

**Linkset: Media Types and a Link Relation Type for Link Sets**  
**draft-wilde-linkset-05**

Abstract

This specification defines two media types for representing sets of links and a link relation type for referring to sets of links. The media types can be used to represent links in a standalone fashion, in one case in the native format as used in the HTTP "Link" header field, and in the other case in a JSON-based format. The link relation type can be used to point at a resource that provides a set of links.

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 July 3, 2020.

## Copyright Notice

Copyright (c) 2019 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 (<https://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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Scenarios . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Third-Party Links . . . . .	<a href="#">3</a>
<a href="#">3.2.</a>	Challenges Writing to HTTP Link Header Field . . . . .	<a href="#">4</a>
<a href="#">3.3.</a>	Large Number of Links . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Document Formats for Sets of Links . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	HTTP Link Document Format: application/linkset . . . . .	<a href="#">5</a>
<a href="#">4.2.</a>	JSON Document Format: application/linkset+json . . . . .	<a href="#">6</a>
<a href="#">4.2.1.</a>	Set of Links . . . . .	<a href="#">6</a>
<a href="#">4.2.2.</a>	Link Context Object . . . . .	<a href="#">7</a>
<a href="#">4.2.3.</a>	Link Target Object . . . . .	<a href="#">7</a>
<a href="#">4.2.4.</a>	Link Target Attributes . . . . .	<a href="#">9</a>
<a href="#">5.</a>	The "linkset" Relation Type for Linking to a Set of Links . .	<a href="#">13</a>
<a href="#">6.</a>	Examples . . . . .	<a href="#">14</a>
<a href="#">6.1.</a>	Set of Links Provided By Value as application/linkset . .	<a href="#">14</a>
<a href="#">6.2.</a>	Set of Links Provided By Reference as application/linkset+json . . . . .	<a href="#">15</a>
<a href="#">7.</a>	Implementation Status . . . . .	<a href="#">18</a>
<a href="#">7.1.</a>	Open Journal Systems (OJS) . . . . .	<a href="#">18</a>
<a href="#">8.</a>	IANA Considerations . . . . .	<a href="#">18</a>
<a href="#">8.1.</a>	Link Relation Type: linkset . . . . .	<a href="#">18</a>
<a href="#">8.2.</a>	Media Type: application/linkset . . . . .	<a href="#">19</a>
<a href="#">8.2.1.</a>	IANA Considerations . . . . .	<a href="#">19</a>
<a href="#">8.3.</a>	Media Type: application/linkset+json . . . . .	<a href="#">20</a>
<a href="#">9.</a>	Security Considerations . . . . .	<a href="#">21</a>
<a href="#">10.</a>	Normative References . . . . .	<a href="#">21</a>
<a href="#">Appendix A.</a>	Acknowledgements . . . . .	<a href="#">22</a>
<a href="#">Appendix B.</a>	JSON-LD Context . . . . .	<a href="#">22</a>
	Authors' Addresses . . . . .	<a href="#">23</a>



## **1. Introduction**

Resources on the Web often use typed Web Links [[RFC8288](#)], either as an embedded part of resource representations, for example using the <link> element for HTML representations, or by using the HTTP "Link" header field for representations using any media type. In some cases, however, providing links in this manner is impractical or impossible.

To that end, this specification defines two document formats and associated media types to represent sets of links. It also defines the "linkset" relation type that can be used to support discovery of a resource that conveys links.

## **2. Terminology**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

This specification uses the terms "link context" and "link target" as defined in [[RFC8288](#)]. 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.

## **3. Scenarios**

The following sections outline scenarios in which providing links by means of a standalone document instead of in an HTTP "Link" header field or as links embedded in the resource representation is advantageous or necessary. It is important to keep in mind that when providing links by means of a standalone representation and possibly linking to it, other links can still be provided by value (i.e., it is possible combine various mechanisms of how links are conveyed).

### **3.1. Third-Party Links**

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

- o Providing links in which the resource is involved not just as link context but also as link target.
- o Providing links pertaining to the resource that the server hosting that resource is not aware of.



- o External management of links pertaining to the resource in a special-purpose link management service.

In such cases, a specific resource can provide a set of links pertaining to a resource. This specific resource may be managed by the same or by another custodian as the linked resource. For clients intent on consuming links pertaining to the resource, it would be beneficial if the following conditions were met:

- o The resource is able to link to the specific resource that provides (additional) links using a well-known link relation type.
- o The link resource provides the links in a document that uses a well-defined media type.

These two needs (linking to a resource that represents a set of links, and providing media types for representing sets of links) are addressed in this specification through the link relation type and the two defined media types.

### **[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 means of a standalone document can be a solution. Making the resource that provides these links discoverable can be achieved by means of a typed link.

### **[3.3.](#) Large Number of Links**

When conveying links in an HTTP "Link" header field, 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 bound for their size. But when the number of links is unpredictable, estimating a reliable upper bound 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 bound 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 field, for example by limiting the links they provide to those with select relation types, thereby limiting the value of the HTTP "Link" header field to clients. Providing links by means of a standalone document overcomes challenges related to the unpredictable nature of the size of HTTP "Link" header fields.

In more extreme scenarios it is conceivable that the number of links to be conveyed becomes so large that even a standalone document would become too large. For example, this could be the case for highly interlinked resources and when links are provided in which such resources participates as both link context and link target. In such cases, the links could be delivered incrementally, for example, by means of a paged resource model.

#### **4. Document Formats for Sets of Links**

A set of links 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 Web Linking [[RFC8288](#)] that defines a link as consisting of a "link context", a "link relation type", a "link target", and optional "target attributes".

The format defined in [Section 4.1](#) is identical to the payload of the HTTP "Link" header field as specified in Web Linking [[RFC8288](#)].

The format defined in [Section 4.2](#) is based on JSON but encodes the same abstract model as defined in Web Linking [[RFC8288](#)].

##### **4.1. HTTP Link Document Format: application/linkset**

This document format is identical to the payload of the HTTP "Link" header field as defined in [Section 3 of \[RFC8288\]](#), more specifically by its ABNF production rule for "Link" and subsequent ones.

The assigned media type for this format is "application/linkset".

In order to support use cases where "application/linkset" documents are re-used outside the context of an HTTP interaction, it is RECOMMENDED to make them self-contained by adhering to the following guidelines:

- o For every link provided in the set of links, explicitly provide the link context using the "anchor" attribute.
- o For link context ("anchor" attribute) and link target ("href" attribute), use absolute URIs (as defined in [Section 4.3 of \[RFC3986\]](#)).





If these recommendations are not followed, interpretation of the "application/linkset" will depend on the which URI is used as context URI.

#### **4.2. JSON Document Format: application/linkset+json**

This document format uses JSON [[RFC8259](#)] as the syntax to represent a set of links. The set of links follows the abstract model defined by Web Linking [[RFC8288](#)].

The assigned media type for this format is "application/linkset+json".

In order to support use cases where "application/linkset+json" documents are re-used outside the context of an HTTP interaction, it is RECOMMENDED to make them self-contained by adhering to the following guidelines:

- o For every link provided in the set of links, explicitly provide the link context using the "anchor" member.
- o For link context ("anchor" member) and link target ("href" member), use absolute URIs (as defined in [Section 4.3 of \[RFC3986\]](#)).

The "application/linkset+json" serialization is designed such that it can directly be used as the content of a JSON-LD serialization by adding an appropriate context. [Appendix B](#) shows an example of a possible context that, when added to the a JSON serialization, allows it to be interpreted as RDF.

##### **4.2.1. Set of Links**

In the JSON representation of a set of links:

- o A set of links MUST be represented as a JSON object which MUST have "linkset" as its sole member.
- o The "linkset" member is an array in which a distinct JSON object - the "link context object" (see [Section 4.2.2](#)) - MUST be used to represent links that have the same link context.
- o If necessary, the "linkset" member MAY contain information in addition to link context objects, in which case that information MUST NOT change the semantics of the links provided by those link context objects.



- o Even if there is only one link context object, it MUST be wrapped in an array. Members other than link context objects MUST NOT be included in this array.

#### **4.2.2. Link Context Object**

In the JSON representation, one or more links that have the same link context are represented by a JSON object, the link context object. A link context object adheres to the following rules:

- o Each link context object MUST have an "anchor" member with a value that represents the link context. This value SHOULD be an absolute URI as defined in [Section 4.3 of \[RFC3986\]](#). Cases whereby no value is to be provided for the "anchor" member (i.e. the resource providing the set of links is the link context for each link in the link context object) MUST be handled by providing an "anchor" member with null value ("anchor": null).
- o For each distinct relation type that the link context has with link targets a link context object MUST have an additional member. This member is an array in which a distinct JSON object - the "link target object" (see [Section 4.2.3](#)) - MUST be used for each link target for which the relationship with the link context (value of the encompassing anchor member) applies. The name of this member expresses the relation type of the link as follows:
  - o
    - \* For registered relation types [\[RFC8288\]](#), the name of this member is the registered name of the relation type.
    - \* For extension relation types [\[RFC8288\]](#), the name of this member is the URI that uniquely represents the relation type.
- o Even if there is only one link target object it MUST be wrapped in an array. Members other than link target objects MUST NOT be included in this array.

#### **4.2.3. Link Target Object**

In the JSON representation a link target is represented by a JSON object, the link target object. A link target object adheres to the following rules:

- o Each link target object MUST have an "href" member with a value that represents the link target. This value SHOULD be an absolute URI as defined in [Section 4.3 of \[RFC3986\]](#). Cases whereby no value is to be provided for the "href" member (i.e. the resource



providing the set of links is the target of the link in the link target object) MUST be handled by providing an "href" member with null value ("href": "").

- o In many cases, a link target is further qualified by attributes, either defined by the serialization of Web Linking [RFC8288] or extension attributes defined and used by communities. These attributes are conveyed as additional members of the link target object as detailed in [Section 4.2.4.1](#), [Section 4.2.4.2](#), and [Section 4.2.4.3](#).

Note that the JSON representation does not use the "rel" or "rev" constructs used by Web Linking [RFC8288] and the above specified HTTP "Link" document format. Rather:

- o A link that uses the "rel" construct in those approaches is conveyed using the URI of the link context as the value for "anchor" and the URI of the link target as the value for "href".
- o A link that uses the "rev" construct in those approaches is conveyed using the URI of the link target as the value for "anchor" and the URI of the link context as the value for "href".

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

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

The following example of a JSON-serialized set of links represents two links that share link context and relation type but have different link targets.



```
{
  "linkset":
  [
    { "anchor": "http://example.net/bar",
      "item": [
        {"href": "http://example.com/foo1"},
        {"href": "http://example.com/foo2"}
      ]
    }
  ]
}
```

The following example shows a set of links that represents two links, each with a different link context, link target, and relation type. One relation type is registered, the other is an extension relation type.

```
{
  "linkset":
  [
    { "anchor": "http://example.net/bar",
      "next": [
        {"href": "http://example.com/foo1"}
      ]
    },
    { "anchor": "http://example.net/boo",
      "http://example.com/relations/baz" : [
        {"href": "http://example.com/foo2"}
      ]
    }
  ]
}
```

#### **4.2.4. Link Target Attributes**

A link may be further qualified by target attributes, either defined by the serialization of Web Linking [[RFC8288](#)] or extension attributes defined and used by communities.

##### **4.2.4.1. Target Attributes Defined by [RFC 8288](#)**

[RFC 8288](#) 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. In the JSON representation, these attributes MUST be conveyed as additional members of the link target object as follows:





- o "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 attribute for a link - which follows the same model as in the [\[RFC8288\]](#) syntax.
- o "media": The optional and not repeatable "media" target attribute MUST be represented by a "media" member in the link target object, and its value MUST be a string that follows the same model as in the [\[RFC8288\]](#) syntax.
- o "type": The optional and not repeatable "type" target attribute MUST be represented by a "type" member in the link target object, and its value MUST be a string that follows the same model as in the [\[RFC8288\]](#) syntax.
- o "title": The optional and not repeatable "title" target attribute MUST be represented by a "title" member in the link target object, and its value MUST be a string that follows the same model as in the [\[RFC8288\]](#) syntax.
- o "title\*": The optional and not repeatable "title\*" target attribute is motivated by character encoding and language issues and follows the model defined in [\[RFC8187\]](#). The details of the JSON representation that applies to title\* are described in [Section 4.2.4.2](#).

The following example illustrates how the repeatable "hreflang" and the not repeatable "type" target attributes are represented in a link target object.

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



#### **4.2.4.2. Internationalized Target Attributes**

In addition to the target attributes described in [Section 4.2.4.1](#), [\[RFC8288\]](#) also supports attributes that follow the content model of [\[RFC8187\]](#). In [\[RFC8288\]](#), these target attributes are recognizable by the use of a trailing asterisk in the attribute name, such as "title\*". The content model of [\[RFC8187\]](#) 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 target attributes MUST be as follows:

- o An internationalized target attribute is represented as a member of the link context object with the same name (including the \*) of the attribute.
- o The character encoding information as prescribed by [\[RFC8187\]](#) is not preserved; instead, the content of the internationalized attribute is represented in the character encoding used for the JSON set of links.
- o The value of the internationalized target attribute is an array that contains one or more JSON objects. The name of the first member of such JSON object is "value" and its value is the actual content (in its unescaped version) of the internationalized target attribute, i.e. the value of the attribute from which the encoding and language information are removed. The name of the optional second member of such JSON object is "language" and its value is the language tag [\[RFC5646\]](#) for the language in which the attribute content is conveyed.

The following example illustrates how the "title\*" target attribute defined by [\[RFC8288\]](#) is represented in a link target object.



```
{
  "linkset":
  [
    { "anchor": "http://example.net/bar",
      "next": [
        { "href":      "http://example.com/foo",
          "type":      "text/html",
          "hreflang":  [ "en" , "de" ],
          "title":     "Next chapter",
          "title*":    [ { "value": "nachstes Kapitel" , "language" :
"de" } ]
        }
      ]
    }
  ]
}
```

The above example assumes that the German title contains an umlaut character (in the native syntax it would be encoded as `title*=UTF-8'de'n%c3%a4chstes%20Kapitel`), which gets encoded in its unescaped form in the JSON representation. This is not shown in the above example due to the limitations of RFC publication. Implementations MUST properly decode/encode internationalized target attributes that follow the model of [\[RFC8187\]](#) when transcoding between the "application/linkset" and the "application/linkset+json" formats.

#### **[4.2.4.3](#). Extension Target Attributes**

Extension target attributes are attributes that are not defined by [RFC 8288](#) (as listed in [Section 4.2.4.1](#)), 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.

The JSON serialization for these target attributes MUST be as follows:

- o An extension target attribute is represented as a member of the link context object with the same name of the attribute, including the \* if applicable.
- o The value of an extension attribute MUST be represented by an array, even if there only is one value to be represented.



- o If the extension target 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 attribute.
- o If the extension attribute has a name with a trailing asterisk (it follows the content model of [RFC8187]), then each value in that array MUST be a JSON object. The value of each such JSON object MUST be structured as described in [Section 4.2.4.2](#).

The example shows a link target object with three extension target attributes. The value for each extension target attribute is an array. The two first are regular extension target attributes, with the first one ("foo") having only one value and the second one ("bar") having two. The last extension target attribute ("baz\*") follows the naming rule of [RFC8187] and therefore is encoded according to the serialization described in [Section 4.2.4.2](#).

```
{
  "linkset":
  [
    { "anchor": "http://example.net/bar",
      "next": [
        { "href": "http://example.com/foo",
          "type": "text/html",
          "foo": [ "foovalue" ],
          "bar": [ "barone", "bartwo" ],
          "baz*": [ { "value": "bazvalue" , "language" : "en" } ]
        }
      ]
    }
  ]
}
```

## 5. The "linkset" Relation Type for Linking to a Set of Links

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

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

A resource MAY provide more than one link with a "linkset" relation type. Multiple such links can refer to the same set of links expressed using different media types, or to different sets of links, potentially provided by different third-party services.





A user agent that follows a "linkset" link **MUST** be aware that the set of links provided by the resource that is the target of the link can contain links in which the resource that is the context of the link does not participate; it **MAY** decide to ignore those links.

A user agent that follows a "linkset" link and obtains links for which anchors and targets are not expressed as absolute URIs **MUST** properly determine what the context is for these links; it **SHOULD** ignore links for which it is unable to unambiguously make that determination.

## **6. Examples**

[Section 6.1](#) and [Section 6.2](#) show examples whereby the set of links are provided as "application/linkset" and "application/linkset+json" documents, respectively.

### **[6.1.](#) Set of Links Provided By Value as application/linkset**

Figure 1 shows a client issuing an HTTP GET request against resource `http://example.org/resource1`.

```
GET /resource1 HTTP/1.1
Host: example.org
Connection: close
```

Figure 1: Client HTTP GET request

Figure 2 shows the response to the GET request of Figure 1. The response contains a Content-Type header specifying `application/linkset`. A set of links, including links that pertain to the responding resource, is provided in the body of the response.



```
HTTP/1.1 200 OK
Date: Mon, 12 Aug 2019 10:35:51 GMT
Server: Apache-Coyote/1.1
Content-Length: 729
Content-Type: application/linkset
Connection: close

<http://authors.example.net/johndoe>
  ; rel="author"
  ; type="application/rdf+xml"
  ; anchor="http://example.org/resource1",
<http://authors.example.net/janedoe>
  ; rel="author"
  ; type="application/rdf+xml"
  ; anchor="http://example.org/resource1",
<http://example.org/resource1/items/AF48EF.pdf>
  ; rel="item"
  ; type="application/pdf"
  ; anchor="http://example.org/resource1",
<http://example.org/resource1/items/CB63DA.html>
  ; rel="item"
  ; type="text/html"
  ; anchor="http://example.org/resource1",
<http://example.org/resource1>
  ; rel="latest-version"
  ; anchor="http://example.org/resource41/",
<http://example.org/resource40>
  ; rel="prev"
  ; anchor="http://example.org/resource41/"
```

Figure 2: Response to HTTP GET includes a set of links

## **6.2. Set of Links Provided By Reference as application/linkset+json**

Figure 3 shows a client issuing an HTTP HEAD request against resource `http://example.org/article/view/7507`

```
HEAD article/view/7507 HTTP/1.1
Host: example.org
Connection: close
```

Figure 3: Client HTTP HEAD request

Figure 4 shows the response to the HEAD request of Figure 3. The response contains a Link header with a link that has the "linkset" relation type. It indicates that a set of links is provided by



resource `http://example.com/links/article/7507`, which provides a representation with media type `application/linkset+json`.

```
HTTP/1.1 200 OK
Date: Mon, 12 Aug 2019 10:45:54 GMT
Server: Apache-Coyote/1.1
Link: <http://example.com/links/article/7507>
      ; rel="linkset"
      ; type="application/linkset+json"
Content-Length: 236
Content-Type: text/html; charset=utf-8
Connection: close
```

Figure 4: Response to HTTP HEAD request

Figure 5 shows the client issuing an HTTP GET request against the target IRI of the "linkset" link provided in Figure 4. In the request, the client uses an Accept header to indicate it requests a response in "application/linkset+json" format.

```
GET links/article/7507 HTTP/1.1
Host: example.com
Accept: application/linkset+json
Connection: close
```

Figure 5: Client issues HTTP GET to obtain set of links

Figure 6 shows the response to the HTTP GET request of Figure 5. The set of links is serialized according to the media type "application/linkset+json" and includes links in which the Context IRI of the "linkset" link of Figure 4 participates. It also includes a link in which it does not.



```
HTTP/1.1 200 OK
Date: Mon, 12 Aug 2019 10:46:22 GMT
Server: Apache-Coyote/1.1
Content-Type: application/linkset+json
Content-Length: 802

{
  "linkset": [
    {
      "anchor": "https://example.org/article/view/7507",
      "author": [
        {
          "href": "https://orcid.org/0000-0002-4311-0897",
          "type": "rdf/xml"
        }
      ],
      "item": [
        {
          "href": "https://example.org/article/7507/item/1",
          "type": "application/pdf"
        },
        {
          "href": "https://example.org/article/7507/item/2",
          "type": "text/csv"
        }
      ],
      "cite-as": [
        {
          "href": "https://doi.org/10.841/zk2557"
        }
      ]
    },
    {
      "anchor": "https://example.com/links/article/7507",
      "alternate": [
        {
          "href": "https://mirror.example.com/links/article/7507",
          "type": "application/linkset"
        }
      ]
    }
  ]
}
```

Figure 6: Response to the client's request for the set of links





## **7. Implementation Status**

Note to RFC Editor: Please remove this section before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC 6982](#) [[RFC6982](#)]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC 6982](#), "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

### **7.1. Open Journal Systems (OJS)**

Open Journal Systems (OJS) is an open-source software for the management of peer-reviewed academic journals, and is created by the Public Knowledge Project (PKP), released under the GNU General Public License. Open Journal Systems (OJS) is a journal management and publishing system that has been developed by PKP through its federally funded efforts to expand and improve access to research.

The OJS platform has implemented "linkset" support as an alternative way to provide links when there are more than a configured limit (they consider using about 10 as a good default, for testing purpose it is currently set to 8).

## **8. IANA Considerations**

### **8.1. Link Relation Type: linkset**

The link relation type below has been registered by IANA per [Section 6.2.1](#) of Web Linking [[RFC8288](#)]:

Relation Name: linkset



Description: The Target IRI of a link with the "linkset" relation type provides a set of links, including links in which the Context IRI of the link participates.

Reference: [[ This document ]]

## **8.2. Media Type: application/linkset**

### **8.2.1. 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: Erik Wilde <erik.wilde@dret.net>

Intended usage: COMMON

Restrictions on usage: none

Author: Erik Wilde <erik.wilde@dret.net>



Change controller: IETF

### **8.3. 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: Erik Wilde <erik.wilde@dret.net>

Intended usage: COMMON

Restrictions on usage: none

Author: Erik Wilde <erik.wilde@dret.net>

Change controller: IETF



## 9. Security Considerations

...

## 10. 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", [BCP 47](#), [RFC 5646](#), DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [RFC5988] Nottingham, M., "Web Linking", [RFC 5988](#), DOI 10.17487/RFC5988, October 2010, <<https://www.rfc-editor.org/info/rfc5988>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [RFC 6982](#), DOI 10.17487/RFC6982, July 2013, <<https://www.rfc-editor.org/info/rfc6982>>.
- [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, <<https://www.rfc-editor.org/info/rfc7231>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8187] Reschke, J., "Indicating Character Encoding and Language for HTTP Header Field Parameters", [RFC 8187](#), DOI 10.17487/RFC8187, September 2017, <<https://www.rfc-editor.org/info/rfc8187>>.





- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, [RFC 8259](#), DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8288] Nottingham, M., "Web Linking", [RFC 8288](#), DOI 10.17487/RFC8288, October 2017, <<https://www.rfc-editor.org/info/rfc8288>>.

## [Appendix A.](#) Acknowledgements

Thanks for comments and suggestions provided by Mark Nottingham, Stian Soiland-Reyes, Sarven Capadisli, ...

## [Appendix B.](#) JSON-LD Context

When adding the below context to the representation of a link set rendered according to the JSON serialization defined in [Section 4.2](#), it can be interpreted as RDF.



```
{
  "@context": [
    {
      "@vocab": "https://www.iana.org/assignments/link-relations/",
      "anchor": "@id",
      "href": "@id",
      "dct": "http://purl.org/dc/terms/",
      "link": "https://www.iana.org/assignments/link-relations#",
      "title": {
        "@id": "http://purl.org/dc/terms/title"
      },
      "title*": {
        "@id": "http://purl.org/dc/terms/title"
      },
      "type": {
        "@id": "dct:format",
        "@type": "@vocab"
      }
    },
    {
      "language": "@language",
      "value": "@value",
      "hreflang": {
        "@id": "https://www.w3.org/ns/activitystreams#hreflang",
        "@container": "@set"
      }
    }
  ]
}
```

Figure 7: Context for JSON-LD that could be used with the JSON serialization

#### Authors' Addresses

Erik Wilde

Email: [erik.wilde@dret.net](mailto:erik.wilde@dret.net)

URI: <http://dret.net/netdret/>

Herbert Van de Sompel

Data Archiving and Networked Services

Email: [herbert.van.de.sompel@dans.knaw.nl](mailto:herbert.van.de.sompel@dans.knaw.nl)

URI: <https://orcid.org/0000-0002-0715-6126>

