

LinkedDataPlatform

A page to describe in how far Clerezza complias with W3C [LDP](#) best practices for a read-write Linked Data architecture.

Looking at the various requirements ("Yes" means that Clerezza satisfies all should and must level requirementst)

- 4.1.1 LDPR servers must at least be HTTP/1.1 conformant servers [HTTP11].

Yes.

- 4.1.2 LDPR servers must provide an RDF representation for LDPRs. The subject is typically the LDPR itself.

Yes. All resources in Clerezza can directly be dereferenced in all supported RDF formats with the exception of binary resources where the RDF description can be got at alternative URI. The reason for this is that system should behave consistently when an RDF and a non RDF file is uploaded so this are treated non-LDPR resources.

- 4.1.3 LDPR servers may host a mixture of LDPRs and non-LDPRs. For example, it is common for LDPR servers to need to host binary or text resources that do not have useful RDF representations.

Yes.

- 4.1.4 Clients can access a LDPR using multiple URLs, for example when DNS aliasing is used. A LDPR server must respond to each of those requests using a single consistent URL, a canonical URL, for the LDPR which may be found in the response's Location header and potentially also in the representation of the LDPR. Clients should use that canonical URL to identify the LDPR.

There's currently no support for the same resource to be retrieved from multiple URIs.

- 4.1.5 LDPR predicates should use standard vocabularies such as Dublin Core [DC-TERMS], RDF [RDF-PRIMER] and RDF Schema [RDF-SCHEMA], whenever possible. LDPRs should reuse existing vocabularies instead of creating their own duplicate vocabulary terms.

Yes. Clerezza uses mostly existing vocabularies like FOAF and SIOC for permissions.

- 4.1.6 LDPR predicates must use well-known RDF vocabularies as defined in section 4.8 Common Properties wherever a predicate's meaning matches one of them.

Uh? What's the difference to 4.1.5?

- 4.1.6.1 LDPRs must use the predicate `rdf:type` to represent the concept of type. The use of non-standard type predicates, as well as `dcterms:type`, is discouraged. [DC-RDF]

Yes.

- 4.1.7 LDPR representations should have at least one `rdf:type` set explicitly. This makes the representations much more useful to client applications that don't support inferencing.

Partially. Resources generated by Clerezza conform to the requirement but its possible to insert arbitrary data.

The description of any named resource in the content graph is served if a get request for that resource is sent against the clerezza instance and the URI is not handled by a dedicated Jax-RS resource and the resource if not of a type for which a Typehandler is registered. When addin triples to the content graph there's no mechanism enforcing that all named resources have an RDF type statement.

- 4.1.8 Predicate URIs used in LDPR representations should be HTTP URLs. These predicate URIs must identify LDPRs whose representations are retrievable. LDPR servers should provide an RDF Schema [RDF-SCHEMA] representation of these predicates.

Partially. Triples generated by Clerezza should conform to the requirement but its possible to insert arbitrary data.

As arbitrary triples can be uploaded there's no mechanism enforcing meaningful dereferenceability of the type statements.

- 4.1.9 LDPR representations must use only the following standard datatypes. RDF does not by itself define datatypes to be used for literal property values, therefore a set of standard datatypes based on [XMLSCHEMA11-2] and [RDF-PRIMER] are to be used: `xsd types + rdf-xml`

Partially. With the exception of the [WebId] module [verify] literals generated by Clerezza conform to the requirement but its possible to insert arbitrary data.

- 4.1.10 LDPRs must use at least one RDF triple to represent a link (relationship) to another resource. In other words, having the source resource's URI as the subject and the target resource's URI as the object of the triple represent the link (relationship) is enough and does not require the creation of an intermediate link resource to describe the relationship.

[REC!] This requirements seems to imply that the properties have a preferred direction, which contradict <http://dig.csail.mit.edu/breadcrumbs/node/72>. Clerezza does not currently enforce a link it is possible to add to the contentgraph a triple with a subject IRI not otherwise used and a literal object. The resources generated by clerezza do have at least a type statement, would this qualify as link?

- 4.1.11 LDPR servers may support additional standard representations. These could be other RDF formats, like N3 or NTriples, but non-RDF formats like HTML [HTML401] and JSON [RFC4627] would be likely be common.

Yes.

- 4.1.12 LDPRs may be created, updated and deleted using methods not defined in this document, for example through application-specific means, SPARQL UPDATE, etc. [SPARQL-UPDATE]

Yes.

- 4.1.13 LDPR server responses must contain accurate response ETag header values.

No.

- 4.2.1 LDPR servers must support the HTTP GET Method for LDPRs.

Yes

- 4.2.2 LDPR servers must provide an text/turtle representation of the requested LDPR.[TURTLE]

Yes

- 4.2.3 LDPR servers should provide a application/rdf+xml representation of the requested LDPR.[RDF-SYNTAX]

Yes.

- 4.2.4 In the absence of special knowledge of the application or domain, LDPR clients must assume that any LDPR may have multiple values for rdf:type.

Yes. (For the limited code where clerezza acts as client)

- 4.2.5 In the absence of special knowledge of the application or domain, LDPR clients must assume that the rdf:type values of a given LDPR may change over time.

Partially. DbPedia client makes no assumption on persistence but Resource providers doesn't update.

- 4.4.1 If HTTP PUT is performed on an existing resource, LDPR servers must replace the entire persistent state of the identified resource with the entity representation in the body of the request. The only recognized exception are the properties dcterms:modified and dcterms:creator that are never under client control - LDPR servers must ignore any values of these properties that are provided by the client. Any LDPR servers that wish to support a more sophisticated merge of data provided by the client with existing state stored on the server for a resource must use HTTP PATCH, not HTTP PUT.

[SPEC: This should probably say LPDR instead of resource in the first sentence]. No. Clerezza will treat any entity body of a PUT request as a binary resource, that will be stored encoded in a literal of the content graph so that subsequent get request will return the same entity body as in the put request.

- 4.4.2 LDPR clients should use the HTTP If-Match header and HTTP ETags to ensure it isn't modifying a resource that has changed since the client last retrieved its representation. LDPR servers should require the HTTP If-Match header and HTTP ETags to detect collisions. LDPR servers must respond with status code 412 (Condition Failed) if ETags fail to match if there are no other errors with the request. [HTTP11]

No

- 4.4.3 LDPR clients should always assume that the set of predicates for a resource of a particular type at an arbitrary server is open in the sense that different resources of the same type may not all have the same set of predicates in its triples, and the set of predicates that are used in the state of a resource is not limited to any pre-defined set.

Yes. For the limited number of Clerezza modules that act as client.

- 4.4.4 LDPR clients should assume that a LDPR server could discard triples whose predicates the server does not recognize or otherwise chooses not to persist. In other words, LDPR servers may restrict themselves to a known set of predicates, but LDPR clients must not restrict themselves to a known set of predicates when their intent is to perform a later HTTP PUT to update the resource.

N/A.

- 4.4.5 A LDPR client must preserve all triples retrieved using HTTP GET that it doesn't change whether it understands the predicates or not, when its intent is to perform an update using HTTP PUT. The use of HTTP PATCH instead of HTTP PUT for update avoids this burden for clients [RFC5789].

N/A. A client that could use this principle is the javascript Discobit editor, it does not currently use PUT (but a revoke/assert via post)

- 4.4.6 LDPR servers may choose to allow the creation of new resources using HTTP PUT.

[SPEC: should probably be LDPR instead of Resource]

- 4.4.7 LDPR servers should allow clients to update resources without requiring detailed knowledge of server-specific constraints. It is common for LDPR servers to put restrictions on representations – for example, the range of rdf:type, datatypes of predicates and number of occurrences of predicates in triples, but servers should minimize those restrictions. In other words, LDPR servers need to enable simple modification of LDPRs. Enforcement of more complex constraints will greatly restrict the types of clients that can modify resources. For some server applications, excessive constraints on modification of resources may be required.

Yes, Clerezza poses no restriction at all, arbitrary triples can be added

- 4.5.1 LDPR servers must remove the resource identified by the Request-URI. After a successful HTTP DELETE, a subsequent HTTP GET on the same Request-URI must result in a 404 (Not found) or 410 (Gone) status code, until another resource is created or associated with the same Request-URI.

No. DELETE is not supported

- 4.5.2 LDPR servers may alter the state of other resources as a result of an HTTP DELETE request. For example, it is acceptable for the server to remove triples from other resources whose subject or object is the deleted resource. It is also acceptable and common for LDPR servers to not do this – behavior is server application specific.

N/A

- 4.6.1 LDPR servers must support the HTTP HEAD method.

Yes but not very smart (same processing as for GET).

- 4.6.2 LDPR servers must indicate their support for HTTP Methods by responding to a HTTP HEAD request on the LDPR's URL with the HTTP Method tokens in the HTTP response header "Allow".

?

- 4.7.1 LDPR servers may implement HTTP PATCH to allow modifications, especially partial replacement, of their resources. [RFC5789] No minimal set of patch document formats is mandated by this document.

Not implemented

- 4.7.2 LDPR servers should allow clients to update resources without requiring detailed knowledge of server-specific constraints. It is common for LDPR servers to put restrictions on representations – for example, the range of rdf:type, datatypes of predicates and number of occurrences of predicates in triples – but server enforcement of detailed, domain-specific constraints will greatly restrict the types of clients who can update resources.

N/A

[SPEC: wondering that section 5 is non normative, despite the many MUST and SHOULD statements]

Sections 5.3-5.9 are normative.