GSOC10 - Backport of DSpace 2 Storage Services API for DSpace 1.x

Backport of DSpace 2 Storage Services API for DSpace 1.x

Student: Andrius Blažinskas
Mentor: Mark Diggory

Abstract

DSpace 2.0 storage mechanism provides convenient way to store DSpace contents in various storage solutions. It is based on set of interfaces for which various implementations are possible and some beta releases already exist (Jackrabbit, Fedora, etc). DSpace 2.0 is in its early stages of development and DSpace 1.x releases yet can not take advantage of this new mechanism. To fix this, it is necessary to port DSpace 2.0 storage interfaces to 1.x. I propose implementing this backport. – Andrius Blažinskas

Relevant modules/classes

Module /class name	Description/Comments	Source code
dspace-api	DSpace API	http://scm.dspace.org/svn/repo/dspace/trunk/dspace-api
dspace-xmlui	XMLUI (Manakin)	http://scm.dspace.org/svn/repo/dspace/trunk/dspace-xmlui
storage-api	Constitute of DSpace 2 storage interfaces. Will be referenced from dspace-xmlui and other modules which will use new storage mechanism. Subject to change. (Update: heavily refactored - moved from mixin solution to services concept)	http://scm.dspace.org/svn/repo/modules/dspace-storage/trunk/api/
storage-legacy	Module will implement storage-api interfaces. Basically it will be the shim allowing modules to access DSpaceObjects (in dspace-api) using new storage-api.	http://scm.dspace.org/svn/repo/modules/storage-legacy/
dspace-services	DSpace services module. DSpace services framework will be used to manage and gain access to storage-api implementations.	http://scm.dspace.org/svn/repo/modules/dspace-services/
ProvidedStorageService	Class which acts as a mediator between caller and storage service implementations. However, its usage is questionable. (Update: since dspace-storage-api has been refactored and instead of mixin solution services way there chosen, this class or its modifications most likely will not be used.)	http://scm.dspace.org/svn/repo/modules/dspace-storage/trunk/impl/src/main/java/org/dspace/services/storage/ProvidedStorageService.java

Development plan

- Analysis part:
  - Analysis of dspace-api module
  - Analysis of dspace-services module
  - Deeper review of spring usage in DSpace
  - Analysis of dspace-database module
  - Analysis of dspace-storage-db-2.0.x module
  - Analysis of AIP prototype
- dspace-api adaptation to changing needs:
  - Evaluation and incorporation of changes described at https://wiki.duraspace.org/display/DSPACE/GSoC+Collaboration+Scratchpad
  - Implementation of changes decided during commiter/student meetings
- Implementation of storage-legacy module
- dspace-xmlui relation to storage-api
- Creation of java documentation
  ...

Evolution of storage-api

Recommended changes to "existing" DSpace 2 storage-api:

- "StorageProperty[] parameters should be dropped from the StorageEntity object all together." [DSPACE:2]
- "StorageProperty service methods for performing CRUD operations on Storage properties be maintained on a separate mixin interface." [DSPACE:2]
- "StorageRelation be removed from the object model and relations be captured only by attaching StorageEntities as "values" of StorageProperties." [DSPACE:2]
- "... remove methods like getEnititesAtLocation("/community/collection") and would recommend the use of the Search API instead for the retrieval..."

- "Mapping a prefix to the provider should warrant needing a separate interface to be implemented. That could just be part of assigning the StorageService to the map it is cached in the ProvidedStorageService."

Update: after long discussions on how dspace-storage-api should look like, it was chosen to refactor whole api and move from mixin solution to services concept, thus some of initial proposals on api changes does not reflect in current model implementation.

Proposed dspace-storage-api

Most current basic dspace-storage-api implementation class diagram provided below:

blocked URL

Short reference history how dspace-storage-api class diagram evolved during discussions can be found here: http://andriusb.labt.lt/gsoc/ (PNG files only).

Provided api will evolve further, but most likely that basic components provided in diagram won't change or only minor changes can be introduced. Where are plans on incorporating interfaces for indexing, search and ContentModel services.

Backporting strategies

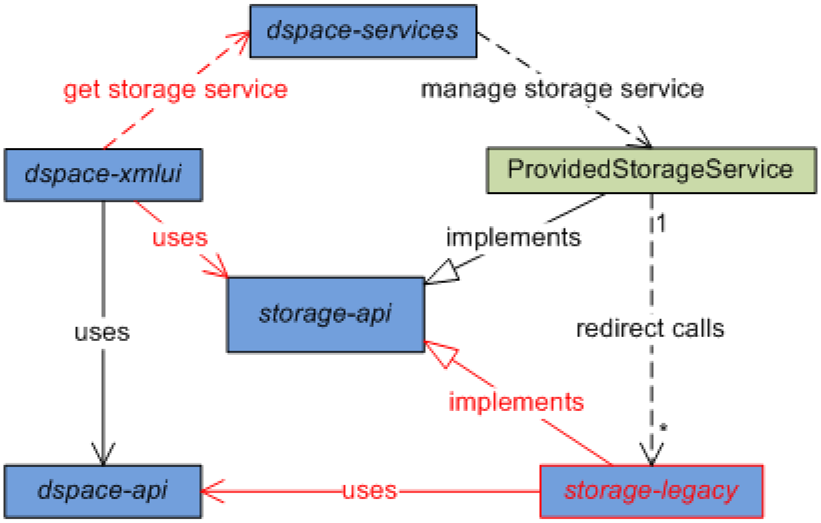
There are different ways to backport dspace-storage into DSpace 1.x, some of these are described here.

Since DSpace 1.x model data is mainly accessed through particular DSpace 1.x entities (Community, Collection, Item, Bundle, Bitstream, BitstreamFormat), new storage mechanism somehow will interact with them. There was discussions (during IRC meetings) on whether DSpaceObjects should be backed by dspace-storage or is it something what should be "covered over" by dspace-storage.

- Backing DSpaceObjects by dspace-storage allows immediate effect since all current modules uses these entities. However, this approach also involves changing internals of these entities, which opens possibility to introduce bugs affecting everything. This way created storage-legacy module would probably have to overtake the most DSpaceObjects internals which also are coupled back with dspace-api (authorization etc.).
- DSpaceObjects "cover over" by dspace-storage, if correctly implemented, is a cleaner choice, since changes in dspace-api can be avoided. storage-legacy module in this case would act only as a shim, providing access to dspace-api through storage-api. Conceptually, such solution probably is bad (storage logics should reside in storage-legacy), however it is a good "temporary" measure helping in moving DSpace 1.x to using new storage api.

Proposed backport strategy

Shim or "cover over" solution is chosen as backporting strategy. Diagram below describes it in more detail.



Elements in red are being implemented.

Update: since dspace-storage-api was moved from mixin solution to services, class ProvidedStorageService is replaced with EntityStorageService, PropertyStorageService and BinaryStorageService.

References

1. GSOC 2010 proposal: Backport of DSpace 2 Storage Services API for DSpace 1.x, http://andriusb.labt.lt/gsoc/2010/dspace/proposal1.html
2. GSoC Collaboration Scratchpad, https://wiki.duraspace.org/display/DSPACE/GSoC+Collaboration+Scratchpad