussions, which finalized
    in these main points)
    ▪ In general, we should be aiming to align with Spring Data REST best practices (where they exist)
    ▪ In Spring Data REST, the best practice to create/update/manage associations (i.e. links) between resources (DSpace Objects)
      is to do the following:
• First, create all resources (objects) the need to be linked (via appropriate POST requests), if they are not already created
• Second, establish the relationship between resources by sending a PUT request to the main object’s association endpoint, where the following is true
  • The request body of the request should be a list of URIs to link/associate with the main object
  • The request content type should be "text/uri-list".
• Better examples of this recommendation exist here https://www.baeldung.com/spring-data-rest-relationships
• Relationships on Object Creation (usually POST)
  • For object creation (POST of new object) in DSpace, we usually need to send the whole object representation (JSON) in the request body. For DSpace REST API, we’ve chosen to always send JSON in the request body (Content Type: application/json or application/hal+json)
  • There are some object types in DSpace that cannot be created without a Parent Object (i.e. without a link /association to another object). Such examples include Items (which cannot be created without a Collection) and Collections (which cannot be created without a Community)
    • In this scenario, we cannot align directly with Spring Data REST best practices as the POST request (to create the object) must also specify the link/association (with the Parent Object). The alignment is impossible in this scenario because we cannot send a single POST request that sends both JSON (application/json) content and "text/uri-list".
    • Therefore, in this scenario, we have chosen to specify the link/association via a query parameter in the POST request. So, for example, a new Collection is created via a single POST request that passes the (new) Collection data in the request body (as JSON) and the Parent Community link as a parameter (e.g. ?parent= [uuid]).
  • For any object types that can be created without a Parent Object, we should follow Spring Data REST best practices. The object should be created first (via POST), then relationships established (via a PUT).
• Updating Relationships for existing Objects (via PUT)
  • Whenever we are updating links/associations between objects (via PUT requests), both resources (objects) already exist. Therefore, we can align with Spring Data REST best practices and send a PUT request (whose request body is the list of URIs to link to, and whose request content type is "text/uri-list").
  • SIDENOTE: If we tried to apply the same query parameter exception to this scenario, the result would be a PUT request with an empty request body (as the link/association would be sent as a query parameter).
    • This approach is generally frowned upon in REST best practices, as both POST and PUT are expected to specify their data in either the request's body or headers. So, sending a POST/PUT with no body would be equivalent to sending a request to create a new empty object. In other words, the main data of a PUT or POST request is expected to be sent in the body of the request – so, a request with no body is a request that sends no data.

• We did not get to any of the below topics (as discussion went long). Below notes are covered over from last week. Any updates or requests for more feedback should be made on Slack.
• Angular Team Updates (Art)
  • Merged PRs:
    • In Progress tickets / PRs:
  • Tickets / PRs requiring review:
    • https://github.com/DSpace/dspace-angular/pull/349 (Highest Priority)
    • https://github.com/DSpace/dspace-angular/pull/350 (based on same branch as 349)
    • https://github.com/DSpace/dspace-angular/pull/348 (based on same branch as 348)
    • https://github.com/DSpace/dspace-angular/pull/347 (end goal to support PATCHes of Metadata)
    • https://github.com/DSpace/dspace-angular/pull/342 (Item Actions - NEED Demo REST API Update)
    • https://github.com/DSpace/dspace-angular/pull/335 (BLOCKED by REST Contract discussions about Move Item)
    • https://github.com/DSpace/dspace-angular/pull/339 (Merge conflicts need fixing)
  • Submission/Workflow PR: https://github.com/DSpace/dspace-angular/pull/279
  • Giuseppe says this will be ready for final review by Monday, January 14. He will ping us
  • ACTION: Anyone who has added past comments (that are now fixed), please go in and resolve your past comments once you've verified the fix. That'll help us figure out what is outstanding (if anything)
• REST Team Updates
  • Open PRs: https://github.com/DSpace/DSpace/pulls?q=is%3Apr+is%3Aopen+label%3A%22REST+API+v7%22+sort%3Aupdated-desc
  • Merged PRs:
  • In Progress tickets / PRs:
  • Tickets / PRs requiring review:
    • REST Contract tickets blocking other work (High Priority to get unblocked)
      • Request to use Spring Data REST "text-uri/list" to manage links between objects (Resources: https://docs.spring.io/spring-data/rest/docs/current/reference/html/#repository-resources.association-resource and https://www.baeldung.com/spring-data-rest-relationships)
        • https://github.com/DSpace/Rest7Contract/pull/34
        • https://github.com/DSpace/Rest7Contract/pull/35
        • https://github.com/DSpace/Rest7Contract/pull/41
      • ACTION: Tim Donohue will reach out to Andrea Bollini (4Science) to understand whether we can move forward with parameters for now, and update this contract/implementation later. It’s unclear to the team why these three tickets are blocked for Spring Data REST alignment, while others that involve links/associations (e.g. PR#37 [Item CRUD] and PR#35 [Metadata Registry]) were approved.
        • Should we consider fixing all these implementations to better align with Spring Data REST recommendations (perhaps in a future Ticket/PR)?
      • ACTION: We need to better document the requirements / best practices for our REST API. Currently alignment with Spring Data REST is not mentioned in our REST Contract README: https://github.com/DSpace/Rest7Contract/blob/master/README.md
    • Updating Collection/Community contracts (Requires re-review)
      • Feedback from Andrea, but needs re-review

    • https://github.com/DSpace/Rest7Contract/pull/33
• Metadata as a Map *(Requires re-review)*
  - REST Contract: [https://github.com/DSpace/Rest7Contract/pull/39](https://github.com/DSpace/Rest7Contract/pull/39)
  - Contract has feedback from Andrea & Approved by Tim. Waiting for rereview from Andrea Bollini (4Science)
  - JIRA: [DS-4107](https://github.com/DSpace/DSpace/pull/2287) - Represent metadata as a map instead of a list **CLOSED**
  - REST Implementation: [https://github.com/DSpace/DSpace/pull/2287](https://github.com/DSpace/DSpace/pull/2287)
  - Angular Updates: [https://github.com/DSpace/dspace-angular/pull/347](https://github.com/DSpace/dspace-angular/pull/347)

• Metadata Registry (8.1) - **Needs Reviews**
  - JIRA: [DS-4109](https://github.com/DSpace/DSpace/pull/2287) - Rest api metadata registry CRUD **CLOSED**
  - (Merged) Contract: [https://github.com/DSpace/Rest7Contract/pull/38](https://github.com/DSpace/Rest7Contract/pull/38)
  - REST: [https://github.com/DSpace/DSpace/pull/2291](https://github.com/DSpace/DSpace/pull/2291)

• Administrative Edit Item (metadata, bitstreams, assign roles) (7.13) - **Needs Reviews**
  - JIRA: [DS-4406](https://github.com/DSpace/DSpace/pull/2287) - Rest api item CRUD **CLOSED**
  - (Merged) Contract: [https://github.com/DSpace/Rest7Contract/pull/37](https://github.com/DSpace/Rest7Contract/pull/37)
  - REST: [https://github.com/DSpace/DSpace/pull/2290](https://github.com/DSpace/DSpace/pull/2290)

• *(Did not discuss in detail)* Discussion of EPerson Deletion *(Tim again asked that all write in your feedback on DS-4306 ticket)*
  - Ticket: [DS-4036](https://github.com/DSpace/DSpace/pull/2229) - Delete EPersons in DSpace **CODE REVIEW NEEDED**
  - PR: [https://github.com/DSpace/DSpace/pull/2229](https://github.com/DSpace/DSpace/pull/2229)
    - ACTION: ALL will add comments/thoughts to DS-4036 so we can finalize a decision on whether “null” EPersons are OK or not.
  - *(Did not discuss)* DSpace 7 installation/customization process.
    - Updated PR could use more eyes: [https://github.com/DSpace/DSpace/pull/2265](https://github.com/DSpace/DSpace/pull/2265)
    - See also early docs at: DSpace Backend as One Webapp

• Development planning/updates in [Development Planning Spreadsheet](https://github.com/DSpace/DSpace/pull/2229)
• The Next Meeting will be Thurs, Jan 24 at 15:00UTC (10:00am EST) in DSpace Meeting Room