

# Introduction to Fedora 5.0 and Alternate Implementations

Overview, core features, extensions

David Wilcox, DuraSpace



# Workshop Preparations

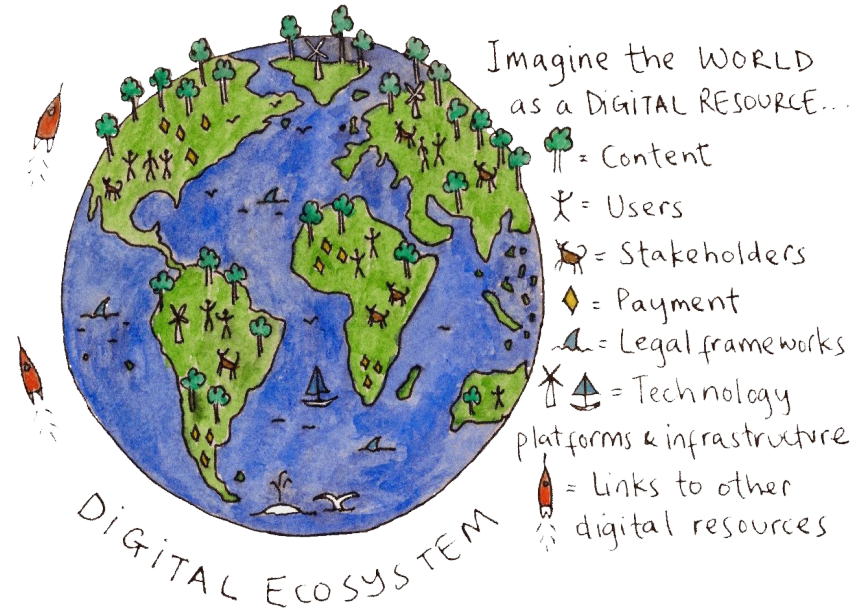
1. Download [one-click Fedora 5.0.0 application](#)
2. Make sure you have [Java 8](#) installed
3. If you are using Windows, ensure you have a fully-featured Command Line Interface such as the Powershell included with [GitHub Desktop](#)

# About DuraSpace

DuraSpace is not for profit organization that provides leadership and innovation for **open technologies**

We work to ensure that **current and future generations** have **durable** and **persistent access** to our **collective digital heritage**

Our **community** is part of an **interconnected, worldwide, scholarly ecosystem**



**DURASPACE**

# DuraSpace open source projects



**Fedora**<sup>TM</sup>



# Fedora API Specification



## Motivation

“The Fedora specification will detail the exact services and interactions required for a server implementation to be verified as "doing Fedora".”

<https://fedora.info/spec>

# Specification Goals

Stability for clients

Support for alternative use cases

Standardization

# Specification Components

**Resource Management** (Linked Data Platform)

**Resource Versioning** (Memento)

**Resource Authorization** (Web Access Control)

**Notifications** (Activity Streams)

**Binary Resource Fixity** (HTTP Headers)



# Resource Management



# Resource Management

Extends the Linked Data Platform

Some optional LDP features are mandatory

Expanded support for external binary content

# Run the One-Click Application

1. Double click the fcrepo-webapp-5.0.0-jetty-console.jar file
2. Click the Start button
3. Navigate to <http://localhost:8080>

## Step 1a: RDF Resource Creation (POST)

1. Go to <http://localhost:8080/rest> (root node)
2. In “**Type**” select field choose “**basic 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!

## Step 1b: RDF Resource Creation (POST)

1. You will be redirected to <http://localhost:8080/rest/basic>
2. In “**Type**” select field choose “**basic 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.

## Step 1c: RDF Resource Creation (POST)

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

## 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/rest/basic/images>
2. In “**Type**” select field choose “**binary**” In “**Identifier**” text field enter “**snow**”
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/rest/basic/images/snow/fcr:metadata>
2. Notice the fcr:metadata part!
  - a. Image is LDP contained in “/snow”
  - 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.

## Step 5: Update RDF Properties (PATCH)

1. Navigate to <http://localhost:8080/rest/basic/images/snow>
2. We will add an “ebucore:width” property using “Update Properties”
  - a. Make sure “PREFIX ebucore” 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/rest/basic/images/snow/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/rest/basic/images/snow>

What do you see?

# Departed

Fedora creates a tombstone resource at  
“original/path/fcr:tombstone” URL, in this case  
“basic/images/snow/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

# External Binary Content



# External Binary Content

rel="<http://fedora.info/definitions/fcrepo#ExternalContent>" link header

Target of the link is a URI from which content may be retrieved

Handling attribute for external content: either copy, redirect, or proxy

# Copy, Redirect, Proxy

**Copy:** Copy the binary from the provided URI and then treat it as a normal internal LDP-NR (binary).

**Redirect:** On requests for the LDP-NR provide a redirect (302 Found or 307 Temporary Redirect) to the external URI.

**Proxy:** On requests for the LDP-NR proxy the request through the Fedora server with the same interaction as an internally stored LDP-NR.

# External Binary Content Configuration

1. Create a file called allow.txt in your application directory
2. Add the full path to a local file directory and a website, e.g.

```
file:///Users/dwilcox/Documents/Workshops/2019-01_SCSFUG/  
https://duraspace.org/
```

3. Start the application with the following arguments:

```
java -Dfcrepo.external.content.allowed=/full/path/to/allow.txt -jar  
fcrepo-webapp-5.0.0-jetty-console.jar
```

[See online documentation](#)



# External Binary Content Usage - Proxy

```
curl -i -H"Link:  
<file:///Users/dwilcox/Documents/Workshops/2019-01_SCSFUG/test.png>;  
rel=\"http://fedora.info/definitions/fcrepo#ExternalContent\";  
handling=\"proxy\"; type=\"image/png\"" -XPUT  
http://localhost:8080/rest/test_proxy_local_file
```

# External Binary Content Usage - Redirect

```
curl -i -H"Link:  
<https://duraspace.org/wp-content/uploads/2018/03/fedora-log  
o.jpg>;  
rel=\"http://fedora.info/definitions/fcrepo#ExternalContent\  
\"; handling=\"redirect\"; type=\"image/jpeg\" -XPUT  
http://localhost:8080/rest/test_redirect_remote_file
```

# External Binary Content Usage - Copy

```
curl -i -H"Link:  
<https://duraspace.org/wp-content/uploads/2018/03/fedora-log  
o.jpg>;  
rel=\"http://fedora.info/definitions/fcrepo#ExternalContent\  
\"; handling=\"copy\"; type=\"image/jpeg\" -XPUT  
http://localhost:8080/rest/test_copy_remote_file
```

# Resource Versioning



# Resource Versioning

Extends the Memento specification

Versioned resources provide a TimeGate interaction model

# Hands-on Versioning: UI

1. Create a new container called “version-test”
2. Click “Create Version”
3. Modify the resource, e.g.

DELETE {}

INSERT { <> dc:title "My resource" }

WHERE {}

# Hands-on Versioning: UI

1. Click “View Versions”
2. Click the date stamp

# Hands-on Versioning: cURL

Get a list of all available versions:

```
curl -H "Accept: text/turtle"  
http://localhost:8080/rest/version-test/fcr:versions
```

Create a new version with default Memento-DateTime:

```
curl -X POST  
http://localhost:8080/rest/version-test/fcr:versions
```



# Hands-on Versioning: cURL

Create a new version with supplied body and Memento-Datetime:

```
curl -X GET http://localhost:8080/rest/version-test >  
resource.ttl
```

```
curl -X POST -H "Memento-Datetime: Wed, 30 May 2018 23:02:44  
GMT" -H "Content-Type: text/turtle" --data-binary  
"@resource.ttl"  
http://localhost:8080/rest/version-test/fcr:versions
```

# Hands-on Versioning: cURL

Get a specific version:

```
curl -H "Accept: text/turtle"  
http://localhost:8080/rest/version-test/fcr:versions/20180530230244
```

Delete a specific version:

```
curl -X DELETE  
http://localhost:8080/rest/version-test/fcr:versions/20180530230244
```

# Hands-on Versioning: cURL

Get a specific version:

```
curl -H "Accept: text/turtle" -H"Prefer: return=representation;  
omit=\"http://fedora.info/definitions/v4/repository#ServerManaged\""  
http://localhost:8080/rest/version-test/fcr:versions/20190117195545 >  
version.ttl
```

Restore a specific version:

```
curl -X PUT -H "Content-Type: text/turtle" -H "Prefer: handling=lenient;  
received=\"minimal\"" --data-binary @version.ttl  
http://localhost:8080/rest/version-test -i
```

# Hands-on Versioning: cURL

1. Make a container and create a new version
2. Delete the versions container:

```
curl -X DELETE
```

```
http://localhost:8080/rest/path/to/resource/fcr:versions
```

# Resource Authorization

Based on Web Access Control recommendations from Solid

Access Control Lists are LDP RDF Sources

A default policy can be set and overridden

ACL inheritance follows LDP containment

# Notifications

Based on Activity Streams and Linked Data Notifications

Notifications are created whenever a resource is changed via HTTP

Serializations must conform to Activity Streams and should use the AS vocabulary

# Binary Resource Fixity



# Binary Resource Fixity

Two fixity verification mechanisms:

**Transmission:** A digest header can be included to guard against transmission faults

**Persistence:** A checksum can be retrieved to compare against a known value



# Transmission Fixity

1. Download [poster.jpg](#)
2. Upload to Fedora including a SHA-256 checksum:

```
curl -i -X PUT --data-binary "@poster.jpg" -H"digest:  
sha-256=06d788c46e45bb1c1b1ac75493f4636b880bcd7074df4f838470ca61749e8  
89c" "http://localhost:8080/rest/basic/images/poster"
```

# Persistence Fixity

Request the SHA-256 checksum:

```
curl -I -H "Want-Digest: sha-256"  
"http://localhost:8080/rest/basic/images/poster"
```

# Persistence Fixity

Make SHA-256 the default algorithm:

```
curl -X PATCH -H "Content-Type: application/sparql-update"  
--data-binary "@body.rdf"
```

<http://localhost:8080/rest/basic/images/poster/fcr:metadata>

Calculate and compare the checksum:

```
curl -H "Accept: text/turtle"
```

<http://localhost:8080/rest/basic/images/poster/fcr:fixity>

# External Services



# Two Service Types

## 1) External components

Consume and act off repository messages

## 2) API-X Integrations

Web services wired in at the HTTP layer

# 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



elastic

# 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





# Alternate Implementations



# Implementations Under Development

Cavendish

Derby

DRAS-TIC

Lambdora

Trellis

Trilpy

# Cavendish

<https://github.com/cavendish-ldp/cavendish/>

Fedora on a triplestore

Backed by Blazegraph

Content stored externally

Current Status: on hiatus due to Amazon acquisition of Blazegraph

# Derby

<https://github.com/fcrepo4-labs/derby/>

Fedora on Ruby

Used for Samvera testing

Current Status: Experimental

# DRAS-TIC

<https://github.com/UMD-DRASTIC>

Digital Repository At Scale That Invites Computation To Improve Collections

Data is stored in a distributed Apache Cassandra database

Current Status: Active

# Lambdora

<https://github.com/duraspace/lambdora/>

Serverless Fedora deployed in AWS

Uses AWS Lambda, API Gateway and DynamoDB

Current Status: on hiatus

# Trellis

<https://www.trellisldp.org>

A scalable platform for building linked data applications

Trellis underlies DRAS-TIC

Current Status: Active

# Trilpy

<https://github.com/zimeon/trilpy/>

Fedora on Python

Currently deployable as a test server for the Fedora API

Current Status: Active



# 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)



# ...its members

The image displays a diverse collection of logos for various academic institutions and libraries. The logos are arranged in a grid-like pattern across the page. Notable members include Carnegie Mellon University, Duke University, Cornell University, Weill Cornell Medical College, University of Washington, The National Library of Finland, University of Notre Dame, UNIL | Université de Lausanne, Columbia University Libraries, Texas A&M University, University of Texas Libraries, Emory Libraries & Information Technology, Wisconsin, University of Michigan MLibrary, Johns Hopkins University, Lafayette College, University of Oxford, University of Wisconsin, The University of New Mexico, Northwestern University, University of California, Santa Barbara, Stanford University, U.S. National Library of Medicine, EDINA, Imperial College London, UPEI University, University of California, Berkeley, its@PennState, National Agricultural Library, ASU Libraries, Tufts University, University of Manitoba, Montana State University, Art Institute of Chicago, Texas Digital Library, VirginiaTech, Smithsonian Institution, UC San Diego, The George Washington University, University of Pittsburgh ULS, CINECA, The University of Melbourne, Brown University, Penn University of Pennsylvania, Universiteit Gent, Harvard University, Georgia Tech Library, University of Alberta Libraries, ETH BIBLIOTHEK, University of Houston, Concytec, MITLibraries, UNSW Australia, Northeastern University, University of Hull, Rice Fendren Library, Sigma, Princeton University Library, The Ohio State University Libraries, University of Minnesota, LSE, Temple University Libraries, Rutgers University, University of Toronto, University of British Columbia, The University of Tennessee, Griffith University, TriCollege Libraries, IUPUI, University of Cambridge, UCSB, and KU Libraries. The University of Kansas is also listed at the bottom.



DURASPACE

# Useful Resources

Fedora 5.0 documentation

<https://wiki.duraspace.org/display/FEDORA5x>

Fedora wiki

<https://wiki.duraspace.org/display/FF>

Fedora mailing lists

<https://wiki.duraspace.org/display/FF/Mailing+Lists+etc>