Developer Guide

Fedora Repository (fcrepo4)

Browser Interface

The application can be accessed through the browser interface for demonstration purposes of the repository's basic functionality. Once deployed (see Deploying Fedora 4 Complete Guide) navigate to either *http://localhost:8080/rest* using the one click install or *http://localhost:8080/<war file name>/rest* using the tomcat or jetty servers.

REST API

The primary way for applications to work with a Fedora 4 repository is through the RESTful HTTP API.

- The Fedora 4 HTTP API is generally a RESTful API. HTTP methods like GET, PUT, POST and DELETE are implemented on most resource paths. The API also relies heavily on content negotiation to deliver context-appropriate responses, and a HATEOAS-driven text/html response (providing a decent GUI experience on top of the repository).

Indexing Message Listener

In tandem with the main fcrepo4 webapp is the fcrepo-jms-indexer-pluggable webapp. It can be configured to listen to the Fedora JMS topic, retrieve messages including pid and eventType, look up object properties, execute transformations, and pass this output to external applications providing search and triplestore functionality.

JMS Indexing Helper App

To support the differing needs for sophisticated, rich searching, Fedora 4 comes with a standard mechanism and integration point for indexing content in an external service. This could be a general search service such as Apache Solr or a standalone triplestore such as Fuseki or Blazegraph

Frameworks

Samvera and Fedora 4

Islandora and Fedora 4

Extending Fedora 4

Authorization Modules

Federation Connector

Storage Policy

Contributing to Fedora

Fedora is built with the contributions of developers just like you. If you've found a bug, or have a feature you want to add, we would love your help. Our Guide for New Developers includes links to our project resources (source code, issue tracker, mailing lists, etc.) and our Git workflow.

Maven Build System

Fedora and external module projects use Maven3 as the build system. See also: Guide for New Developers

Dynamic build properties

Integration tests use Build Helper Maven Plugin to dynamically generate configuration properties. This is used to automatically detect and configure unused ports for several server components. All dynamically generated properties should have the word "dynamic" in their name.

Example usage

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>build-helper-maven-plugin</artifactId>
  <configuration>
    <portNames>
      <portName>fcrepo.dynamic.test.port</portName>
      <portName>fcrepo.dynamic.jms.port</portName>
      <portName>fcrepo.dynamic.stomp.port</portName>
      <portName>jetty.dynamic.stop.port</portName>
    </portNames>
  </configuration>
</plugin>
```

e not defined in the <properties> section of the POMs. See:

Unable to locate Jira server for this macro. It may be due to Application Link configuration.