Individual Submission
Internet-Draft
Intended status: Standards Track
Expires: March 24, 2016

J. Snell
September 21, 2015

HTTP Link and Unlink Methods
draft-snell-link-method-12

Abstract

This specification defines the semantics of the LINK and UNLINK HTTP methods.

Status of This Memo

This Internet-Draft is submitted to IETF 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 March 24, 2016.

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 Relationships

The LINK and UNLINK methods are used to manage relationships between resources. Those relationships are defined using the Link model established in Section 3 of [RFC5988]. For every individual link, the context IRI, link relation type, target IRI, and optional collection of target attributes MUST be considered; that is, for any effective request URI, there can exist at most one Link relationship between any context and target IRI pairing with any given combination of link relation type and target attributes.

Within LINK and UNLINK requests, a [RFC5988] Link header field is used to describe a Link relationship to be managed. Any single LINK or UNLINK request MAY contain multiple Link header fields, each of which describes a separate relationship between a context IRI and target IRI. When a LINK request contains multiple Link header fields, the server MUST create all of the specified relationships or not create any of them. Likewise, when an UNLINK request contains multiple Link header fields, the server MUST either remove all the specified relationship or not remove any of them.

The target and context IRIs of a Link relationship are determined following the requirements specified in Sections 5.1 and 5.2 of [RFC5988].

When determining whether or not a relationship already exists between a context IRI and target IRI, implementations will need to compare the given IRIs with other, previously established relationships. To do so, the implementation MUST first resolve the IRIs as required by [RFC5988] and then compare on a case-sensitive, character-by-character basis. For instance, the IRIs "http://example.org/foo" and "http://example.org/Foo" MUST NOT be considered to be equivalent.

3. 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 the relationships being established. A payload within a LINK request has no defined semantics.

LINK requests are idempotent but are not safe. Establishing a relationship causes an inherent change to the state of the target resource. Note, however, that acceptance of a LINK request is inherently noncomittal on the part of the server. That is, the server MAY signal the acceptance of a LINK request without taking any further action on it.

Any successful response (using a 2xx status code) to a LINK request indicates that all of the Link relationships described in the request have been established. No specific 2xx status code is required.

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 [RFC7234]).

The semantics of the LINK method change to a "conditional LINK" if the request message includes an If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, or If-Range header field ([RFC7232]). A conditional LINK requests that the relationship be established only under the circumstances described by the conditional header field(s).

4. 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. Note, however, that acceptance of an UNLINK request is inherently noncomittal on the part of the server. That is, the server MAY signal the acceptance of an UNLINK request without taking any further action on it.

Any successful response (using a 2xx status code) to an UNLINK request indicates that all of the Link relationships described in the request have been removed. No specific 2xx status code is required.

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 [RFC7234]).

The semantics of the UNLINK method change to a "conditional UNLINK" if the request message includes an If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, or If-Range header field ([RFC7232]). A conditional UNLINK requests that the relationship be established only under the circumstances described by the conditional header field(s).

5. 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 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.

6. 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"
```

Possible response:

```
HTTP/1.1 202 Accepted
```

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"
```

Possible response:

```
HTTP/1.1 204 No Content
```

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"
```

Example 7: Using the Link anchor attribute to change the context IRI (in this example, a link relationship is established between the IRIs "acct:joe@example.org" and "acct:sally@example.org")

```
LINK /my-resource HTTP/1.1
Host: example.org
Link: <acct:joe@example.org>; rel="follow";
  anchor="acct:sally@example.org"
```

Example 8: Using fragment identifiers to establish links with embedded resources. In this example, a link relationship is established between "http://example.org/alice#me" and "http://example.org/bob#me".

```
LINK /alice HTTP/1.1
Host: example.org
Link: <http://example.org/bob#me>; rel="knows"; anchor="#me"
```

7. Security Considerations

The LINK and UNLINK methods are subject to the same general security considerations as all HTTP methods as described in [RFC7231].

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

8. IANA Considerations

IANA is requested to add the LINK and UNLINK methods to the permanent registry at <http://www.iana.org/assignments/http-methods> (see Section 8.1 of [RFC7231]).

Method Name	Safe	Idempotent	Specification
LINK	No	Yes	Section 3
UNLINK	No	Yes	Section 4

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <http://www.rfc-editor.org/info/rfc5988>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <http://www.rfc-editor.org/info/rfc7232>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

9.2. Informational References

[RFC2068] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, DOI 10.17487/RFC2068, January 1997, <http://www.rfc-editor.org/info/rfc2068>.

Author's Address

James M Snell

Email: jasnell@gmail.com