Abstract

This specification defines two media types and a link relation type for sets of links. The media types can be used to represent links in a standalone fashion, in one case in the native format used in the HTTP Link header, and in the other case in a JSON-based format. The link relation can be used to reference these kinds of standalone link sets, so that a resource can indicate that additional links putting it into context are available somewhere else. One typical scenario is when the number of links to put in an HTTP Link header field becomes too big, and thus these links should be provided by a distinct resource.

Note to Readers

Please discuss this draft on the ART mailing list (<https://www.ietf.org/mailman/listinfo/art>).

Online access to all versions and files is available on GitHub (<https://github.com/dret/I-D/tree/master/linkset>).

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 https://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 10, 2018.
1. Introduction

Resources on the Web often convey typed Web Links [I-D.nottingham-rfc5988bis] as a part of resource representations, for example, using the <link> element for HTML representations, or the "Link" header field in HTTP response headers for representations of any media type. In some cases, however, providing links by value is impractical or impossible. In these cases, an approach to provide links by reference (instead of by value) can solve the problem.

To that end, this specification defines two document formats and associated media types to represent a set of links. It also defines the "linkset" relation type that a resource can use to support discovery of another resource that conveys links in which the former resource participates.

2. Terminology

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 RFC 2119 [RFC2119].

This specification uses the terms "link context" and "link target" as defined in [I-D.nottingham-rfc5988bis]. (These terms respectively correspond with "Context IRI" and "Target IRI" as used in RFC 5988 [RFC5988]). Although defined as IRIs, in common scenarios they are also URIs.

Additionally, this specification uses the following terms for types of resources involved in providing links by reference:

- A "link set resource" is a resource that conveys a set of links. Section 4 defines two representations for a set of links, based on the abstract link model defined in [I-D.nottingham-rfc5988bis].

- An "origin resource" is a resource that participates in one or more links provided by a link set resource. An origin resource can support discovery of an associated link set resource by using the relation type defined in Section 5. As such, from the perspective of the origin resource, the links conveyed by the link set resource are provided by reference.

3. Scenarios

The following sections outline some scenarios in which it is useful to have the ability to separate resources from the links that pertain to them.
These are all scenarios in which providing links by reference is advantageous or necessary. It is important to keep in mind that, when providing some links by reference, some/other links can still be provided by value.

3.1. Third-Party Links

In some cases, it is useful that links pertaining to an origin resource are provided by a server other than the one that hosts the origin resource. For example, this allows:

- Providing links in which the origin resource is involved not just as link context but also as link target.
- Providing links pertaining to the origin resource that the server hosting that resource is not aware of.
- External management of links pertaining to the origin resource in a special-purpose link management service.

In such cases, a third-party link set resource provides links pertaining to the origin resource. This link set resource may be managed by the same custodian as the origin resource, or by a third party.

In order for the server hosting the origin resource to provide an up-to-date and complete set of links in which the origin resource participates, the server would need to obtain the links from the link set resource, and embed them in the origin resource’s representations prior to responding to a client. Doing so would increase latency and load, which may be unnecessary if a client is not intent on consuming these links. Providing links by reference, instead of by value, removes the server-to-server communication and resulting overhead required to obtain links. Instead, the consumer of the origin resource can decide if they need the additional links as context for the resource.

3.2. Challenges Writing to HTTP Link Header Field

In some cases, it is not straightforward to write links to the HTTP Link header field from an application. This can, for example, be the case because not all required link information is available to the application or because the application does not have the capability to directly write HTTP headers. In such cases, providing links by reference can be a solution because a link to a link set resource associated with an origin resource can typically be added by means of a web server rewrite rule that leverages the resource’s URI.
3.3. Large Number of Links

When conveying links in the HTTP "Link" header, it is possible for the size of the HTTP response header to become unpredictable. This can be the case when links are determined dynamically dependent on a range of contextual factors. It is possible to statically configure a web server to correctly handle large HTTP response headers by specifying an upper boundary for their size. But when the number of links is unpredictable, estimating a reliable upper boundary is challenging.

HTTP [RFC7231] defines error codes related to excess communication by the user agent ("413 Request Entity Too Large" and "414 Request-URI Too Long"), but no specific error codes are defined to indicate that response header content exceeds the upper boundary that can be handled by the server, and thus it has been truncated. As a result, applications take counter measures aimed at controlling the size of the HTTP "Link" header, for example by limiting the links they provide to those with select relation types, thereby limiting the value of the HTTP "Link" header to clients. Providing links by reference, instead of by value, overcomes challenges related to the unpredictable nature of the extent of HTTP "Link" headers.

In more extreme scenarios it is conceivable that the number of links pertaining to the origin resource becomes so large that the response from the associated link set resource becomes too large. This could be the case for highly popular origin resources, when the link set includes links in which the origin resource participates as both link context and link target. In such cases, the link set resource could deliver responses incrementally, for example, using a paged resource model that clients can consume as required.

4. Document Formats for Link Sets

Link set resources can be represented by serializations that support conveying both link contexts and link targets. This section specifies two document formats based on the abstract model specified in Section 2 of [I-D.nottingham-rfc5988bis] that defines a link as consisting of a "link context", a "link relation type", a "link target", and optional "target attributes". In those terms, a link set resource is a resource that conveys a collection of one or more such links.

The format defined in Section 4.1 is identical to the payload of the HTTP Link header as specified in [I-D.nottingham-rfc5988bis]. The format defined in Section 4.2 is based on JSON and thus does not have the character encoding limitations of HTTP header fields.
4.1. HTTP Link Document Format: application/linkset

This document format is identical to the payload of the HTTP Link header. It is defined in Section 3 of [I-D.nottingham-rfc5988bis], more specifically by its ABNF production rule for "Link" and subsequent ones. The assigned media type for this format is "application/linkset".

4.2. JSON Document Format: application/linkset+json

This document format is based on the native Link header field representation defined by [I-D.nottingham-rfc5988bis], but uses JSON [RFC7159] as the syntax to represent the information contained in the native format. It is described in this section. The assigned media type for this format is "application/linkset+json".

4.2.1. Link Sets

In the JSON representation of a link set:

- A link set MUST be represented as a JSON array.
- A distinct JSON object - the "link object" - in this array MUST be used to represent an individual link or multiple links that have the same link context and link target. (see Section 4.2.2).
- Implementations MUST wrap link objects in an array, even if the link set contains only one link.
- Implementations MUST NOT include any members other than link objects in the array that represents a link set.

4.2.2. Links

In the JSON representation, individual links (as well as multiple links that have the same link context and link target) are represented by a JSON object, the link object. A link object adheres to the following rules:

- Each link object MUST have an "href" member with a value that represents the link target.
- Each link object MAY have an "anchor" member with a value that represents the link context. If this member is not provided, the link context is the link set resource.
- The value of both the "href" and "anchor" members MUST be a URI-reference as defined in Section 4.1 of [RFC3986]. Note that for
relative references, the baseURI is the URI of the link set resource.

- Each link object MUST have a "rel" or "rev" member (with the latter one deprecated) with a value that represents one or more link relation types, represented as an array of strings. The value of each "rel"/"rev" link relation type MUST either be a registered relation type or an extension relation type (URI), as described in Sections 2.1.1 and 2.1.2 of [I-D.nottingham-rfc5988bis], respectively.

- Each link object MAY have additional members representing target attributes (see Section 4.2.3).

The following example of a JSON-serialized link set represents one link with the core components of a link: link context, link relation type, and link target.

```
[ { "href" : "http://example.com/foo",
    "anchor" : "http://example.net/bar",
    "rel" : [ "next" ] } ]
```

The following example of a JSON-serialized link set represents two links with the core components of a link: link context, link relation type, and link target. The syntax of this example takes advantage of the fact that multiple relation types can be used in one link object to represent multiple links that have the same link context and link target.

```
[ { "href" : "http://example.com/foo",
    "anchor" : "http://example.net/bar",
    "rel" : [ "next", "http://example.net/linkrel" ] } ]
```

The previous example is equivalent to the following one, which does not use the syntax shortcut and therefore represents the two links individually.

```
[ { "href" : "http://example.com/foo",
    "anchor" : "http://example.net/bar",
    "rel" : [ "next" ] },
  { "href" : "http://example.com/foo",
    "anchor" : "http://example.net/bar",
    "rel" : [ "http://example.net/linkrel" ] } ]
```
4.2.3. Link Target Attributes

In many cases, a link is further qualified by target attributes, either defined by the serialization of [I-D.nottingham-rfc5988bis] or extension attributes defined and used by communities.

4.2.3.1. JSON Representation of RFC 5988bis Target Attributes

RFC 5988bis defines the following target attributes that may be used to annotate links: "hreflang", "media", "title", "title*", and "type". These target attributes follow different occurrence and value patterns and MUST be handled in the JSON representation as follows:

- **"hreflang"**: The optional and repeatable "hreflang" target attribute MUST be represented by an array (even if there only is one value to be represented), and each value in that array MUST be a string - representing one value of the "hreflang" target attributes for a link - which follows the same model as in the [I-D.nottingham-rfc5988bis] syntax.

- **"media"**: The optional and not repeatable "media" target attribute MUST be represented by a "media" member in the link object, and its value MUST be a string that follows the same model as in the [I-D.nottingham-rfc5988bis] syntax.

- **"type"**: The optional and not repeatable "type" target attribute MUST be represented by a "type" member in the link object, and its value MUST be a string that follows the same model as in the [I-D.nottingham-rfc5988bis] syntax.

- **"title"**: The optional and not repeatable "title" target attribute MUST be represented by a "title" member in the link object, and its value MUST be a string that follows the same model as in the [I-D.nottingham-rfc5988bis] syntax.

- **"title*"**: The optional and not repeatable "title*" target attribute is motivated by character encoding and language issues and follows the model defined in [RFC5987]. The details of the JSON representation that applies to title* are described in Section 4.2.3.2.

The following example illustrates how the repeatable "hreflang" and the not repeatable "type" target attributes are represented in a link object.
In addition to the target attributes described in Section 4.2.3, [I-D.nottingham-rfc5988bis] also supports attributes that follow the content model of [RFC5987]. In [I-D.nottingham-rfc5988bis], these target attributes are recognizable by the use of a trailing asterisk in the attribute name, such as "title*". The content model of [RFC5987] uses a string-based microsyntax that represents the character encoding, an optional language tag, and the escaped attribute value encoded according to the specified character encoding.

The JSON serialization for these attributes is as follows:

- Internationalized target attributes use the attribute name (including the *) as their name.
- The character encoding information as prescribed by [RFC5987] is not preserved; instead, the attribute value is represented in the character encoding used for the JSON link set.
- The attribute value is an array that contains one or more arrays. In each of these arrays, the first value is a string representing the attribute value (in its unescaped version), and the optional second array value is a language tag [RFC5646] associated with that attribute value.

The following example illustrates how the "title*" target attribute defined by [I-D.nottingham-rfc5988bis] is represented in a link object.

```json
[ { "href" : "http://example.com/foo",
  "anchor" : "http://example.net/bar",
  "rel" : [ "next" ],
  "hreflang" : [ "en", "de" ],
  "type" : "text/html" },

  "title*" : [ [ "nachstes Kapitel", "de" ] ] }
```

The above example assumes that the german title contains an umlaut character (which in the native syntax would be encoded as title*=UTF-8\'\"n\%c3\%a4chstes\%20Kapitel\"), which gets encoded in its unescaped form (but not shown here due to the limitations of RFC publication) in the JSON representation. Implementations MUST properly decode/
encode RFC 5987 style target attributes when transcoding between the native and the JSON syntax.

4.2.4. Extension Target Attributes

Extension target attributes are attributes that are not defined by RFC 5988bis (as listed in Section 4.2.3.1), but are nevertheless used to qualify links. They can be defined by communities in any way deemed necessary, and it is up to them to make sure their usage is understood by target applications. However, lacking standardization, there is no interoperable understanding of these extension attributes. One important consequence is that their cardinality is unknown to generic applications. Therefore, in the JSON serialization, all extension target attributes are treated as repeatable:

- Extension target attributes use the attribute name (including the *, if applicable) as their name.

- The value of an extension attribute MUST be represented by an array (even if there only is one value to be represented).

- If the extension attribute does not have a name with a trailing asterisk, then each value in that array MUST be a string that represents one value of the extension target attribute.

- If the extension attribute has a name with a trailing asterisk (it follows the content model of [RFC5987]), then each value in that array MUST be an array. The value of each array MUST be structured as described in Section 4.2.3.2.

5. The "linkset" Relation Type for Linking to Link Sets

The target of a link with the "linkset" relation type - the link set resource - provides a set of links, including links in which the link context - the origin resource - participates.

A link with the "linkset" relation type MAY be provided in the header and/or the body of the origin resource’s representation. It may also be discovered by other means, such as through client-side information.

More than one link with a "linkset" relation type MAY be provided. Multiple such links can refer to the same set of links expressed using different representations, or to different sets of links (potentially provided by different services).
The use of a link with the "linkset" relation type does not preclude the provision of links with other relation types, i.e. the origin resource can provide typed links other than a "linkset" link. Therefore, the effective set of links pertaining to the origin resource is the union of the links that the resource itself provides, and of all links - provided by the link set resource - in which it participates.

The link set resource MAY provide the links that pertain to the origin resource in its HTTP response header and/or body.

In common scenarios (the origin resource is distinct from the link set resource), it is essential for link set representations to make the URI of the origin resource explicit for those links in which the origin resource acts as link context.

A user agent that follows a "linkset" link from an origin resource MUST be aware that the link set resource can provide links in which the origin resource does not participate and MAY decide to ignore those links.

There is no constraint on the target URI of a link with the "linkset" relation type; designing and using these URIs is left to the discretion of implementers.

If an origin resource provides a "linkset" link pointing at a link set resource, and that link set resource provides a "linkset" link in turn, then this latter link points at links pertaining to the link set resource. This means that in the context of the latter link, the link set resource is an origin resource. This means that linkset relations are not transitive, and it is up to a client to decide whether they follow "nested chains" of "linkset" links or not.

6. Examples

Sections Section 6.1 and Section 6.2 show examples whereby the link set resource provides links pertaining to the origin resource, in its response header and body, respectively.

6.1. Links Provided in the Header of the Link Set Resource

Figure 1 shows a client issuing an HTTP head request against origin resource http://example.org/resource1.
HEAD /resource1 HTTP/1.1
Host: example.org
Connection: close

Figure 1: Client HTTP HEAD Request

Figure 2 shows the response to the HEAD request of Figure 1. The response contains a Link header with a link that uses the "linkset" relation type. It indicates that links pertaining to the origin resource are provided by link set resource http://example.com/links?uri=http%3A%2F%2Fexample.org%2Fresource.

HTTP/1.1 200 OK
Date: Mon, 28 Nov 2016 14:37:51 GMT
Server: Apache-Coyote/1.1
Link: <http://example.com/links?uri=http%3A%2F%2Fexample.org%2Fresource>; rel="linkset"; type="text/html"
Content-Length: 5214
Content-Type: text/html;charset=utf-8
Connection: close

Figure 2: Response to HTTP HEAD on Origin Resource

While in this example the URI of the link set resource uses a pattern that represents the URI of the origin resource, this is opaque to the client, which simply follows target URI to retrieving the link set resource.

Figure 3 shows the client issuing an HTTP GET request against the link set resource provided in Figure 2.

GET /links?uri=http%3A%2F%2Fexample.org%2Fresource HTTP/1.1
Host: example.com
Connection: close

Figure 3: Client HTTP GET against the Link Set Resource

Figure 4 shows the response headers to the HTTP GET request of Figure 3. The links pertaining to the origin resource are provided in the Link response header of the link set resource. As can be seen, in order to support an unambiguous determination of the link context of each link, the "anchor" attribute is always provided. Note that most, but not all, links have the origin resource as link context (anchor).
HTTP/1.1 200 OK
Date: Mon, 28 Nov 2016 14:40:02 GMT
Server: Apache-Coyote/1.1
Link: <http://authors.example.net/johndoe>; rel="author"
     <http://authors.example.net/janedoe>; rel="author"
     <http://example.org/resource1/items/AF48EF.pdf>; rel="item"
     <http://example.org/resource1/items/CB63DA.html>; rel="item"
     <http://example.net/resource41/>; rel="related"
Content-Type: text/html
Content-Length: 3018

Figure 4: Response to HTTP GET against the Link Set Resource

6.2. Links Provided in the Body of the Link Set Resource, Link Set
Serialized as application/linkset+json

Figure 5 is an example of a client issuing an HTTP head request against origin resource http://example.org/article?id=10.1371/journal.pone.0167475

HEAD article?id=10.1371/journal.pone.0167475 HTTP/1.1
Host: example.org
Connection: close

Figure 5: Client HTTP HEAD Request

Figure 6 shows the response to the HEAD request of Figure 5. The response contains a Link header with a link that has the "linkset" relation type. It indicates that links pertaining to the origin resource are provided by link set resource http://example.com/links/10.1371/journal.pone.0167475, which provides a representation with media type application/linkset+json.
Figure 6: Response to HTTP HEAD on Origin Resource

In this example, the URI of the linkset resource does not directly represent the URI of the origin resource anymore. There still is an association possible through the use of a URI pattern that includes identifying information. But as in the example above, the URI of the link set resource is opaque to the client that simply accesses the URI to retrieve the link set resource.

Figure 7 shows the client issuing an HTTP GET request against the link set resource provided in Figure 6.

GET /links/10.1371/journal.pone.0167475 HTTP/1.1
Host: example.com
Accept: application/linkset+json
Connection: close

Figure 7: Client HTTP GET against the Link Set Resource

Figure 8 shows the response headers to the HTTP GET request of Figure 7. The links pertaining to the origin resource are provided in the response body of the link set resource and are serialized according to the media type application/linkset+json.
Figure 8: Response to HTTP GET against the Link Set Resource

If Figure 6 would have provided a link to a link set with media type application/linkset, and if the client would have requested that link set, then the body of the response would have been similar to Figure 8. But it would have had application/linkset as Content-Type, the payload of the HTTP Link header of Figure 4 as body, and an accordingly adjusted value for Content-Length.

7. IANA Considerations

7.1. Link Relation Type: linkset

The link relation type below has been registered by IANA per Section 6.2.1 of Web Linking [I-D.nottingham-rfc5988bis]:

Relation Name: linkset

Description: The Target IRI of a link with the "linkset" relation type provides a set of links that pertain to the Context IRI of the link.

Reference: [[ This document ]]}
7.2. Media Type: application/linkset

7.2.1. IANA Considerations

The Internet media type [RFC6838] for a natively encoded link set is application/linkset.

Type name: application

Subtype name: linkset

Required parameters: none

Optional parameters: none

Encoding considerations: ...

Security considerations: ...

Interoperability considerations: ...

Published specification: [[ This document ]]

Applications that use this media type: ...

Additional information:

Magic number(s): N/A

File extension(s): This media type does not propose a specific extension.

Macintosh file type code(s): TEXT

Person & email address to contact for further information: Herbert Van de Sompel <herbertv@lanl.gov>

Intended usage: COMMON

Restrictions on usage: none

Author: Herbert Van de Sompel <herbertv@lanl.gov>

Change controller: IETF
7.3. Media Type: application/linkset+json

The Internet media type [RFC6838] for a JSON-encoded link set is application/linkset+json.

   Type name: application
   Subtype name: linkset+json
   Required parameters: none
   Optional parameters: none
   Encoding considerations: ...
   Security considerations: ...
   Interoperability considerations: ...

Published specification: [[ This document ]]

Applications that use this media type: ...

Additional information:

   Magic number(s): N/A

   File extension(s): JSON documents often use ".json" as the file extension, and this media type does not propose a specific extension other than this generic one.

   Macintosh file type code(s): TEXT

Person & email address to contact for further information: Herbert Van de Sompel <herbertv@lanl.gov>

Intended usage: COMMON

Restrictions on usage: none

Author: Herbert Van de Sompel <herbertv@lanl.gov>

Change controller: IETF
8. Security Considerations

...

9. Normative References

[I-D.nottingham-rfc5988bis]

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


Authors’ Addresses

Erik Wilde
CA Technologies
Email: erik.wilde@dret.net
URI: http://dret.net/netdret/

Herbert Van de Sompel
Los Alamos National Laboratory
Email: herbertv@lanl.gov
URI: http://public.lanl.gov/herbertv/