RESTful Relationship API

Problem

In order to consider Fedora's REST API complete (FCREPO-543), we need to have functional equivalents to API-M's relationship management methods.

This includes the ability to:

- add and remove relationships incrementally
- get a subset of relationships

Approaches

We started by talking about a couple general approaches during the November 3rd Committer Meeting (notes and audio here).

1. Generic endpoint for updating / querying all relationships in the repository. This would allow one to update relationships without specifying which object or datastream they resided in explicitly. There was some concern that this might result in confusion because it sets up the expectation that arbitrary RDF is accepted, but in reality, it would only accept changes that were valid in RELS-EXT and RELS-INT datastreams (e.g., those whose subject is a Fedora object and whose predicate is not a reserved predicate).
2. Object-specific endpoints for updating relationships on a per-datastream basis. This approach would require that applications using the API are aware of which objects and datastreams the relationships are asserted in. While discussing this approach, we noted that this pattern could be followed in a general way to make partial datastream updates possible for other types of datastreams as well.

Strawman Proposal

During the call, we talked most about approach #2, and started to define what it would entail:

1. Define a generic method (or set of methods) for applying partial changes to datastreams.
2. Implement it for RELS-EXT and RELS-INT (RDF-based) so that additions and deletions to those datastreams can be made via SPARQL update.

Example:

- **HTTP Verb:** POST
- **URL:** /objects/$pid/fedora-system:SomeBuiltInSDef/updateDatastream?dsID=$dsID&type=sparql-update
- **Body:** A SPARQL/Update document (UTF-8?) to apply to the datastream
- **Successful response:**
  - **Code:** 200
  - **Location header:** URL to datastream version
  - **Content-type header:** text/plain; charset=utf-8
  - **Body:** URL to datastream version
- **Failed response (bad request):**
  - **Code:** 400
  - **Content-type header:** text/plain; charset=utf-8
  - **Body:** (one-line message)
    - Unrecognized update type: $type
    - No such object: $pid
    - No such datastream in $pid: $dsID

Notes:

- Adding or modifying the entire content of a datastream would still be possible via a POST or PUT, respectively, to the datastream URL as it is today.
- The syntax for invoking disseminations has not yet been decided for the REST API; the URL above with fedora-system:SomeBuiltInSDef assumes an obvious possible syntax.

Outstanding Issues:

1. Does SPARQL Update have a mime type? If so that might make sense to use as the value of the type parameter
2. The strawman does not yet include a way to get relationships. Perhaps this could also be a built-in function of SomeBuiltInSDef.
    1. Q: Would it require one to specify which datastream (e.g. RELS-EXT) the relationships reside in?
    2. Q: What form of query would it accept? A simple S,P,O style query, or SPARQL?

Thoughts?

Please feel free to comment on this proposal here or on the dev list.