

HTTP Link and Unlink Methods
draft-snell-link-method-00

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 April 11, 2013.

Copyright Notice

Copyright (c) 2012 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.	LINK	3
3.	UNLINK	4
4.	Example	5
5.	Security Considerations	6
6.	Normative References	6
	Author's Address	7

1. Introduction

This specification updates the HTTP LINK and UNLINK methods originally defined in [[RFC2068](#)]. These were originally defined as "additional request methods" that were later dropped entirely from follow-on iterations of the HTTP specification due to previous lack of interest or use.

TODO: Fill in explanation as to why this is needed.

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 Link relationships between the existing resource identified by the effective request URI and other resources. Metadata contained within Link header fields [[RFC5988](#)] provide the information about which other resources are being linked to the target resource and the type of link being established. A payload within a LINK request message has no defined semantics.

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 ([[I-D.ietf-httpbis-p4-conditional](#)]). A conditional LINK requests that the Link be established only under the circumstances described by the conditional header field(s).

LINK request messages are idempotent. For any pair of resources, only a single Link of any given relation type can exist. However, multiple links of different relation types can be established between those resources.

LINK request messages are not safe, however, in that establishing a Link causes an inherent change to the state of the target resource.

The response to a LINK request is cacheable and MAY be used to satisfy subsequent LINK requests (see [[I-D.ietf-httpbis-p6-cache](#)]). However, caching such responses is not likely to provide any significant benefit or be supported by existing infrastructure.

A single LINK request message can contain multiple Link header fields, each of which establishes a separate Link relationship with the target resource. In such cases, the server MUST accept the

entire set of Links atomically. If any of the specified Link relationships cannot be created, the server MUST NOT create any of the Links.

A successful response SHOULD be 200 (OK) if the response includes a representation describing the status, 201 (Created) if the action results in the creation of a new resource that represents the newly established link, 202 (Accepted) if the action has not yet been enacted, or 204 (No Content) if the action has been enacted but the response does not include a representation.

3. UNLINK

The UNLINK method is used to remove one or more Link relationships of between the existing resource identified by the effective request URI and other resources. Metadata contained within Link header fields [[RFC5988](#)] provide the information about the resources to which links of a specific type are to be removed. A payload within an UNLINK request message has no defined semantics.

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 ([[I-D.ietf-httpbis-p4-conditional](#)]). A conditional UNLINK requests that the Link be removed only under the circumstances described by the conditional header field(s).

UNLINK request messages are idempotent.

UNLINK request messages are not safe, however, in that removing a Link causes an inherent change to the state of the target resource.

The response to an UNLINK request is cacheable and MAY be used to satisfy subsequent UNLINK requests (see [[I-D.ietf-httpbis-p6-cache](#)]). However, caching such responses is not likely to provide any significant benefit or be supported by existing infrastructure.

A single UNLINK request message can contain multiple Link header fields, each of which identifies a separate Link relationship to remove. In such cases, the server MUST remove the entire set of Links atomically. If any of the specified Link relationships cannot be removed, the server MUST NOT remove any of the Links.

A successful response SHOULD be 200 (OK) if the response includes a representation describing the status, 202 (Accepted) if the action has not yet been enacted, or 204 (No Content) if the action has been enacted but the response does not include a representation.

The UNLINK method MAY be overridden by human intervention (or other means) on the origin server. The client cannot be guaranteed that the operation has been carried out, even if the status code returned from the origin server indicates that the action has been completed successfully. However, the server SHOULD NOT indicate success unless, at the time the response is given, it intends to remove the specified Links.

4. 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="about"
```

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 it's current state (e.g. pub/sub)

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

5. Security Considerations

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

Implementers need to be aware of the possible ways the LINK method can be abused as a means of propagating inappropriate links to external resources. For instance, the unregulated acceptance of LINK requests can be used as a vector for spam or malware distribution.

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.

6. Normative References

[I-D.ietf-httpbis-p2-semantic]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content",
[draft-ietf-httpbis-p2-semantic-21](#) (work in progress),
October 2012.

[I-D.ietf-httpbis-p4-conditional]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests",
[draft-ietf-httpbis-p4-conditional-21](#) (work in progress),
October 2012.

[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-21](#) (work in progress),

October 2012.

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

Author's Address

James M Snell

Email: jasnell@gmail.com

