

Use Case Evaluation

This page describes a potential set of questions to help evaluate use cases, present them in a clear and uniform way, and distill into requirements. The goal is to explore possible ways in which the API-X can be used to address each use case as concretely as possible, but falling short of providing a true specification or extension design.

Web Resources and Interactions

What web resources (URIs, patterns) are relevant to this particular API Extension? Would this extension expose any new resources, or involve existing ones (e.g. URI of a fedora resource, as defined via the Fedora HTTP API)? How does the user interact with the resource, and how is the extension involved in that interaction?

Preconditions

Under what conditions should the API extension architecture expose a URI, or invoke this extension? The current understanding of 'invoke' means "direct an HTTP request or response to an extension for processing." If this particular use case uses 'invoke' in a different way, please define.

Examples of preconditions include:

- Extension is invoked when a request is made to any repository resource.
- Extension is invoked when a request to any resource contains certain content (e.g. is a POST, or contains a particular HTTP header)
- URIs with pattern `/path/to/object/ext:myExtension` are exposed whenever a given Fedora object has an `rdf:type` of `myns:myType`
- Extension is invoked when a request is made to a web resource (URI) exposed by the API-X.

Deployment or Implementation notes

Are there any deployment or implementation-related details that may be relevant to the API extension architecture?

Examples include:

- We anticipate implementing logic for this extension as a web service written in Ruby
- This extension uses features of modeshape, and inherently needs to be installed with Fedora
- This extension will use fcrepo-kernel-api, and needs to be co-located with an implementation of fcrepo-kernel.
- Implemented as camel routes that can be deployed into any osgi container

Proposed Requirements

What requirements may this use case place on the API extension architecture?

API-X Value Proposition

What do you see as a potential value proposition of the API extension architecture for this use case above and beyond, say, deploying web service somewhere and having clients point to its URI, or compiling an extension into Fedora using webapp+, or deploying into an OSGi container beside Fedora?

Examples include:

- The API-X defines a single public URI for this service, which abstracts away the details of its real location on my backend infrastructure, allowing the implementing web service to be easily re-located.
- The API-X allows the presence of this service to be discovered by clients
- The API-X provides a convenient way to integrate this service as a filter for all requests.
- The API-X provides a convenient way to distribute and deploy the extension so that others can easily use or evaluate it.