

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 18, 2013

E. Wilde
EMC
M. Nottingham
Akamai
January 14, 2013

A Media Type for XML Patch Operations draft-wilde-xml-patch-00

Abstract

The XML Patch media type "application/xml-patch" defines an XML document structure for expressing a sequence of patch operations that are applied to an XML document. The XML Patch document format's foundations are defined in [RFC 5261](#), this specification defines a document format and a media type registration, so that XML Patch documents can be labeled with a media type, for example in HTTP conversations.

Note to Readers

This draft should be discussed on the apps-discuss mailing list [\[8\]](#).

Online access to all versions and files is available at github [\[9\]](#).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 18, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	IANA Considerations	3
3.	Patch Document Format	4
4.	Patch Examples	5
5.	Security Considerations	6
6.	References	6
6.1.	Normative References	6
6.2.	Informative References	6
Appendix A.	XSD from RFC 5261	7
	Authors' Addresses	9

1. Introduction

The Extensible Markup Language (XML) [1] is a common format for the exchange and storage of structured data. HTTP PATCH [6] extends HTTP [7] with a method to perform partial modifications to resources. HTTP PATCH requires that patch documents are being sent along with the request, and it is therefore useful if there are standardized patch document formats (identified by media types) for popular media types.

The XML Patch media type "application/xml-patch" is an XML document structure for expressing a sequence of operations to apply to a target XML document, suitable for use with the HTTP PATCH method. Servers can freely choose which patch formats they want to accept, and "application/xml-patch" could be a simple default format that can be used unless a server decides to use a different (maybe more sophisticated) patch format for XML.

The format for patch documents is based on the XML Patch Framework defined in RFC 5261 [2]. While RFC 5261 does define a concrete syntax as well as the media type "application/patch-ops-error+xml" for error documents, it only defines XSD types for patch operations, and thus the concrete document format and the media type for patch operations are defined in an XSD defined in this specification.

2. IANA Considerations

The Internet media type [3] for an XML Patch Document is application/xml-patch.

Type name: application

Subtype name: xml-patch

Required parameters: none

Optional parameters: Same as charset parameter for the media type "application/xml" as specified in RFC 3023 [1].

Encoding considerations: Same as encoding considerations of media type "application/xml" as specified in RFC 3023 [1].

Security considerations: This media type has all of the security considerations described in RFC 3023 [1] and RFC 5261 [2], plus those listed in Section 5.

Interoperability considerations: N/A

Published specification: RFC XXXX

Applications that use this media type: Applications that manipulate XML documents.

Additional information:

Magic number(s): N/A

File extension(s): XML documents should use ".xml" as the file extension.

Macintosh file type code(s): TEXT

Person & email address to contact for further information: Erik Wilde <erik.wilde@emc.com>

Intended usage: COMMON

Restrictions on usage: none

Author: Erik Wilde <erik.wilde@emc.com>

Change controller: IETF

3. Patch Document Format

The XML patch document format is based on a simple schema that uses a "patch" element as the document element, and allows a arbitrary sequence of "add", "remove", and "replace" elements as the children of the document element. These children follow the semantics defined in [RFC 5261](#), which means that each element is treated as an individual patch operation, and the result of each patch operation is a patched XML document that is the target XML document for the next patch operation.

The following example patch document uses the example from [RFC 5261](#), and simply uses a "patch" element and a new XML namespace. It shows the general structure of an XML patch document, as well as an example for each operation.


```

<p:patch xmlns="urn:ietf:params:xml:ns:xxx"
xmlns:y="urn:ietf:params:xml:ns:yyy"
  xmlns:p="urn:ietf:rfc:XXXX" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
  xsi:schemaLocation="urn:ietf:rfc:XXXX patch-document-00.xsd">
  <p:add sel="doc/elem[@a='foo']">
    <!-- This is a new child -->
    <child id="ert4773">
      <y:node/>
    </child>
  </p:add>
  <p:replace sel="doc/note/text()">Patched doc</p:replace>
  <p:remove sel="*/elem[@a='bar']/y:child" ws="both"/>
  <p:add sel="*/elem[@a='bar']" type="@b">new attr</p:add>
</p:patch>

```

As this example demonstrates, both the document element "patch" and the patch operation elements are in the same XML namespace. This is the result of [RFC 5261](#) only defining types for the patch operation elements, which then can be reused in schemas to define concrete patch elements.

[RFC 5261](#) defines an XML Schema (XSD) for the patch operation types, which is included for reference [Appendix A](#). The normative version of this schema is the one given in [RFC 5261](#). The following schema for the XML Patch media type is based on the types defined in [RFC 5261](#), which are imported as "[rfc5261.xsd](#)" in the following schema. The schema defines a "patch" document element, and then allows an unlimited (and possible empty) sequence of the "add", "remove", and "replace" operation elements, which are directly based on the respective types from the schema defined in [RFC 5261](#).

```

<xs:schema targetNamespace="urn:ietf:rfc:XXXX"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:import schemaLocation="rfc5261.xsd" />
  <xs:element name="patch">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="add" type="add"/>
        <xs:element name="remove" type="remove"/>
        <xs:element name="replace" type="replace"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

4. Patch Examples

Since the semantics of the XML patch operations are defined by [RFC 5261](#), please refer to the numerous examples in that specification for

concrete XML patch document examples. Most importantly, the examples in [RFC 5261](#) can be taken literally as examples for the XML Patch media type, as long as it is assumed that the XML namespace for the operation elements in these examples is the URI "urn:ietf:rfc:XXXX".

5. Security Considerations

...

6. References

6.1. Normative References

- [1] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [2] Urpalainen, J., "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors", [RFC 5261](#), September 2008.
- [3] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [4] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [5] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.

6.2. Informative References

- [6] Dusseault, L. and J. Snell, "PATCH Method for HTTP", [RFC 5789](#), March 2010.
- [7] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.

URIs

- [8] <<https://www.ietf.org/mailman/listinfo/apps-discuss>>
- [9] <<https://github.com/dret/I-D/tree/master/xml-patch>>

Appendix A. XSD from [RFC 5261](#)

For reference, this section contains a copy of the XSD defining the add, replace, and remove types in [RFC 5261](#) [2]. This section is informational only, and the definitive version of the schema is the one listed in [RFC 5261](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE schema [
  <!ENTITY ncname "&i\c*">
  <!ENTITY qname "(&ncname;:)?&ncname;">
  <!ENTITY aname "@&qname;">
  <!ENTITY pos "\[&d+\]">
  <!ENTITY attr "\[&aname;='(.)*'\]|\[&aname;=&quot;(.)*&quot;;\]">
  <!ENTITY valueq "\[(&qname;|\.)=&quot;(.)*&quot;;\]">
  <!ENTITY value "\[(&qname;|\.)='(.)*'\]|\[&valueq;">
  <!ENTITY cond "&attr;|&value;|&pos;">
  <!ENTITY step "(&qname;|\*)(&cond;)*">
  <!ENTITY piq "processing-instruction\((&quot;&ncname;&quot;;)\)">
  <!ENTITY pi "processing-instruction\(('&ncname;')?&piq;">
  <!ENTITY id "id\(('&ncname;')?&piq;|id\((&quot;&ncname;&quot;;)?&piq;">
  <!ENTITY com "comment\(\)">
  <!ENTITY text "text\(\)">
  <!ENTITY nsapa "namespace::&ncname;">
  <!ENTITY cnodes "(&text;(&pos;))|(&com;(&pos;))|((&pi;)(&pos;))">
  <!ENTITY child "&cnodes;|&step;">
  <!ENTITY last "(&child;|&aname;|&nsapa;)">
]>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <xsd:simpleType name="xpath">
    <xsd:restriction base="xsd:string">
      <xsd:pattern
        value="(\/)?((&id;)((/(&step;)*(/&last;))?)|(&step;/)*(&last;))"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="xpath-add">
    <xsd:restriction base="xsd:string">
      <xsd:pattern
        value="(\/)?((&id;)((/(&step;)*(/&child;))?)|
(&step;/)*(&child;))"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="pos">
    <xsd:restriction base="xsd:string">
```

<xsd:enumeration value="before"/>

```
        <xsd:enumeration value="after"/>
        <xsd:enumeration value="prepend"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="type">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="&aname;|&nspace;"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="add">
    <xsd:complexContent mixed="true">
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:any processContents="lax" namespace="##any"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="sel" type="xpath-add"
                use="required"/>
            <xsd:attribute name="pos" type="pos"/>
            <xsd:attribute name="type" type="type"/>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="replace">
    <xsd:complexContent mixed="true">
        <xsd:restriction base="xsd:anyType">
            <xsd:sequence>
                <xsd:any processContents="lax" namespace="##any"
                    minOccurs="0" maxOccurs="1"/>
            </xsd:sequence>
            <xsd:attribute name="sel" type="xpath" use="required"/>
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>

<xsd:simpleType name="ws">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="before"/>
        <xsd:enumeration value="after"/>
        <xsd:enumeration value="both"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="remove">
    <xsd:attribute name="sel" type="xpath" use="required"/>
```



```
        <xsd:attribute name="ws" type="ws"/>
    </xsd:complexType>

</xsd:schema>
```

Authors' Addresses

Erik Wilde
EMC

Email: erik.wilde@emc.com

Mark Nottingham
Akamai

Email: mnot@mnot.net

