2015-09-11 - Fedora API Extensions Meeting

Dial In Details

Date: Friday September 11, 2pm EDT (-4 UTC)
- U.S./Canada toll free: 866-740-1260, participant code: 2257295
- International toll free: http://www.readytalk.com/intl
- Use the above link and input 2257295 and the country you are calling from to get your country's toll-free dial-in number
- Once on the call, enter participant code 2257295

Meeting Goals

Shared understanding of
- the discussion surrounding the validation use case
- interactions of API-X effort with other Fedora efforts (like Hydra, Islandora, Async storage)

Determine next steps

Attendees

Agenda

1. Summarize, update this group on potential connections with other Fedora efforts
   a. We were mentioned by William G. Cowan in a recent thread soliciting participation in the Asynchronous storage effort
   b. Another thread discusses some integration patterns in the context of Hydra, Islandora, and Fedora. There seem to be some use cases that overlap nicely with API-X.
   c. (If any of these groups attend the call, I'd encourage a free-flow exchange of how these efforts could work together)
2. Summarize the deep dive that has been occurring on the AIC validation use case (see the comment thread on the page)
   a. Roles of the components in the architecture
3. Review (lightly!) the existing use cases and elicit the roles and brief descriptions
4. Next Steps
   a. Propose more time (two more weeks) for use case evaluation and discussion
   b. allows more use cases to surface
5. When do we think it makes sense to assign roles to people (Stakeholders, Designers, Developers)?
6. Any Other Business
   a. Elliot away: 9/28 - 10/28
   b. Andrew will be traveling for the next call (9/25) so he won't be participating.

Related Resources

Design Page (with use cases outline)
Use Cases Parent Page
Previous meeting agenda, including minutes

Minutes

- Word around the community seems to be spreading
- Elliot looked at Async Storage effort's wiki pages, REST API page. It looked very congruent with API-X use cases and intent
  - It would be reasonable for members of this group to look at their use cases
  - Aaron C: Some use cases are a good fit, some may not be. Seems like the best fit are for interactions that are not expressed easily in LDP or Fedora's APIs
  - Amherst has use cases related to putting objects into Glacier, completely external to Fedora
  - Andrew: There was a module (in fcrepo-labs?) that routed requests to external stores. Would that be useful?
    - This refers to Modeshape configuration for different binary stores, projection over filesystems
    -Modeshape can create separate buckets for binary content that can be routed to different binary stores
    - Fedora provides hints to Modeshape containing name of store to specify routing
  - Is one possible pattern for an API-X extension to intercept request, perform storage steps, and ingest a Fedora resource containing pointer to persisted location?
    - Amherst has use case similar to this (e.g. with Batch ingest). It makes sense to write them down
- The approach of this API-X team to take the initiative to understand other efforts in the community, then perform outreach seems like it could be a successful pattern
  - Seems to be a high degree of awareness of current efforts within this team. Keep it up.
Hydra, Islandora recently floated thoughts on the nature of their interaction with Fedora.

- May be interesting to consider how these interactions would occur if API-X existed
- Common example is complex ingest
  - Fair amount of work on both ends related to ingest workflows
- Unknown User (daniel-dgi) and Unknown User (escowles@ucsd.edu) spoke of similar identifier/URI mapping problems on yesterday's Fedora tech meeting, where business objects in Hydra or Islandora have identifiers that need a lookup to correspond to Fedora resources, and there isn't always a 1:1 mapping
- Where it make sense to consolidate functionality or define common APIs in those communities (as discussed in e-mail threads), could API-X be of value?
  - Don't want any project to feel like they're losing anything if functionality is contained in (or exposed by) an API-X extension, nor use API-X at all if it doesn't make sense
  - Ideally their needs could inform design of API-X to make it a useful tool to them
- Should somebody be ambassadors to Islandora/Hydra communities, at least until the API-X effort has progressed to the point where it's ready for broader conversation?

Validation use case spontaneously evolved into a deep dive.

- Has tested the boundaries of what API-X needs to do
- Highlights from Validation comments:
  - There is a synchronous/filtering use case where all requests to Fedora resources are filtered through validation extension; deny or allow request based on validity
  - There is an asynchronous/service use case where Fedora resources expose a service (similar to fcr:fixity) for producing validation reports
  - Lots of discussion on how API-X knows when to invoke validation extension in synchronous case
    - Globally, for every request?
    - Wherever there is some marker
      - Could this marker be an rdf:type (e.g. cma:validatable)?
      - Would validation extension or API-X need to be aware of hierarchical nature of Fedora resources?
  - Separation of concerns: What does API-X need to know what does validation extension need to know?
  - How would extension implementations interact with Fedora? HTTP+LDP? A client library?
    - forepo-kernel seems too closely tied to JCR, modeshape to allow completely external use cases (i.e. from a separate VM or host)
- Stefano: There seem to be three validation use cases really:
  1. Global validation across the board; do not even persist an object if invalid
  2. Persist invalid resources and do validation later
  3. Validation can be turned on/off for parts of the repository
  - We need to come up with a list of exactly what we want to distill from our ever growing list of use cases
    - Express/visualize interactions between clients, extensions, and services (Fedora, Indexes, Storage)?
    - Roles and responsibilities, e.g. what does API-X do vs extension?
    - What API-X requirements are implied by each use case?
    - What is the API-X value proposition for each use case? Why would it make sense to implement an API-X extension to implement a use case, rather than some other pattern?

Action item: **Aaron Birkland**: Create initial list of “what to distill from use cases”. We’ll debate and ratify over the next two weeks

Action item: **Unknown User (acoburn)**: Write down some Amherst College use cases regarding async storage

Action item: **Stefano Cossu**: Split validation use case into the three use cases mentioned above. The deep dive validation page would be too hard to read/understand going forward

Let's wait a little while longer to define roles (e.g. Stakeholder, Developer, etc). Need to understand use cases more, see if there is more interest from community

Next meeting 9/25, 2 PM EDT (-4 UTC). **Andrew Woods** will not be there due to travel conflict.