JHU API-X Use Case: Package Deposit

Title (Goal)	Package deposit with validation
Primary Actor	Data management specialist
Scope	Component
Level	
Author	Elliot Metsger, rev. by Joshua Westgard
Story	As a data management specialist, I want to deposit, validate, and ingest a package that I have created into the repository, with the option of placing package contents into a new collection or adding them to an existing collection. I have prepared a package of content for deposit into the repository. The package conforms to local standards/requirements, and may conform to a recognized standard like BagIt. The package contains a payload (the domain content to be ingested) and a description of the payload. The payload description includes a manifest of the content in the package, along with fixity for the package contents. I want to deposit my package into the repository, and have the option of creating a new collection populated with the package contents, or deposit the package to an existing collection. Despite being a trained data management specialist, I may have made mistakes when composing the package. Therefore the repository should verify that the package meets local standards/requirements. If the package conforms to a recognized standard, it should be checked for conformity to the recognized standards as well. The repository should utilize the fixity information supplied in the package to verify the integrity of the package payload. Upon accepting the package, the repository should provide some mechanism for me to understand where in the deposit workflow my package is.

Web Resources and Interactions

This extension would create new resources by POSTing those resources to some location in the repository, either an existing collection or the the /rest URI for new collections. The extension would be mediating the interaction of the user with the resources because it would need to interpret the package and make the necessary series of requests to create the new resources, or to PATCH existing resources in the case where items are being added to an existing collection. In addition to the payload, manifest, and fixity information, presumably some RDF graph would need to be submitted with the package in order to guide API-X on structuring the deposit.

Preconditions

The API extension architecture should expose this feature via a general URI such as 'fcrepo/deposit' and should manage the interaction with individual existing resources for the user, based on the provided package.

A possible alternative trigger would be when a particular type of package is POSTed to any container, but it is unclear how such an interaction would be distinguished from interactions made possible by the existing Fedora API.

Deployment or Implementation notes

I would anticipate the need for a Java-, Python-, or Ruby-based tool to manage the validation and deposit.

Proposed Requirements

This extension will need to leverage Fedora's existing transaction feature.

API-X Value Proposition

It seems likely that package deposit will be a widely desired and adopted feature, and to the extent that API-X would provide a common model for implementing this feature that would allow for easier deployment and sharing of such deposit tools, that would be a benefit for the larger community.

In addition, it seems that a single public URI for this service makes the most sense, and therefore having API-X obscure some of the details of the backend infrastructure would increase the "pluggability" of this sort of webapp.