The 'publish' Link Relation Type
draft-hamnaberg-publish-link-relation-01

Abstract

This memo defines a 'publish' link relation and provides a number of examples.

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 21, 2014.

Copyright Notice

Copyright (c) 2014 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.
Internet-Draft      The 'publish' Link Relation Type        January 2014

1. Introduction

This specification outlines the "publish" link relation and what it
means to publish something. The specification will register the
relation according to [RFC5988].

No assumptions will be made about which media type the target IRI
will accept, but some examples and guidelines will be given.

1.1. Notational Conventions

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].

1.2. Editorial Note (to be removed by RFC Editor)

Please discuss this draft on the apps-discuss@ietf.org mailing list.

2. Publishing

What does it mean to publish something? In effect it can be
interpreted as making something available for general consumption.

A newspaper can publish an article, then it can be read by
‘everyone’.

A publisher can publish a book, so it can be read.

A video can be submitted to youtube so it can be watched.
2.1. The ‘publish’ link relation

The "publish" link relation allows a resource to be published using the target IRI.

Clients SHOULD use an appropriate write method of the target IRI protocol uniform interface.

Servers MAY ignore the request to publish something if the resource URI is already published.

Examples can be found in Appendix A

3. IANA Considerations

IANA is asked to register the link relation "publish", as per [RFC5988]

Relation Name:
publish

Description:
Allows resources to be published using the target IRI.

Reference:
[ this document ]

4. Security Considerations

TBD

5. Acknowledgements

Thanks to Jan Algermissen, Peter Rushforth and Darrel Miller for their valuable feedback and comments.

6. References

6.1. Normative References


6.2. Informative References


Appendix A. Examples

Let’s say you have an atom feed like the one below
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
     xmlns:app="http://www.w3.org/2007/app">
   <title>Content feed</title>
   <id>tag:example.org,2003:3</id>
   <link rel="self" type="application/atom+xml"
        href="http://example.org/feed"/>
   <link rel="publish" href="http://example.org/publish"/>
   <entry>
      <title>Item 1</title>
      <link rel="edit" href="http://example.org/item/1"/>
      <id>urn:id:1</id>
      <app:control>
         <app:draft>yes</app:draft>
      </app:control>
      <updated>2012-05-04T12:00:29Z</updated>
      <author>
         <name>Erlend Hamnaberg</name>
      </author>
      <content type="text">
         Some Content here.
      </content>
   </entry>
   <entry>
      <title>Item 2</title>
      <link rel="edit" href="http://example.org/item/2"/>
      <id>urn:id:1</id>
      <app:control>
         <app:draft>yes</app:draft>
      </app:control>
      <updated>2012-05-04T12:29:29Z</updated>
      <author>
         <name>Erlend Hamnaberg</name>
      </author>
      <content type="text">
         Some Content here.
      </content>
   </entry>
</feed>

Using Atompub ([RFC5023]), a Client would have to GET each entry using it's "edit" relation and PUT each back with

<app:draft>no</app:draft>
being the only change in both entries.

This is highly inefficient. So let us find a better way.

A.1. Publishing multiple resources

To be able to publish many resources at once we need a representation which allows this.

Appendix A in [RFC2169] defines the "text/uri-list" media type.

Extra linefeeds are for display purposes only.

POST /publish
Host: example.org
Content-Type: text/uri-list

http://example.org/item/1
http://example.org/item/2

The problem with this approach is that is is a separate resource, and will not invalidate the caches of the feed. This problem is also apparent in the single item approach.

One might mediate this by making the feed resource also accept "text/uri-list" for publishing.

We could then change the request to this:

POST /feed
Host: example.org
Content-Type: text/uri-list

http://example.org/item/1
http://example.org/item/2

Author’s Address

Erlend Hamnaberg

Email: erlend@hamnaberg.net
URI: http://www.hamnis.org/