

# Fedora4: The Open Linked Data Platform

Andrew Woods, Duraspace, [awoods@duraspace.org](mailto:awoods@duraspace.org)

Robert Sanderson, Stanford University, [azaroth@stanford.edu](mailto:azaroth@stanford.edu)

*Keywords:* Web Standards, Linked Data, Fedora4

## Abstract

Linked Open Data has moved from being a buzzword to a fundamental building block of modern repositories and information systems. Its explosion, taking in this domain the form of scholarly and scientific datasets, publications, annotations, cultural heritage descriptions and other repository-based content, offers unprecedented opportunity for scientific and societal advancement. It is the interconnections that integrate systems and resources, however, that turn disparate ideas into unanticipated solutions.

It has been almost nine years since Tim Berners-Lee laid out the guiding principles[1] of the Semantic Web:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using the standards (RDF\*, SPARQL)
4. Include links to other URIs so that they can discover more things

Fortunately, we now have the tools at our disposal to realize the potential of a semantically meaningful, interconnected web of data *and* documents.

The Open Repositories community largely understands the value of linked data. The trouble has been in answering the question of “how?” to do this together, rather than “why?” do it at all. In order to be effective, we need to have solid guidelines, common practices, and well specified toolsets and products. A confluence of developments has opened the door to exactly this.

In October of 2012, the initial W3C working draft of the Linked Data Platform 1.0 (LDP) document was published[2]. In July of 2012, the demand for a next-generation Fedora platform was channeled into the Fedora 4 (F4) project, which at the time was termed Fedora Futures. The alignment of these two stars set in motion events that provided both requirements and solutions for the community.

Today, the LDP specification has graduated to the W3C status of “proposed recommendation”[3], with a likely transition to the status of full “technical recommendation” before the conference takes place. The specification defines the HTTP interaction model expected of a linked data server. In other words, if an application is storing and serving Linked

Data on the web, LDP can be thought of as the guidelines by which users or other applications should conceptualize those resources and expect to interact with that server. LDP standardizes first-class participation in the read/write web.

In parallel to the development of LDP, F4 was also taking form. With a re-conceived Fedora came the opportunity to rethink the application's RESTful API. A design decision was made to have F4 speak RDF natively, which meant consuming and serving RDF (the fundamental element of linked data) in an intuitive fashion by default. Recognizing the long term benefits of designing to standards, where they exist, it became apparent that LDP represented a constructive direction. As a result, F4 adopted the LDP model in implementing its RESTful API.

This presentation is intended to unveil some of the mysteries around practical usage of linked data by discussing the Linked Data Platform specification on the one hand, and its implementation in Fedora 4 as a fully featured repository product on the other.

## **Linked Data Platform**

The Linked Data Platform is a rationalization of best practices in the realm of REST and Linked Data. The purpose of the specification is to try and ensure that the many read/write web, or "Web 2.0", platforms can interoperate. As most already follow the same basic patterns, LDP simply tidies up in the corners where different decisions are made in the absence of best practice guidelines. The recommendations and specification apply equally to linked data resources in RDF (RDF Sources, in LDP terminology), and to any other content that should be sent between systems -- images, video, audio, html, or arbitrary binary data (conversely, non-RDF Sources).

The majority of the specification merely states the appropriate patterns to use to communicate via HTTP when creating, retrieving, updating and deleting resources. These operations use the HTTP methods PUT, GET, POST/PATCH and DELETE respectively. Where it extends beyond simple CRUD operations is in the standardization of containers and the interaction patterns with multiple resources, rather than individuals.

LDP has the notion of 3 types of container, or folder, in which RDF and non-RDF resources can be managed. Basic Containers are just that, a folder that knows of its contained resources and can list them, as well as providing a URL to add new resources. Direct Containers add the ability to automatically link resources in the container to other resources, for example uploading an image of the person to a container might link that image's newly minted URI with the URI that identifies the person in the system. Indirect Containers add further, similar, functionality with an additional level of indirection.

Additional features that are coming in related specifications include paging through the resources in a collection, a format for HTTP PATCH requests explicitly designed for linked

data, and later will expand to authentication based on the Web ACL structure, and resource oriented query patterns.

## Fedora4

The first section of the presentation will discuss LDP with worked examples of interaction with both individual resources and containers. With this foundation laid, the second section will discuss the role LDP plays in Fedora 4. The Fedora and general repository communities have grown very accustomed to long-standing concepts such as “objects” and “datastreams”. As F4 aligns with the broader expectations of a standardized linked data server, so must the terminology and modeling of the assets and resources therein, including resources and containers. Additional interaction patterns are used in order to take full advantage of the dynamic aspects of a RESTful LDP interaction. The degree of this adoption can be seen in the results of Fedora’s compliance with LDP’s test compatibility kit[4].

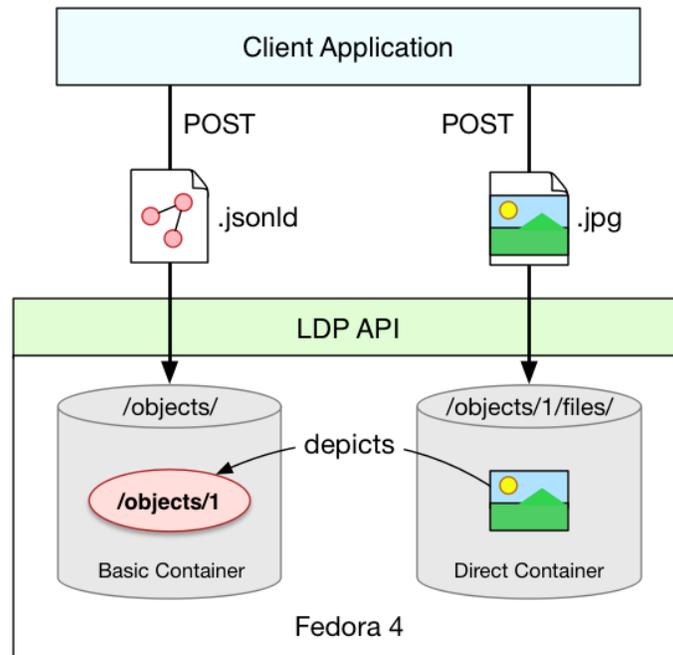


Figure 1. LDP Containers in Fedora4

Conceptually, Linked Data represents a community pathway for addressing issues of integrating repositories and surrounding services with the broader web of data and things. LDP defines a common blueprint for designing our application interactions. Fedora 4 provides a concrete implementation of this blueprint simplifying our collective next step into a searchable, navigable web of scholarly output, researchers, and curated representations.

## References:

[1] <http://www.w3.org/DesignIssues/LinkedData.html>

[2] <http://www.w3.org/TR/2012/WD-ldp-20121025/>

[3] <http://www.w3.org/TR/LDP/>

[4] [https://dvcs.w3.org/hg/ldpwg/raw-file/default/tests/reports/ldp.html#subj\\_7](https://dvcs.w3.org/hg/ldpwg/raw-file/default/tests/reports/ldp.html#subj_7)