

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 3, 2013

E. Wilde
EMC
April 1, 2013

**Home Documents for HTTP Services: XML Syntax
draft-wilde-home-xml-00**

Abstract

The current draft for HTTP Home Documents provides a JSON syntax only. This draft provides an XML syntax for the same underlying data model, so that the concept of HTTP Home Documents can be consistently exposed in both JSON- and XML-based HTTP services.

Note to Readers

Please discuss this draft on the apps-discuss mailing list [5].

Online access to all versions and files is available on github [6].

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 3, 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.](#) XML Example [3](#)
- [3.](#) XML Schema [4](#)
- [4.](#) IANA Considerations [6](#)
 - [4.1.](#) Media Type application/home+xml [6](#)
- [5.](#) Open Issues [7](#)
- [6.](#) Normative References [8](#)
- Author's Address [8](#)

1. Introduction

An Internet Draft currently under development [[1](#)] proposes the concept of "Home Documents for HTTP APIs" and described them as follows:

"This document proposes a 'home document' format for non-browser HTTP clients. [...] The goal of home documents is to serve as a starting point for hypermedia APIs, where clients need to have an entry point, and then can use the API by following links. Home documents thus serve the same purpose as home pages on web sites: They are stable entry points that provide starting points for clients with some knowledge of the services linked from them."

While this general concept of a home document is independent of the representation format, the current draft only defines a JSON syntax. In order to make this concept available across representations, this draft defines an XML syntax for the concepts defined in [[1](#)].

At this point it is undecided whether both drafts will be merged eventually, or whether they will both be published as separate documents. Regardless of the final publication setup, it should be noted that this draft is only defining the XML syntax, whereas all the concepts represented in this syntax are defined by [[1](#)].

2. XML Example

The following Home Document in XML syntax uses the same data as the Home Document shown in Section 3 of [[1](#)]:

```
<resources xmlns="urn:ietf:rfc:XXXX">
  <resource rel="http://example.org/rel/widgets">
    <link href="/widgets/" />
  </resource>
  <resource rel="http://example.org/rel/widget">
    <template href-template="/widgets/{widget_id}">
      <var name="widget_id" URI="http://example.org/param/widget" />
    </template>
    <hints>
      <allow>
        <i>GET</i>
        <i>PUT</i>
        <i>DELETE</i>
        <i>PATCH</i>
      </allow>
      <representations>
        <i>application/json</i>
      </representations>
      <accept-patch>
        <i>application/patch+json</i>
      </accept-patch>
      <accept-post>
        <i>application/xml</i>
      </accept-post>
      <accept-ranges>
        <i>bytes</i>
      </accept-ranges>
    </hints>
  </resource>
</resources>
```

The mapping between JSON arrays and XML uses "item" elements `<i/>`, where each of those elements represents one array item. For properties that have a single values (i.e., they are not defined as an array of values), this value is directly contained as content in the corresponding element.

Currently, the draft does not specify an extension model (how to represents hints that are not specified in the draft itself), and therefore the extension model for XML is currently undefined as well. The XML syntax will be updated to reflect the extension model once it has been specified for the JSON syntax.

3. XML Schema

The following XML Schema is describing the XML shown in [Section 2](#). Since there currently is no extension model, the XML Schema does

currently not contain any extension points.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ietf:rfc:XXXX"
elementFormDefault="qualified" xmlns:home="urn:ietf:rfc:XXXX">
  <xs:element name="resources">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded" minOccurs="0">
        <xs:element name="resource">
          <xs:complexType>
            <xs:sequence>
              <xs:choice>
                <xs:element name="link">
                  <xs:complexType>
                    <xs:attribute name="href" type="xs:anyURI" use="required"/>
                  </xs:complexType>
                </xs:element>
                <xs:element name="template">
                  <xs:complexType>
                    <xs:sequence maxOccurs="unbounded" minOccurs="0">
                      <xs:element name="var">
                        <xs:complexType>
                          <xs:attribute name="name" use="required"/>
                          <xs:attribute name="URI" type="xs:anyURI" use="required"/>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                    <xs:attribute name="href-template" use="required"/>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:sequence>
            <xs:element minOccurs="0" name="hints">
              <xs:complexType>
                <xs:choice maxOccurs="unbounded" minOccurs="0">
                  <xs:element name="allow" type="home:arrayType"/>
                  <xs:element name="representations" type="home:arrayType"/>
                  <xs:element name="accept-patch" type="home:arrayType"/>
                  <xs:element name="accept-post" type="home:arrayType"/>
                  <xs:element name="accept-put" type="home:arrayType"/>
                  <xs:element name="accept-ranges" type="home:arrayType"/>
                  <xs:element name="prefer" type="home:arrayType"/>
                  <xs:element name="docs" type="xs:anyURI"/>
                  <xs:element name="precondition-req" type="home:arrayType"/>
                  <xs:element name="auth-req">
                    <xs:complexType>
                      <xs:sequence maxOccurs="unbounded" minOccurs="0">
                        <xs:element name="scheme">
                          <xs:complexType>
                            <xs:sequence maxOccurs="unbounded" minOccurs="0">
```

```
        <xs:element name="realm"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:token"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="status">
  <xs:simpleType>
    <xs:restriction base="xs:token">
      <xs:enumeration value="deprecated"/>
      <xs:enumeration value="gone"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
  <xs:attribute name="rel" type="xs:anyURI" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="arrayType">
  <xs:sequence maxOccurs="unbounded" minOccurs="0">
    <xs:element name="i"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

4. IANA Considerations

This specification registers a media type for the XML syntax of Home Documents (as defined in [\[1\]](#)).

4.1. Media Type application/home+xml

The Internet media type [\[2\]](#) for a Home Document in XML syntax is application/home+xml.

Type name: application

Subtype name: home+xml

Required parameters: none

Optional parameters: Same as charset parameter for the media type "application/xml" as specified in [RFC 3023](#) [3].

Encoding considerations: Same as encoding considerations of media type "application/xml" as specified in [RFC 3023](#) [3].

Security considerations: This media type has all of the security considerations described in [RFC 3023](#) [3] and [1].

Interoperability considerations: N/A

Published specification: RFC XXXX

Applications that use this media type: Applications that publish Home Documents for HTTP services using XML syntax.

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

5. Open Issues

- o What is the extension model for the XML syntax? Should processing of other namespaces be defined as "should ignore", so that same-namespace extensions are encouraged?
- o Should the XML syntax provide support for embedded human-readable documentation? This would probably not be supported in the JSON syntax, but could be marked as strictly optional and XML-specific.

6. Normative References

- [1] Nottingham, M., "Home Documents for HTTP APIs", [draft-nottingham-json-home-02](#) (work in progress), September 2012.
- [2] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), January 2013.
- [3] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [4] Sperberg-McQueen, C., Yergeau, F., Paoli, J., Maler, E., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126>>.
- [5] <<https://www.ietf.org/mailman/listinfo/apps-discuss>>
- [6] <<https://github.com/dret/I-D/tree/master/template-desc>>

Author's Address

Erik Wilde
EMC
6801 Koll Center Parkway
Pleasanton, CA 94566
U.S.A.

Phone: +1-925-6006244
Email: erik.wilde@emc.com
URI: <http://dret.net/netdret/>