                    XML Hypertext Application Language
                         draft-michaud-xml-hal-00

Abstract

   This document proposes a media type for representing resources and
   their relations with hyperlinks.

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 October 12, 2015.

Table of Contents

1. Introduction

There is an emergence of non-HTML HTTP applications ("Web APIs")
which use hyperlinks to direct clients around their resources.

The XML Hypertext Application Language (HAL) is a standard which
establishes conventions for expressing hypermedia controls, such as
links, with XML [W3C.REC-xml-20081126].

HAL is a generic media type with which Web APIs can be developed and
exposed as series of links. Clients of these APIs can select links

by their link relation type and traverse them in order to progress through the application.

HAL's conventions result in a uniform interface for serving and consuming hypermedia, enabling the creation of general-purpose libraries that can be re-used on any API utilizing HAL.

The primary design goals of HAL are generality and simplicity. HAL can be applied to many different domains, and imposes the minimal amount of structure necessary to cover the key requirements of a hypermedia Web API.

2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. HAL Documents

A HAL Document uses the format described in [W3C.REC-xml-20081126] and has the media type "application/hal+xml".

Its root MUST be a resource element.

For example:

```
GET /orders/523 HTTP/1.1
Host: example.org
Accept: application/hal+xml

HTTP/1.1 200 OK
Content-Type: application/hal+xml

<resource rel="self" href="/orders/523">
    <link rel="warehouse" href="/warehouse/56"/>
    <link rel="invoice" href="/invoices/873"/>
    <currency>USD</currency>
    <status>shipped</status>
    <total>10.20</total>
</resource>
```

Here, we have a HAL document representing an order resource with the URI "/orders/523". It has "warehouse" and "invoice" links, and its own state in the form of "currency", "status", and "total" elements.

[4](). Resource

The resource element represents a Resource and SHOULD also host the set of attributes that describe a Link (as defined by [[RFC5988]()]). See [Section 5](). and [Section 8.1](). for details.

It has two reserved elements:

(1) "link": a link to another resource.

(2) "resource": an embedded resource.

Everything else MUST be valid XML and represents the current state of the Resource.

[4.1](). Reserved Elements

[4.1.1](). link

The reserved "link" element is OPTIONAL.

This element MAY be used to host the set of attributes that describe a Link (as defined by [[RFC5988]()]). See [Section 5](). for details.

[4.1.2](). resource

The reserved "resource" element is OPTIONAL

If present, this element MUST observe the same constraints defined for the root resource but MUST additionally host a Link so that embedded resources can more easily be identified when processed.

Embedded resources MAY be a full, partial, or inconsistent version of the representation served from the target URI.

[5](). Link

A Link is defined as a set of attributes that represent a hyperlink from the containing resource to a URI. The attributes are as follows:

[5.1](). rel

The "rel" attribute is REQUIRED.

It is an attribute whose relation values are link relation types (as defined by [[RFC5988]()]).

5.1. href

The "href" attribute is REQUIRED.

Its value is either a URI [RFC3986] or a URI Template [RFC6570].

If the value is a URI Template then the Link SHOULD have a
"templated" attribute whose value is true.

5.2. templated

The "templated" attribute is OPTIONAL.

Its value is boolean (as defined by [W3C.REC-xmlschema-2-20041028])
and SHOULD be true when the Link's "href" attribute is a URI
Template.

Its value SHOULD be considered false if the attribute is absent.

5.3. type

The "type" attribute is OPTIONAL.

Its value is a string used as a hint to indicate the media type
expected when dereferencing the target resource.

5.4. deprecation

The "deprecation" attribute is OPTIONAL.

Its presence indicates that the link is to be deprecated (i.e.
removed) at a future date. Its value is a URL that SHOULD provide
further information about the deprecation.

A client SHOULD provide some notification (e.g. by logging a warning
message) whenever it traverses over a link that has this attribute.
The notification SHOULD include the deprecation attribute's value so
that a client maintainer can easily find information about the
deprecation.

5.5. name

The "name" attribute is OPTIONAL.

Its value MAY be used as a secondary key for selecting Links which
share the same relation type.

5.6. **profile**

   The "profile" attribute is OPTIONAL.

   Its value is a string which is a URI that hints about the profile (as
   defined by [RFC6906]) of the target resource.

5.7. **title**

   The "title" attribute is OPTIONAL.

   Its value is a string and is intended for labeling the link with a
   human-readable identifier (as defined by [RFC5988]).

5.8. **hreflang**

   The "hreflang" attribute is OPTIONAL.

   Its value is a string and is intended for indicating the language of
   the target resource (as defined by [RFC5988]).

6. **Example**

   The following is an example representing a list of orders

```
GET /orders HTTP/1.1
Host: example.org
Accept: application/hal+xml

HTTP/1.1 200 OK
Content-Type: application/hal+xml

<resource rel="self" href="/orders">
    <link rel="next" href="/orders?page=2"/>
    <link rel="find" href="/orders{?id}" templated="true"/>
    <resource rel="order" href="/orders/123">
        <link rel="basket" href="/baskets/98712"/>
        <link rel="customer" href="/customers/7809"/>
        <total>30.00</total>
        <currency>USD</currency>
        <status>shipped</status>
    </resource>
    <resource rel="order" href="/orders/124">
        <link rel="basket" href="/baskets/97213"/>
        <link rel="customer" href="/customers/12369"/>
        <total>20.00</total>
        <currency>USD</currency>
        <status>processing</status>
    </resource>
    <currentlyProcessing>14</currentlyProcessing>
    <shippedToday>20</shippedToday>
</resource>
```

   Here, the order list document provides a "next" link directing to the
   next page, and a "find" link containing a URI Template which can be
   expanded with an 'id' variable to go directly to a specific order.

   It also has two embedded resources, "order". Each of these has its
   own links to the associated "basket" and "customer" resources, and
   properties showing their "total", "currency" and "status".

   Additionally, the order list resource has its own properties
   "currentlyProcessing" and "shippedToday".

7. Media Type Parameters

7.1. profile

   The media type identifier application/hal+xml MAY also include an
   additional "profile" parameter (as defined by [RFC6906])

   HAL documents that are served with the "profile" parameter still
   SHOULD include a "profile" link belonging to the root resource.

8. Recommendations

8.1. Self Link

   Each Resource SHOULD have a 'self' link that corresponds with the
   IANA registered 'self' relation (as defined by [RFC5988]) and whose
   target is the resource's URI.

8.2. Link relations

   Custom link relation types (Extension Relation Types in [RFC5988])
   SHOULD be URIs that when dereferenced in a web browser provide
   relevant documentation, in the form of an HTML page, about the
   meaning and/or behavior of the target Resource. This will improve
   the discoverability of the API.

   The CURIE Syntax [W3C.NOTE-curie-20101216] MAY be used for brevity
   for these URIs. CURIEs are established within a HAL document via XML
   namespace on the root Resource.

```
<resource rel="self" href="/orders" xmlns:acme="http://a.com/rels/">
  <link rel="acme:widgets" href="/widgets"/>
</resource>
```

   The above demonstrates the relation "http://a.com/rels/widgets" being
   abbreviated to "acme:widgets" via CURIE syntax.

8.3. Hypertext Cache Pattern

   The "hypertext cache pattern" allows servers to use embedded
   resources to dynamically reduce the number of requests a client
   makes, improving the efficiency and performance of the application.

   Clients MAY be automated for this purpose so that, for any given link
   relation, they will read from an embedded resource (if present) in
   preference to traversing a link.

   To activate this client behavior for a given link, servers SHOULD add

an embedded resource into the representation with the same relation.

Servers SHOULD NOT entirely "swap out" a link for an embedded
resource (or vice versa) because client support for this technique is
OPTIONAL.

The following examples shows the hypertext cache pattern applied to
an "author" link:

Before:

```
<resource rel="self" href="/books/the-way-of-zen">
    <link rel="author" href="/people/alan-watts"/>
</resource>
```

After:

```
<resource rel="self" href="/books/the-way-of-zen">
    <link rel="author" href="/people/alan-watts"/>
    <resource rel="author" href="/people/alan-watts">
        <name>Alan Watts</name>
        <born>January 6, 1915</born>
        <died>November 16, 1973</died>
    <resource>
</resource>
```

8.4. Namespace and prefix

The default namespace of "http://stateless.co/hal/ns" SHOULD be used
to allow for mixed, and potentially conflicting, xml vocabularies to
coexist in the same xml instance (e.g. embedding an
application/hal+xml payload in another xml).

If a namespace prefix is needed, "hal" MAY be used by default.

9. Security Considerations

TBD

10. IANA Considerations

TBD

11. Normative References

[RFC6906] Wilde, E., "The 'profile' Link Relation Type", RFC 6906,
           March 2013.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifier (URI): Generic Syntax", STD 66, RFC
            3986, January 2005.

[W3C.REC-xml-20081126]

            Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E.,
            Yergeau, F., "Extensible Markup Language (XML) 1.0 (Fifth
            Edition)", World Wide Web Consortium REC REC-xml-20081126,
            November 2008, <http://www.w3.org/TR/2008/REC-xml-
            20081126/>.

[RFC5988]   Nottingham, M., "Web Linking", RFC 5988, October 2010.

[RFC6570]   Gregorio, J., Fielding, R., Hadley, M., Nottingham, M.,
            and D. Orchard, "URI Template", RFC 6570, March 2012.

[W3C.NOTE-curie-20101216]

            McCarron, S. and M. Birbeck, "CURIE Syntax 1.0", World
            Wide Web Consortium NOTE NOTE-curie-20101216, December
            2010, <http://www.w3.org/TR/2010/NOTE-curie-20101216>.

[W3C.REC-xmlschema-2-20041028]

            Biron, P. V., Malhotra, A., "XML Schema Part 2: Datatypes
            Second Edition", World Wide Web Consortium REC REC-
            xmlschema-2-20041028, October 2004,
            <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

Appendix A. Acknowledgements

Thanks to Mike Kelly for making this specification possible through
the creation of HAL and the hal+json specification.

I would also like to thank folks on the Hal Discuss Google group for
providing feedback for the draft.

The author takes all responsibility for errors and omissions.

Appendix B. **Frequently Asked Questions**

B.1. **How should a client know the meaning/structure/semantics/type of a** resource?

There are two main approaches to solving this problem. Both involve exposing additional documentation describing the resource which may be human and/or machine readable (e.g. an HTML page, an XML Schema, an ALPS profile, etc.). The difference between the two approaches is in where that URI is shared with the client, which is either:

(1) The relation documentation associated to a Link relation type.

(2) A 'profile' (via a Link's profile attribute or via a Link with a 'profile' relation type).

B.2. **Where can I find libraries for working with HAL?**

A list of libraries is maintained here:
http://stateless.co/hal_specification.html

B.3. **Why does HAL have no forms?**

Omitting forms from HAL was an intentional design decision that was made to keep it focused on linking for APIs. HAL is therefore a good candidate for use as a base media type on which to build more complex capabilities.

Author's Address

Jeff Michaud

Email: cometaj2@comcast.net
Twitter: @cometaj2