



Linked Data Management

Capítulo 18: Usando leitura e escrita de dados ligados para integração de aplicativos

Angélica Valiati Maran Derenevick
08 de Novembro de 2016



INTRODUÇÃO

- 
- Linked Data é uma tecnologia utilizada para publicações de dados na Word Wide Web (WWW);
 - Uma característica comum de aplicações que utilizam o Linked Data é que eles utilizam somente leitura, ou seja, dessa forma, normalmente, os dados são carregados e armazenados dentro de um banco de dados local e posteriormente disponibilizado na web ou em um aplicativo local, em modo de leitura;
 - Possui características únicas, como ser distribuído ou escalável;
 - Possui uma plataforma de Linked Data em desenvolvimento, o LDP.



INTEGRAÇÃO

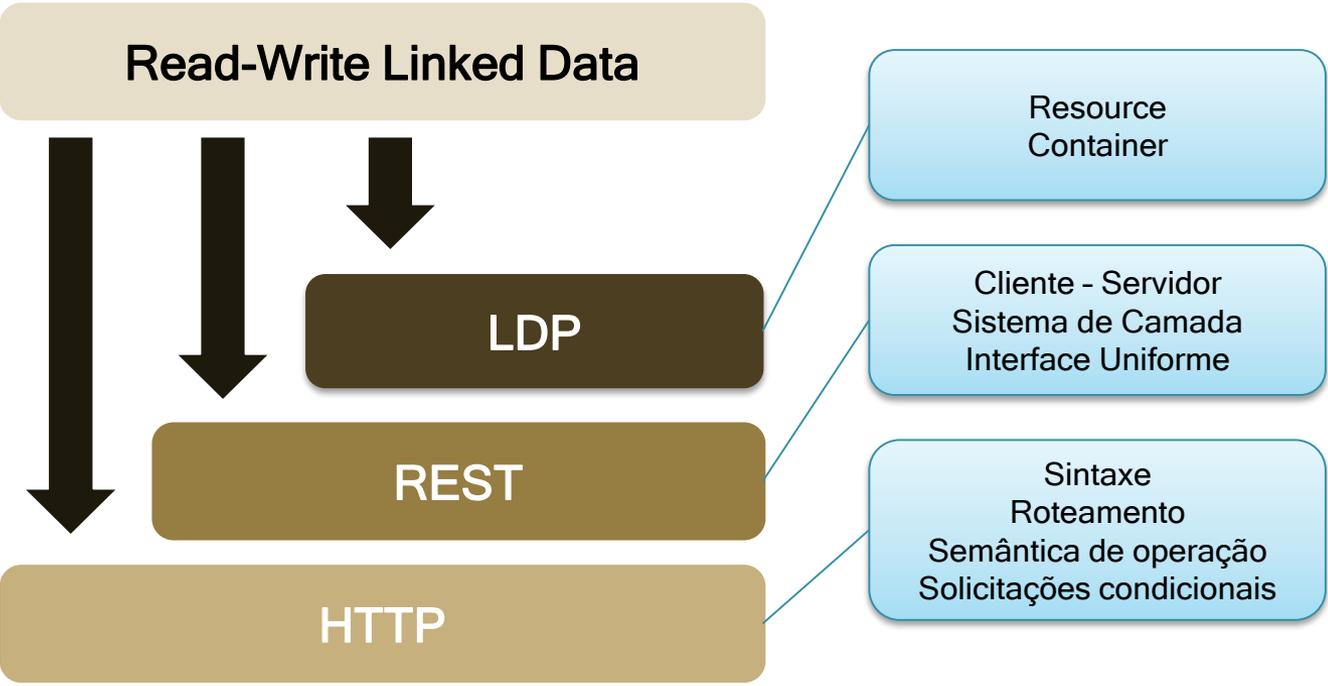
- 
- Problema:
 - Conexão entre aplicativos;
 - Integração de dados.
 - Desafios
 - Heterogeneidade sintática;
 - Heterogeneidade estrutural;
 - Heterogeneidade semântica.



LDP

- 
- LDP - Linked Data Platform;
 - LDP é uma iniciativa da W3C (World Wide Web Consortium)
 - O LDP complementa outros padrões de Linked Data, como o SPARQL;
 - LDP traz características de integração de dados de RDF para RestFul;
 - LDP fornece uma base para a integração de aplicativos usando read-write Linked Data.

- 
- LDPR (Linked Data Platform Resource) - um conjunto de técnicas HTTP e padrão RDF. Auxilia na comunicação de leitura e escrita de dados entre cliente e servidor;
 - LDPC (Linked Data Platform Containers) - é uma especialização do LDPR, atua como um recurso de coleção que ajuda a organizar LDPRs e criando novas LDPRs como seus membros;
 - Cliente - Um aplicativo ou programa que estabelece conexão e envia requisições ao servidor;
 - Servidor - Um aplicativo ou programa que aceita a conexão para atender solicitações enviando respostas de retorno.





LDPR

- 
- LDPR são recurso HTTP de creating (HTTP POST), modifying (HTTP PUT), deleting (HTTP DELETE) e reading (HTTP GET);
 - LDPRs também podem ser criados, atualizados e excluídos usando SPARQL;
 - Usam RDF para definir seu estado;
 - LDPR pode fazer request para as sintaxe Turtle, XML, JSON
 - LDP-Client utiliza a detecção de colisão e otimista em update.



Requisição do Cliente

- 1 - GET /container1/member1 HTTP/1.1
- 2 - Host: example.org
- 3 - Accept: text/turtle

Resposta do Servidor

- 1 - # Removed HTTP headers to save some space
- 2 - @prefix dcterms: <http://purl.org/dc/terms/>.
- 3 - @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
- 4 - @prefix ldp: <http://www.w3.org/ns/ldp#>.
- 5 - @prefix bt: <http://example.org/vocab/bugtracker#>.
- 6 -
- 7 - <http://example.org/product1/defects/bug1>
- 8 - A bt:Defect;
- 9 - dcterms:title "Product crashes when shutting down.";
- 10 - dcterms:creator </app/users/johndoe>;
- 11 - dcterms:created "2013-05-05T10:00"^^xsd:dateTime
- 12 - bt:isInState bt:StateNew .

Criando e Adicionando um recurso

Requisição do Cliente

- 1 - POST /product1/defects/ HTTP/1.1
- 2 - Host: example.org
- 3 - Content-type: text/turtle
- 4 - Content-length: 227
- 5 -
- 6 - @prefix dcterms: <http://purl.org/dc/terms/>.
- 7 - @prefix bt: <http://example.org/vocab/bugtracker#>.
- 8 -
- 9 - <>
- 10 - A bt:Defect;
- 11 - dcterms:title "Product stops functioning after a while.";
- 12 - dcterms:creator </app/users/johndoe>.

Criando e Adicionando um Recurso

Resposta do Servidor

1 HTTP/1.1 201 CREATED

2 Content-Location: <http://exmple.org/product1/defects/bug4>



Atualizar e Deletar um recurso

- Atualizar
 - PUT: Precisa enviar toda a requisição
 - PATCH: Envia para o servidor apenas o que deseja alterar.

- Deletar
 - DELETE



LDPC

- 
- É um conjunto de recursos, que são chamados de contêiners, e esse conjunto faz parte de uma LDPR;
 - Um LDPC é um recurso que é um LDPR do tipo `ldp:container`;
 - Os clients podem recuperar a lista de recursos existentar em um LDPC;
 - Um recurso não precisa ser um LDPR com uma representação RDF para fazer um POST para um LDPC;

- 
- Um mesmo recurso pode aparecer em vários containers;
 - Sua representação é de RDF padrão usando o predicado `rdfs:member` ou `ldp:membershipPredicate`
 - Possuem duas finalidades:
 - Listar os recursos de seus membros;
 - Fornecer um mecanismo para a criação de novos recursos de membros.
 - Os métodos HTTP HEAD ou OPTIONS são utilizados para recuperar as propriedades de um membro que não faz parte da LDPC.



Requisição do Cliente

- 1 - GET /product1/defects/ HTTP/1.1
- 2 - Host: example.org
- 3 - Accept: text/turtle

Resposta do Servidor

- 1 - # HTTP headers not displayed
- 2 - @prefix dcterms: <http://purl.org/dc/terms/>.
- 3 - @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
- 4 - @prefix ldp: <http://www.w3.org/ns/ldp#>.
- 5 -
- 6 - <http://example.org/product1/defects>
- 7 - A ldp:Container;
- 8 - ldp:membershipSubject <>;
- 9 - ldp:membershipPredicate rdfs:member;
- 10 - dcterms:title "Product1 defects";
- 11 - rdfs:member
- 12 - <http://example.org/product1/defects/bug1> ,
- 13 - <http://example.org/product1/defects/bug2> ,
- 14 - <http://example.org/product1/defects/bug3> .

Criando e Adicionando um Recurso

Requisição do Cliente

- 1 - POST /container HTTP/1.1
- 2 - Host: example.org
- 3 - Content-type: text/turtle
- 4 - Content-length: 324
- 5 -
- 6 - @prefix dcterms: <http://purl.org/dc/terms/>.
- 7 - @prefix ldp: <http://www.w3.org/ns/ldp#>.
- 8 -
- 9 - <>
- 10 - A ldp:Container;
- 11 - dcterms:title "A very simple container".

Criando e Adicionando um Recurso

Resposta do Servidor

1 - HTTP/1.1 201 CREATED

2 - Content-Location: <http://example.org/container/newcontainer>



Atualizar e Deletar um recurso

- Atualizar (geralmente é atualizada pelo cliente por POSTing)
 - PUT: Precisa enviar toda a requisição
 - PATCH: Envia para o servidor apenas o que deseja alterar.
- Deletar
 - DELETE

Adicionando um Recurso Binário

Requisição do Cliente

- 1 - POST /mycontainer HTTP/1.1
- 2 - Host: example.org
- 3 - Content-type: image/png
- 4 - Content-length: 1048
- 5 -
- 6 - [binary content not displayed]

Adicionando um Recurso Binário

Resposta do Servidor

- 1 - HTTP/1.1 201 CREATED
- 2 - Content-Location: <http://example.org/mycontainer/myimage>
- 3 - Link: <<http://example.org/mycontainer/myimage-info>>;rel=describes

Mostrando o Recurso Binário

Requisição do Cliente

- 1 - GET /mycontainer HTTP/1.1
- 2 - Host: example.org
- 3 - Accept: text/turtle

Adicionando um Recurso Binário

Resposta do Servidor

```
1 # HTTP headers not displayed
2 @prefix dcterms: <http://purl.org/dc/terms/>.
3 @prefix wdrs: <http://www.w3.org/2007/05/powder-s#> .
4
5 <http://example.org/mycontainer/myimage-info>
6 dcterms:rightsHolder "John Z Smith"
7 dcterms:rights "All rights reserved."
8 dcterms:created "2013-04-16T10:15:32.15-08:00";
9 dcterms:format <http://purl.org/NET/mediatypes/image/png>;
10 dcterms:description "Screen capture of the crash".
11
12 <http://example.org/mycontainer/myimage>
13 wdrs:describedBy <http://example.org/mycontainer/myimage-info>
```

Paginação

```
1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
2 @prefix dcterms: <http://purl.org/dc/terms/>.
3 @prefix ldp: <http://www.w3.org/ns/ldp#>.
4 @prefix bt: <http://example.org/vocab/bugtracker#>.
5
6 <http://example.org/product1/defects>
7 A ldp:Container;
8 dcterms:title "The defects of Product #1";
9 ldp:membershipSubject <http://example.org/product1>;
10 ldp:membershipPredicate bt:hasBug.
11
12 <http://example.org/product1/defects?firstPage>
13 A ldp:Page;
14 ldp:pageOf <http://example.org/product1/defects>;
15 ldp:nextPage <http://example.org/product1/defects?p=2>.
```



CONCLUSÃO

- 
- Além da padronização para alterar e deletar recursos, esta em estudo a rastreabilidade entre relacionamentos, análise de impacto e recursos de consulta profunda;
 - LDP permitirá uma integração mais ampla entre os Linked Data.



PERGUNTAS

- 
- 1) Para ocorrer a integração entre os dados da LDP com a web, possui alguns desafios, quais são esses desafios e com quais recursos do Linked Data poderá tratar a integração?
 - 2) Explique sobre a LDP, as funcionalidades da LDPR e LDPC.

angelicavaliati@gmail.com



OBRIGADA!