

LDP Implementation Report

Marmotta LDP Implementation Report

Relevant Documents

- [LDP spec](#)
- [LDP Use Cases and Requirements](#)
- [LDP Best Practices](#)
- [LDP Test Cases](#) (**not updated** yet with the latest changes)

Meetings

- [Feb 10, 2014](#)
- [Feb 11, 2014](#)

Implementation Report Revisions

- [LDP Implementation Report \(2014-09-16\)](#)
- [LDP Implementation Report \(2014-03-11\)](#)

Test Suite

- [LDP Marmotta Test Suite](#)

Implementation Restrictions

We agree on some restrictions:

- **Identifiers**
 - trailing slash is ignored (removed) for URI generation.
thus <http://example.com/foo/> and <http://example.com/foo> are treated as identical.
 - query part of an URL will never be used as part of a URI.
query parameters might be used for some LDP operations such as non-membership triples (tbd)
 - resources with fragments can't be addressed directly, but can be accessed via their container (URL without fragment)
- **Storage**
 - internally every LDPR will be stored in its own (Sesame) context (as could be interpreted from [sec 5.2.3.4](#))
 - that would allow us to extend the Resource concept beyond the common understanding in triplestores (just one level of outgoing triples)
- **Paging and Sorting** is not supported
Anyway the WG has resolved (Feb 17, 2014) to move out paging and ordering into a separate spec: [2014-02-17#r5](#)
- **ETags** for LDP-RS are *weak* and solely based on the `dc:terms:modified` value, while ETags for LDP-NR are based on the md5sum of the file content.
- Both **LDP-NR** and **LDP-IC** are supported
 - Our interpretation of Sec. 5.2.3.12 implements that POST of content types not managed by the platform (i.e., not suitable Rio parser registered) adds a LDP-RS as member of the container, linking (`dc:hasFormat/dc:isFormatOf`) to the actual LDP-NR (URI is build by appending the file extension according the content type)
- **Only LDP-BC** is supported, since current use cases do not justify the inclusion of LDP-DC or LDP-IC
- Sec. 4.2.1.6 (**rel="ldp:constrainedBy"-Link**) is fulfilled by pointing to [this page](#).
Link: `<http://wiki.apache.org/marmotta/LDPImplementationReport/2014-09-16>; rel="http://www.w3.org/ns/ldp#constrainedBy"`
- **PATCH** accepts [application/rdf-patch](#) only (LD Patch support will be discussed)
Patches with the predicate `ldp:contains` will result in a HTTP 409 Conflict

The compliance report for this version is available in the [Implementation Conformance Report](#).

Instructions

For accessing this experimental feature you'd need to follow these instructions:

First install all artifacts:

```
mvn install
```

And launch the webapp (at `launchers/marmotta-webapp/`) with the experimental features enabled:

```
mvn clean tomcat7:run -Pexperimental,kiwi,cleanall
```

Usage

Although not strictly required, it would be good to have some background what's happening under the hood with these commands. For that, please refer to the relevant documents listed above.

Add a source resource to a container

```
curl -i -X POST -d @src/test/resources/test.ttl -H "Content-Type: text/turtle" -H "Slug: test" http://localhost:8080/ldp/container1
curl -i -H "Accept: text/turtle" http://localhost:8080/ldp/container1/test
```

Add a binary resource to a container

```
curl -i -X POST --data-binary @src/test/resources/test.png -H "Content-Type: image/png" -H "Slug: test2" http://localhost:8080/ldp/container2
curl -i http://localhost:8080/ldp/container2/test2
curl -i -H "Accept: text/turtle" http://localhost:8080/ldp/container2/test2.png
curl -i -H "Accept: image/png" http://localhost:8080/ldp/container2/test2.png
```