Introduction to Fedora
Overview, examples, and core features

David Wilcox, DuraSpace
@d_wilcox
Learning Outcomes

Understand the purpose of a Fedora repository

Learn what Fedora can do for you

Understand the key capabilities of the software
Our community is part of an interconnected, worldwide, scholarly ecosystem.
DuraSpace open source projects
DuraSpace services

DURACLOUD™

DSpace DIRECT

@archivesDIRECT
Fedora™

Flexible Extensible Durable Object Repository Architecture

Concept
Implementation
Community
Fedora...

Stores, preserves, and provides access to digital objects

Supports flexible and complex content models for objects

Supports complex semantic relationships between objects inside and outside the repository using RDF

Supports millions of objects, both large and small

Interoperates with other applications and services
Why use Fedora?

Fedora is **flexible**: it can handle both simple and complex use cases

Content in Fedora is **durable**: Fedora supports long-term preservation

Fedora powers successful digital repository and DAM applications

Fedora is **standards-based**

Fedora is backed by a **thriving community**
Fedora Front-Ends

Fedora is *middleware*

You can build a custom framework, or join a broader community:
Fedora in Production
Institutional Repository

https://scholarspace.library.gwu.edu/
Research Data

https://era.library.ualberta.ca/
Manuscripts

York's Archbishops Registers Revealed provides free access to over 20,000 images of Registers produced by the Archbishops of York, 1225-1650, in addition to a growing searchable index of names, subjects, places and organisations. The registers are a valuable, and in many cases, unexploited source for ecclesiastical, political, social, local and family history - covering periods of war, famine, political strife and religious reformation in the Archdiocese of York and the wider Northern Province.

You can browse images using the options below, or search the 2688 indexed entries via the search interface. Further information, guidance and supporting material relating to the registers will be added to the site as time goes on.

Pilot imaging and development, as well as the indexing of the register of Archbishop Neville

https://archbishopsregisters.york.ac.uk
Basic Concepts
Web Resources

Everything is a web resource with a URI

Resources have properties expressed as RDF triples

Resources can contain other resources (containers) or files (binaries)
Book Example

Book Collection

- Book 1
  - Page 1
    - Page1.jpg
    - Page1.tiff
  - Page 2
    - Page2.jpg
    - Page2.tiff

- Book 2
  - Page 1
    - Page1.jpg
    - Page1.tiff
  - Page 2
    - Page2.jpg
    - Page2.tiff

Container

Binary
RDF Properties


Created at
2016-05-16T19:06:02.475Z by fedoraAdmin
Last Modified at
2016-05-16T19:06:02.475Z by fedoraAdmin
Children 0

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>created</td>
<td>2016-05-16T19:06:02.475Z</td>
</tr>
<tr>
<td>createdBy</td>
<td>fedoraAdmin</td>
</tr>
<tr>
<td>lastModified</td>
<td></td>
</tr>
</tbody>
</table>
Core Features
Fedora system architecture

- Fedora HTTP API
- Fedora Services
- ModeShape
- Storage (Containers and Binaries)

Services:
- Access & Preservation
- Repository
- Storage Services
Standards

Focus on existing standards

Fewer customizations to maintain

Opportunities to participate in related communities
Core Services and Standards

1. Create/Read/Update/Delete - Linked Data Platform ✔

2. Versioning - Memento

3. Authorization - Web Access Control ✔


5. Messaging - Activity Streams 2.0 ✔
Hands-on: CRUD

http://localhost:8080/fcrepo/rest/
user/pass: fedoraAdmin/secret3
Available Operations via HTML UI

- **GET/HEAD/OPTIONS** (Retrieval)
- **POST/PUT** (Creation)
- **PATCH** (Update)
- **DELETE** (Removal)
Step 1a: RDF Resource Creation (POST)

1. Go to http://localhost:8080/fcrepo/rest (root node)
2. In “Type” select field choose “container” (default)
3. In “Identifier” text field enter “basic”
4. Press “add” button

This will create a new RDF Resource (LDP Basic Container) and redirect us to our next slide!

Username: fedoraAdmin
Password: secret3
Step 1b: RDF Resource Creation (POST)

1. You will be redirected to [http://localhost:8080/fcrepo/rest/basic](http://localhost:8080/fcrepo/rest/basic)
2. In “Type” select field choose “container” (default)
3. In “Identifier” text field enter “collection”
4. Press “add” button

This will create a new RDF Resource (LDP Basic Container) and redirect us to our next slide. This way we matched what we had in our cheat sheet!
Step 1c: RDF Resource Creation (POST)

1. You will be redirected to [http://localhost:8080/fcrepo/rest/basic/collection](http://localhost:8080/fcrepo/rest/basic/collection)
2. Use “breadcrumb” to go back to [http://localhost:8080/fcrepo/rest/basic](http://localhost:8080/fcrepo/rest/basic)
3. In “Type” select field choose “container” (default)
4. In “Identifier” text field enter “images”
5. Press “add” button

This will create a new RDF Resource (LDP Basic Container) and redirect us to our next slide!
Step 2: Resource Retrieval (GET)

1. Every time you got redirected after creating a Container you were using GET.

2. Retrieval is accessed directly via the LDP Path that defines a resource and contains user and some server managed RDF triples.
Step 3: Binary Resource Creation (POST)

1. Go to http://localhost:8080/fcrepo/rest/basic/images
2. In “Type” select field choose “binary” In “Identifier” text field enter “hotdog”
3. In “File” choose any small image
4. Press “add” button

This will create a new Binary Resource (LDP Non RDF Source) and redirect us to our next slide!
Step 4: Binary Resource Retrieval (GET)

1. You will be redirected to
   http://localhost:8080/fcrepo/rest/basic/images/hotdog/fcr:metadata

2. Notice the fcr:metadata part!
   a. Image is LDP contained in “/hotdog”
   b. Its metadata (rdf properties you can manipulate) in a virtual subpath named /fcr:metadata

Why? That way you can keep operations separated and you can also directly describe via RDF properties binary content.

Access
http://localhost:8080/fcrepo/rest/images/hotdog/
directly to download binary!
Step 5: Update RDF Properties (PATCH)

1. Navigate to http://localhost:8080/fcrepo/rest/basic/collection

2. We will add an “pcdm:Object” property using “Update Properties”
   a. Make sure “PREFIX pcdm” is there
   b. At the end rewrite “DELETE…” to

   ```
   DELETE {}
   INSERT { <> ebucore:width "100"}
   WHERE {}
   ```

   c. Press “Update”
Last step: Delete a resource (DELETE)

1. Stay at
   http://localhost:8080/fcrepo/rest/basic/images/hotdog/fcr:metadata

2. Press “DELETE” (the red one)

3. You will be redirected to the parent resource after deletion.

4. Go again to
   http://localhost:8080/fcrepo/rest/basic/images/hotdog

What do you see?
Fedora creates a tombstone resource at “original/path/fcr:tombstone” URL, in this case “basic/images/hotdog/fcr:tombstone” (try that last path in your Browser)

So, to recreate a resource at that same PATH you need to delete the tombstone placeholder first and that can not be done via HTML UI
Authorization:
Web Access Control
Authorization - Web Access Control

Authorization is optional and pluggable

WebAC is a W3C approach for managing authorization using linked data

Interoperable with other applications that implement the same approach

Implemented in Fedora 4 by community stakeholders
Versioning

Versions can be created on demand via the REST-API

A previous version can be restored via the REST-API
Hands-on: Versioning
Create a container named “Book”
Create version “v0” of “Book”
Add “dc:publisher” to “Book”

```sparql
INSERT {
  <> dc:publisher "University Press"
}
WHERE {}
```
Create version “v1” of “Book”

Inspect and revert to v0
Fixity

Over time, digital objects can become corrupt

Fixity checks help preserve digital objects by verifying their integrity

On ingest, Fedora can verify a user-provided checksum against the calculated value

A checksum can be recalculated and compared at any time via a REST-API request
External Services
External Component Integrations

Leverages the well-supported Apache Camel project

Camel is middleware for integration with external systems

Can handle any asynchronous, event-driven workflow
External - Indexing

Index repository content for search

Indexing is configurable - could be based on any property

Solr and Elasticsearch have been tested
External - Triplestore

An external triplestore can be used to index the RDF triples of Fedora resources

Any triplestore that supports SPARQL-update can be plugged in

Fuseki, RDF4J, and BlazeGraph have been tested
Audit Service

Maintains a history of events for each repository resource

Both internal repository events and events from external sources can be recorded

Uses the existing event system and an external triplestore
Performance and Scalability
Test Plans

Testing large files, many files, and many containers

Tests are performed by community members

Have concerns about performance and scale? Join the group!
Metrics

A number of scalability tests have been run:

- Uploaded a 1 TB file via REST API
- 17 million objects via REST API
- 3.5 million files via REST API
Supporting and Sustaining Fedora
Fedora facts

Managed by DuraSpace (not-for-profit)

Funded by the community

Collaboratively developed by the community

Supported by 2 full-time staff members (not developers)
Useful Resources

Fedora 4 documentation
https://wiki.duraspace.org/display/FEDORA4x/Fedora+4.x+Documentation

Fedora 4 wiki
https://wiki.duraspace.org/display/FF

Fedora 4 mailing lists
https://wiki.duraspace.org/display/FF/Mailing+Lists+etc