History View Links to this page 2012 September 12 | 09:33 am

Status: Unapproved and INCOMPLETE DRAFT

Please provide feedback to the OSLC Mailing List.

Contents

Introduction

There are many motivating cases for supporting a writable Web, existing approaches have focused heavily on the usage of HTTP PUT as the method for updating resources. As experience has shown, this method for updating resources comes with some significant limitations and performance costs. This has motivated the creation of the IETF Proposed Standard titled "PATCH Method for HTTP" http://tools.ietf.org/html/rfc5789 which introduces a new HTTP verb called PATCH. RFC5789 defines the semantics of the new HTTP verb PATCH but does not endorse a PATCH representation.

Motivated by our use cases and requirements, partial update can be simple. Given a resource URL, triples can be added or removed.

This document recommends to use HTTP PATCH to implement partial update, provides a patch model based on a subset of SPARQL Update http://www.w3.org/Submission/SPARQL-Update, includes recommended PATCH document formats based on existing RDF formats and provides examples of how this approach works. If additional use cases need to be satisfied, a full SPARQL Update solution can be used.

Comment [JA1]: The latest public WD is 2/2012, versus 7/2008 on the submission.

Comment [JA2]: W3C site returns a 300 on this. It's preferred link is sans the .html suffix.

Use Cases & Requirements

Use Cases:

- Given a URI for a Change Request resource, I'd like to add a link to a related test plan. I don't want to remove any existing links and if the link already exists, then do nothing.
- Given a URI for a Test Case resource, I'd like to replace all the blocking defect links with the set of the defect links I have (from a picked list of existing defects from various providers).
- Given a URI for a Test Case resource, I'd like to remove only one blocking defect link whether it exists or not.
- Given a URI for a ChangeRequest resource, I'd like to update the link label associated with the related test plan link.

Comment [JA3]: As written, a solution that works for only 1 predicate per request would be sufficient. I'm sure we need more than that for our products, and the examples seem to agree. Render it explicit.

Requirements:

- Add triples to an existing resource
- Delete triples from a resource
- Updating the object(s) of a triple(s) in a resource
- Update reified statements on links, such as link labels
- Guidance on collision detection (usage of If-Match ETag and 412 responses)
- Reduce possibility of errors in updates because partial updates are safer than complete replace operations
- Reduce bandwidth used and load on server because partial updates are smaller

Comment [JA4]: Blank nodes? Hash URIs? Should be listed, even if only to say "no known requirements" so it's clearer what this does/not cover.

Terminology

Partial update - The HTTP operation to modify only a subset of the triples for a given resource.

Patch - See **Partial update**

HTTP PATCH

Model

OSLC clients **SHOULD** use the HTTP PATCH method to apply a partial update. For servers that support partial update but not HTTP PATCH, an OSLC client **SHOULD** use HTTP POST to send the request to the resource URI and will use the HTTP header `X-Method-Override: PATCH` to indicate that the patch is a PATCH. If a server does not support PATCH on a resource, then it **SHOULD** respond with HTTP status code 405 (Method not allowed). If a server does support PATCH on a resource, then it **SHOULD** respond to a HTTP OPTIONS operation with at least the token value "PATCH" for the response header `Allow`.

Comment [JA5]: SHOULD ? MUST?

OSLC Partial Update servers **MUST** treat a single HTTP PATCH request as a single resource update operation and not as independent insert and delete operations.

OSLC Partial Update servers **MUST** treat HTTP PATCH request, on an existing resource identified by the canonical Request-URI, as an atomic modification to that resource. The patch model is limited to web resources that are identified by their URI that maps to the subject of triples. HTTP PATCH of web resource is symmetric with the usage HTTP verbs of GET, PUT and DELETE which operate on a web resource identified by the Request-URI.

OSLC Partial Update servers **MAY** choose to allow the creation of new resources using HTTP PATCH.

Each OSLC client~~s~~ **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. OSLC servers **SHOULD** require the HTTP If-Match header and HTTP ETags to detect collisions. OSLC servers **MUST** respond with status code 412 (Condition Failed) if ETags fail to match if there are no other errors with the request. [RFC2616]

A pre-defined named graph that represents the nodes to match or to insert is sufficient.
The semantics of the PATCH model are based on SPARQL Update. Namely the subset of SPARQL Update is limited to:

- Subset of INSERT DATA
- Subset of DELETE DATA

This model does not handle patching resources across multiple graphs.

Format

To use HTTP PATCH, a client must provide ~~with the request~~ a ~~PATCH~~ document with the request ~~format that is able convey~~describing the changes to be ~~match~~ made~~in the partial update~~. This guidance recommends the use of named graph (quad) formats for the PATCH document. In this way, it does not require a new format or parsers to be created. Some quad formats available include TriG (Turtle-based) and TriX (XML-based).

Discussion: We got confirmation on this approach previously, question is: Should we require a minimum format for servers to support?

Proposed RESOLUTION: Since Core 2.0 requires RDF/XML it would make sense that those implementations would use TriX. For Core 3.0 (W3C LDP based), which requires Turtle, it would make sense that those implementations would use TriG.

Add new property values to a resource

Named graph URI: http://open-services.net/ns/core#insertGraph

To add new triples to a resource, HTTP PATCH them in Patch Document (see examples below) to the resource URI with HTTP Header `Content-Type: application/x-trig`. In this case we

Comment [JA6]: Not seeing what this word communicates. Where is this concept defined in specification space?

Comment [JA7]: Struggling to see how this sentence differs in intent from the preceding one.

Comment [JA8]: Does this mean it will not work for resources identified using hash URIs? If implementations would be so restricted (their alternative being some "less minimal" syntax), want to call that out.

Comment [JA9]: Is the client interaction the same as POST-create? Point to specification saying so/describing it if that's the intent. 5789 does not really address this case, fwiw. I think we want to say it works just like post-create from the client's interaction standpoint… if they get a 201 response, then a resource was created, and its URL is in the Location header.

Comment [JA10]: I thought one of the use cases was "I don't care what's there now, I just want to add a link" which does not obviously require etags. Adding a retrieval step has been one of the reasons implementers have given for not PATCHing. The naughty ones violate HTTP by treating PUT as partial replace.

Comment [JA11]: I think you mean all triples MUST have subjects that map to the same HTTP Request-URI. Although if blank nodes are truly handled (I have my doubts, voiced later) the story is more complicated.

Bottom line, you need to tell me enough here so that I could take a set of triples and know reliably whether or not they are one graph.

Comment [JA12]: I don't see licenses in those documents. Do we know they are safely reusable? Or are these things we could use but are not (reading the sentence carefully, this could be the case at this point in the reading)

Comment [JA13]: Reading further, OK the proposal is to re-use this work. So the licensing question is real.

use the SPARQL Update INSERT DATA command to specify the statements (triples) we wish to add.

OSLC Partial Update servers **MUST** support inserting triples using the INSERT DATA command as defined by SPARQL Update.

SPARQL Update mapping

ToDo: Provide mapping details

Comment [JA14]: A little fuzzy to me what's missing. 1 line example maybe?

Example

Here's an example of what a client might post to add new property values of `dcterms:description` and `ex:status` to a file or directory resource with URL of `http://example.com/tasks/27`.

Before PATCH

```
<http://example.com/tasks/27> <http://purl.org/dc/terms/modified> "1996-09-
16T08:42:11.265Z" .
<http://example.com/tasks/27> <http://example.com/ns/terms/relatedTo>
<http://example.com/resources/1> .
```

PATCH Document

```
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix ex: <http://example.com/ns/terms#>.
<http://open-services.net/ns/core#insertGraph> {
  <http://example.com/tasks/27> dcterms:description "Need to update Website
owners as of May 2012" .
  <http://example.com/tasks/27> ex:status "PENDING" .
}
```

Comment [JA15]: Missing in the AFTER below

After PATCH

```
<http://example.com/tasks/27> <http://purl.org/dc/terms/description> "Website
owners as of May 2012" .
<http://example.com/tasks/27> <http://example.com/ns/terms/status> "PENDING"
.
<http://example.com/tasks/27> <http://purl.org/dc/terms/modified> "2012-05-
21T17:02:13.265Z" .
<http://example.com/tasks/27> <http://example.com/ns/terms/relatedTo>
<http://example.com/resources/1> .
```

And here's an example of adding a new `dcterms:contributor` value to a resource with URL of `http://example.com/tasks/27`. Note that the property has a value that is a blank-node of type `foaf:Person`. Also note that the subject of some of the triples is not the resource, but instead a blank-node reference.

```
<http://open-services.net/ns/core#insertGraph> {
```

```
  <http://example.com/tasks/27> <http://purl.org/dc/terms/contributor>
_:bnode1 .
  _:bnode1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/person> .
  _:bnode1 <http://xmlns.com/foaf/0.1/name> "Fernando Jones" .
}
```

Deleting triples from a resource

Named graph URI: http://open-services.net/ns/core#deleteData

To delete triples from a resource, a client will PATCH a SPARQL Update document (see below) to the resource URI with HTTP Header `Content-type=application/sparql-update`. In this case we use the SPARQL Update DELETE DATA command, which requires at least one triple.

OSLC server implementations that support for Partial Update **MUST** support removing triples using the DELETE DATA command as defined by SPARQL Update.

Since there is not a way to identify blank nodes, deletion of blank nodes is not defined. SPARQL Update could be used to remove blank nodes.

Comment [JA16]: Graph, based on later text

Comment [JA17]: Versus Trix/Trig, which would match preceding and patch doc below.

Comment [JA18]: Not clear how you could update without this either.

Comment [JA19]: Are hash URIs covered?

Comment [JA20]: This confuses readers – makes it sound like doing so MIGHT be part of the proposal, when we just want to know what IS The Proposal, what does It do/not do. Alternatives that could be used to form competing proprosals should be separated out not co-mingled.

SPARQL Update mapping

ToDo: Provide mapping details

Example

Here's an example of what a client would patch to remove a triple from the resource at URL `http://example.com/tasks/27`.

Before PATCH

```
<http://example.com/tasks/27> <http://purl.org/dc/terms/description> "Website
owners as of May 2012" .
<http://example.com/tasks/27> <http://example.com/ns/terms/status> "PENDING"
.
<http://example.com/tasks/27> <http://purl.org/dc/terms/modified> "2012-05-
21T17:02:13.265Z" .
<http://example.com/tasks/27> <http://example.com/ns/terms/relatedTo>
<http://example.com/resources/1> .
```

PATCH Document

```
@prefix ex: <http://example.com/ns/terms#>.
<http://open-services.net/ns/core#deleteGraph> {
   <http://example.com/tasks/27> ex:relatedTo
<http://example.com/resources/1> .
}
```

After PATCH

```
<http://example.com/tasks/27> <http://purl.org/dc/terms/description> "Website
owners as of May 2012" .
<http://example.com/tasks/27> <http://example.com/ns/terms/status> "PENDING"
.
<http://example.com/tasks/27> <http://purl.org/dc/terms/modified> "2012-05-
21T17:27:42.265Z" .
```

Updating the object of a triple

To update objects of triples for a resource, a client will send a HTTP PATCH request (see below) to the resource Request-URI with HTTP Header `Content-type: application/x-trig` or `Content-type: application/trix+xml`. In this case we use a SPARQL Update DELETE DATA command followed by an INSERT DATA command to specify the triples to be updated, the old triples and the new triples.

OSLC Partial Update servers **MUST** support updating triples using the DELETE DATA and INSERT DATA commands as defined by SPARQL Update.

OSLC Partial Update servers **MAY** allow only the ~~the~~ update of only those triples who~~'~~se subject is the HTTP Request-URI.

SPARQL Update mapping

ToDo: Provide mapping details

Example

Before PATCH

```
TBD
```

PATCH Document

```
@prefix ex: <http://example.com/ns/terms#>.
<http://open-services.net/ns/core#deleteGraph> {
  <http://example.com/files/file1> ex:parent
<http://example.com/resources/1>.
  <http://example.com/files/file1> ex:favoriteTeam
<http://example.com/teams/brazil>.
}
<http://open-services.net/ns/core#~~deleteGraph~~insertGraph> {
  <http://example.com/files/file1> ex:parent
<http://example.com/resources/2>.
  <http://example.com/files/file1> ex:favoriteTeam
<http://example.com/teams/paraguay>.
}
```

After PATCH

```
TBD
```

Example - update with blank nodes (link label)

Before PATCH

```
@prefix ex: <http://example.com/bugtracker> .
@prefix oslc: <http://open-services.net/ns/core#> .
@prefix oslc_cm: <http://open-services.net/ns/cm#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://example.com/bugs/2314>
   a oslc_cm:ChangeRequest ;
   dcterms:identifier " 00002314 " ;
   oslc:shortTitle "Bug 2314" ;
   dcterms:title "Invalid installation instructions" ;
   oslc:instanceShape <http://example.com/shapes/defect> ;
   dcterms:description "Invalid installation instructions indicating invalid
patches to be applied." ;
   oslc:discussedBy <http://example.com/bugs/2314/discussion> ;
   oslc_cm:relatedChangeRequest <http://myserver/mycmapp/bugs/1235> ,
      <http://remoteserver/mycmapp/defects/abc123> ;
   ex:priority " Medium " ;
   ex:severity " Normal " .

_:b1 dcterms:title "A bad link title";
   rdf:object <http://myserver/mycmapp/bugs/1235>;
   rdf:predicate oslc_cm:relatedChangeRequest;
   rdf:subject <http://example.com/bugs/2314>;
   a rdf:Statement.
```

Patch document as TriG

```
@prefix ex: <http://example.com/bugtracker>.
@prefix oslc: <http://open-services.net/ns/core#>.
@prefix oslc_cm: <http://open-services.net/ns/cm#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dcterms: <http://purl.org/dc/terms/>.

oslc:deleteGraph {
_:b1 dcterms:title "A bad link title";
   rdf:object <http://myserver/mycmapp/bugs/1235>;
   rdf:predicate oslc_cm:relatedChangeRequest;
   rdf:subject <http://example.com/bugs/2314>;
   a rdf:Statement.
}
oslc:insertGraph {
_:b1 dcterms:title "A very good link title";
   rdf:object <http://myserver/mycmapp/bugs/1235>;
   rdf:predicate oslc_cm:relatedChangeRequest;
   rdf:subject <http://example.com/bugs/2314>;
   a rdf:Statement.
}
```

Comment [JA21]: If the original had a second blank node that differed only in its ID, how would you predictably select one versus the other?

Mapped to SPARQL Update

```
prefix ex: <http://example.com/bugtracker>
prefix oslc: <http://open-services.net/ns/core#>
prefix oslc_cm: <http://open-services.net/ns/cm#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix dcterms: <http://purl.org/dc/terms/>

DELETE {?s dcterms:title ?o . }
INSERT  {?s dcterms:title "A very good link title". }
WHERE {
   ?s rdf:object <http://myserver/mycmapp/bugs/1235> .
   ?s rdf:predicate oslc_cm:relatedChangeRequest .
   ?s rdf:subject <http://example.com/bugs/2314> .
   ?s dcterms:title ?o .
}
```

After PATCH

```
TBD
```

Example - remove the label

Before PATCH

```
TBD
```

Patch document as TriG

```
@prefix oslc: <http://open-services.net/ns/core#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dcterms: <http://purl.org/dc/terms/>.

oslc:deleteGraph {
_:b1 dcterms:title "A very good link title";
   rdf:object <http://myserver/mycmapp/bugs/1235>;
   rdf:predicate oslc_cm:relatedChangeRequest;
   rdf:subject <http://example.com/bugs/2314>;
   a rdf:Statement.
}
```

Mapped to SPARQL Update

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix dcterms: <http://purl.org/dc/terms/>

DELETE  {?s dcterms:title "My special link title". }
WHERE {
   ?s rdf:object <http://myserver/mycmapp/bugs/1235> .
   ?s rdf:predicate oslc_cm:relatedChangeRequest .
   ?s rdf:subject <http://example.com/bugs/2314> .
   ?s dcterms:title ?o .
}
```

After PATCH

```
TBD
```

Note: a clever server may decide to remove the additional reification triples if decides there are no other interesting facts that they hold.

Related work

- [Jazz Foundation item #10252 - Partial Resource Update](https://jazz.net/jazz/web/projects/Jazz Foundation#action=com.ibm.team.workitem.viewWorkItem&id=102252)
- [Jazz Foundation item #101430 - Support for Bulk Operations](https://jazz.net/jazz/web/projects/Jazz Foundation#action=com.ibm.team.workitem.viewWorkItem&id=101430)
- TurtlePatch
- Linked Data Basic Profile - W3C Member Submission
- Talis changeset proposal

Open Items

- ~~Define method to "wildcard" delete/remove triples. For example triples that just match a subject and predicate.~~ No longer needed, not good use case requiring it. Recommend getting all, putting all in a delete statement and then patching OR using SPARQL Update.
- ~~Define the transactionality of the set commands, is it important? should the client control it? Specifically, do if a delete operation fails (user doesn't have enough access to perform the operation), do you continue with the insert?~~ Done - rely on SPARQL Update spec for semantics
- ~~Validate use cases and requirements~~ – Done
- ~~Produce better examples that show the resource before and after the patch~~ – Done
- ~~Should we support various SPARQL Update format shorthands such as: PREFIX, ",", ";", etc. ?~~ – Done, just add PATCH
- ~~With non-RDF repos, ensure that model works with properties like dc:title where occurs=exactly-one (see RTC WorkItems, CQrecords). What happens when receive: DELETE only (assume it would "null" the value), INSERT only (fail request), DELETE/INSERT (should the DELETE object match *exactly* before insert? – Yes)~~ – Done
- ~~PATCH'ing link labels (reified statements), how do we do this since we don't support blank nodes in delete? Proposal: invent new syntax for this use case and not have to support full SPARQL Update Insert/Delete/Where:~~ Agreed to use named graph (quad) format instead. Previous proposal included something like:

```
PREFIX ex: <http://example.com/ns/terms#>
DELETE DATA {
     [ <http://example.com/tasks/27> ex:relatedTo
<http://example.com/resources/1> ]
        dcterms:title "My old link label title".
}
INSERT DATA {
```

```
     [ <http://example.com/tasks/27> ex:relatedTo
<http://example.com/resources/1> ]
        dcterms:title "NEW NEW NEW Link title".
}
```

Items needed to close this work

- resolve all open issues
- insert rules for mapping to SPARQL Update

Named Graph Approach

Idea - use RDF representation itself, to represent the patch format. In order to achieve this, the patch format will be treated like a set of well-known named graphs.

- oslc:deleteGraph to indicate the graph of triples that should be purged, needs to match exactly
- oslc:insertGraph to indicate the graph of triples that should be added. Perhaps it would be better to call them *PartialGraph as they are incomplete.

Pros:

- This has the outstanding quality of not requiring any new (other than named graph) parsers to accomplish this.
- Handles blank nodes (to some extent)
- Meets OSLC's basic use cases

Cons:

- Doesn't support patching named graphs
- Doesn't have a model+syntax for future extension of graph pattern matching

Category:Supporting Documents

Categories

- Supporting Documents

Comment [JA22]: I was looking for this content up front, probably after UC+Requirements

Comment [JA23]: Define that or dump it

Comment [JA24]: Unhelpful – list what it does/not cover