

2014-10-08 - Fedora Technical Working Group

Time/Place

- Time: 12:00pm Eastern Daylight Time US (UTC-4)
- Place: Google-hangout, https://plus.google.com/hangouts/_/event/c1glu6soq43r1rr6ou17qtobug8

Attendees

- Benjamin Armintor
- Chris Beer
- Daniel Davis
- Declan Fleming
- Unknown User (escowles@ucsd.edu)

Agenda

- Review assessment progress
 - REST API specification
 - Performance
 - Preservation-worthiness
 - Clustering
- Discussion: removing functionality not ready for production

Discussion

Assessment:

- Preservation: Esme: ingest working, scaling up -- filed a bug for performance of meadata updates on a federated filesystem which is being worked this sprint.
- REST API: Ben: no update other than LDP paging discussions, but still hopeful of wrapping this up this month.
- Performance: Dan: populating scenarios, with variations on ingest/access, concurrency, different file sizes, etc. Will work with Kevin Clarke, who is working on Grinder this sprint.

Functionality Discussion:

- Chris: Doing LDP alignment work and touching a lot of code, found several "soft spots" where code isn't ready for 4.0 release.
- Features proposed for removal from 4.0 release:
 - Batch API: good idea for possible performance improvement, but not being used by anyone yet.
 - Field Search: promising feature, could be used instead of external search engine
 - Ben: inclined to view SPARQL as better defined, better match to LDP
 - Chris: there are SPARQL fulltext extensions (e.g. <https://jena.apache.org/documentation/larq/>), but not ready for 4.0 either.
 - ID Minting: REST API support for minting identifiers was mostly added for F3 parity, really a separate concern from the repository
 - Ben: more interested in having small and well-defined API going forward than replicating F3 functionality
 - Esme: having pluggable auto-generated ids is the important feature
 - Repository-wide SPARQL updates (fcr:properties): PATCH support makes this redundant
 - Namespace management (fcr:namespaces): there is a need for namespace management, but this implementation isn't necessarily the right approach
 - Sitemaps:
 - Chris: important, but implementation is flawed, should be part of ResourceSync discussions
 - Ben: we need to support sync, but not a special API
 - Workspaces: good idea, with some use cases, but not working well
 - Ben: another JCR feature, multi-tenancy desired by vendors
 - Esme: Need to get engagement on this, refocus on use case instead of JCR functionality
 - Audit:
 - Dan: another case where use case is important, and needed for authenticity -- need central audit log, with object view
 - Chris: implementation doesn't satisfy use case, not heavily used for 3.x
 - Dan: needs to be on roadmap, will be increasingly important for 3.x migrations
 - DC Generator: could be part of OAI-PMH functionality, not required for 4.0
 - Policy-driven storage: Esme: could be more relevant now that Jersey 2 upgrade allows better async support
- Features in need of attention:
 - Versioning: used by Penn State (migrating versions from F3)
 - Current options are confusing
 - May want to revise model for 4.1
 - Ben: Memento could be a good fit
 - Transactions: interactions of transactions with other features, which aren't transaction-aware
 - Chris: Session / Transaction interactions in particular are hacky
 - Ben: maybe Jersey 2 cleanup could help this
 - JAX-RS cleanup: Ben: there were session injection things that didn't work before that we can revisit now
 - RDF: need real-world testing of different formats (Penn State: ntriples, UCSD: rdf/xml, need n3/ttl testing)

Actions

- Provide feedback on performance testing matrix, which are the most important to test before 4.0?
 - [Andrew Woods](#)
 - [Unknown User \(escowles@ucsd.edu\)](#): I picked four tests that seem like the highest priority to me: large number of files ingest, large file size ingest, mixed reads, mixed reads/writes
- [Andrew Woods](#): Prioritize postponed features for inclusion in 4.1