Individual Submission J. Snell
Internet-Draft
Intended status: Informational August 22, 2013
Expires: February 23, 2014

HTTP Link and Unlink Methods
draft-snell-link-method-03

Abstract

This specification defines the semantics of the Link and Unlink HTTP methods.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This specification updates the HTTP LINK and UNLINK methods originally defined in [RFC2068].

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

2. LINK

The LINK method is used to establish one or more relationships between the resource identified by the effective request URI and one or more other resources. Metadata contained within Link header fields [RFC5988] provide information about which other resources are being connected to the target and the type of relationship being established. A payload within a LINK request has no defined semantics.

LINK requests are idempotent. For any pair of resources, exactly one relationship of any given type can exist. However, multiple relationships of different types can be established between the same pair of resources.

LINK requests are not safe, however. Establishing a relationship causes an inherent change to the state of the target resource.

Responses to LINK requests are not cacheable. If a LINK request passes through a cache that has one or more stored responses for the effective request URI, those stored responses will be invalidated (see Section 6 of [I-D.ietf-httpbis-p6-cache]).

A single LINK request can contain multiple Link header fields, each of which establishes a separate, independent relationship with the

target resource. In such cases, the server MUST either create all of the relationships or not create any of them.

3. UNLINK

The UNLINK method is used to remove one or more relationships between the resource identified by the effective request URI and other resources. Metadata contained within Link header fields [RFC5988] provide the information about the relationships that are to be removed. A payload within an UNLINK request has no defined semantics.

UNLINK request messages are idempotent but are not safe. Removing a relationship causes an inherent change to the state of the target resource.

Responses to UNLINK requests are not cacheable. If an UNLINK request passes through a cache that has one or more stored responses for the effective request URI, those stored responses will be invalidated (see Section 6 of [I-D.ietf-httpbis-p6-cache]).

A single UNLINK request message can contain multiple Link header fields, each of which identifies a separate relationship to remove. In such cases, the server MUST either remove the entire set of relationships atomically or not remove any of them.

4. Relationship to other HTTP Methods and Discoverability of Links

The use of the LINK and UNLINK request methods to manage relationships between resources has no direct bearing on the use or appearance of Link header fields within any other HTTP request or response message involving the same effective request URI. Nor do the methods have any direct normative impact on the use of link-like structures within the resource representations returned by a server for any particular resource.

Whether and how to represent relationships managed using LINK and UNLINK is left solely at the discretion of the server implementation.

This specification does not define a means of discovering or enumerating the relationships that have been established using the LINK request method.

5. Example

There exists a broad range of possible use cases for the LINK and UNLINK methods. The examples that follow illustrate a subset of those cases.

Example 1: Creating two separate links between an image and the profiles of two people associated with the image:

```
LINK /images/my_dog.jpg HTTP/1.1
Host: example.org
Link: <http://example.com/profiles/joe>; rel="tag"
Link: <http://example.com/profiles/sally>; rel="tag"
```

Example 2: Removing an existing Link relationship between two resources:

```
UNLINK /images/my_dog.jpg HTTP/1.1
Host: example.org
Link: <http://example.com/profiles/sally>; rel="tag"
```

Example 3: Establish a "pingback" or "trackback" style link to a blog entry about an article

```
LINK /articles/an_interesting_article HTTP/1.1
Host: example.org
Link: <http://example.com/my_blog_post>; rel="mention"
```

Example 4: Establish a link between two semantically related resources:

```
LINK /some-resource HTTP/1.1
Host: example.org
Link: <http://example.com/schemas/my_schema>; rel="describedBy"
```

Example 5: Add an existing resource to a collection:

```
LINK /some-collection-resource HTTP/1.1
Host: example.org
Link: <http://example.com/my-member-resource>; rel="item"
```

Example 6: Link one resource to another that monitors its current state (e.g. pub/sub)

```
   LINK /my-resource HTTP/1.1
   Host: example.org
   Link: <http://example.com/my-monitor>; rel="monitor"
```

6. Security Considerations

The LINK and UNLINK methods are subject to the same general security considerations as all HTTP methods as described in [I-D.ietf-httpbis-p2-semantics].

Because the LINK and UNLINK methods cause changes to a resource's state, the server is responsible for determining the client's authorization to make such changes.

7. References

7.1. Normative References

[I-D.ietf-httpbis-p2-semantics]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", draft-ietf-httpbis-p2-semantics-23 (work in progress), July 2013.

[I-D.ietf-httpbis-p6-cache]
Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", draft-ietf-httpbis-p6-cache-23 (work in progress), July 2013.

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

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

7.2. Informational References

[RFC2068] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

Author's Address

James M Snell

Email: jasnell@gmail.com