Outcomes - REST API

Why Maintain the API as an Independent Product?

A retrospective consideration of over six years of development in the Fedora Commons 3 ecosystem (repository and related application platforms) suggests two broad cases for managing a common repository REST API as a separate product: One emerging from concerns of the Fedora Commons 3 and 4 developers, and one from the broader community of projects and institutions building projects around Fedora Commons and related products.

The Community Case

The institutions and communities served by the Fedora Commons software products are diverse, but they have in common investments in transparency, data mobility, and durability. These common investments often appear in the form of common integration patterns involving Fedora Commons and other software. Rather than elaborate a boutique API bound to a particular implementation of Fedora, we feel these community interests would be better served by creating independent, conformance-testable API modules offered as extensions to an API supported more widely than the Fedora community.

In the 10 years since the 2.0 design work (and arguably earlier), the Fedora Commons project has attempted to overlay linked data descriptions on stored or linked binary assets. Fedora Commons has never been the sole product in this space, and this more general case has evolved into the Linked Data Platform Specification, a W3C standard that describes much of what Fedora Commons provides at the API level. Given the broad audience and support for this specification, the Fedora developers have chosen to make it the basis for a Fedora Repository REST API, of which the Fedora Commons software will be a reference implementation. Building around an independent API based in broader standards will allow the community more security and flexibility: The ecosystem of software around the repository will be able to use more sources of data, back-end migration may be possible with less disruption to running applications, and the chance of third-party tools being available for an institution's data will be greater.

The Developer Case

Over the course of Fedora Commons 3.x, there were four related concerns tracked under a single version number:

1. The HTTP APIs, both REST and SOAP
2. The FOXML serialization format for stored objects (and interpretive software therefor)
3. The Java interfaces for the Fedora Commons 3 modules
4. The repository configuration (fcfg markup and Fedora "home" directory)

The collection of all these concerns under a single compatibility promise was an impediment to development and contributed to a slow release cycle both for bug fixes and for adaptation to a rapidly changing technology landscape. Going forward, the developers believe that separating these concerns will facilitate a more nimble and manageable development lifecycle for the current iteration of Fedora Commons.

Groups of Work

There are several major groups of work created as a result of review and discussion around Fedora's HTTP API. It is important to understand that all of this work assumes the plan of API partitioning as developed by the TWG.

1. Partition the APIs
    a. This piece of work comprises the construction of fully-specified API modules as outlined in the partition plan. These specifications must be of such detail and precision as to enable a complete reimplementation of the APIs they describe without reference to the Fedora Commons implementation code. This work includes the definition of optional API modules and their advertisement, conformance levels if necessary and as appropriate for each, and accompanying conformance test suites.
2. Partition the ontologies
    a. This piece of work parallels the previous item. It entails the processing of the current Fedora ontologies to partition them in line with the partition of the API itself. The ontology modules will, as appropriate, be made part of the API module specifications that they support.
3. Partition the code
    a. This piece of work requires massive changes to the current implementation code. It will require breaking apart the current HTTP API module into multiple modules that each support a separate Fedora API module, and developing useful deployment, configuration and wiring patterns and tools to enable integrators of Fedora Commons repositories to choose for just those API modules desired in a given repository instance.