Registration Protocols Extensions (regext)
Internet-Draft
Intended status: Standards Track
Expires: 5 March 2026

G. Brown
A. Newton
ICANN

1 September 2025

Efficient RDAP Referrals
draft-ietf-regext-rdap-referrals-00

Abstract

This document describes how RDAP servers can provide HTTP "Link" header fields in RDAP responses to allow RDAP clients to efficiently determine the URL of related RDAP records for a resource.

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 5 March 2026.

Table of Contents

1. Introduction

Many Registration Data Access Protocol (RDAP, described in [RFC7480], [RFC7481], [RFC9082], [RFC9083] and others) resources contain referrals to related RDAP resources.

For example, in the domain space, an RDAP record for a domain name received from the registry operator may include a referral to the RDAP record for the same object provided by the sponsoring registrar, while in the IP address space, an RDAP record for an address allocation may include referrals to enclosing or sibling prefixes.

In both cases, RDAP service users are often equally if not more interested in these related RDAP resources than the resource provided by the TLD registry or RIR.

While RDAP supports redirection of RDAP requests using HTTP redirections (which use a 3xx HTTP status and the "Location" header field, see Section 15.4 of [RFC9110]), it is not possible for RDAP servers to know _a priori_ whether a client requesting an RDAP record is doing so because it wants to retrieve a related RDAP record, or its own, so it can only respond by providing the full RDAP response. The client must then parse that response in order to extract the relevant URL from the "links" property of the object.

This results in the wasteful expenditure of time, compute resources and bandwidth on the part of both the client and server.

This document describes how an RDAP server can use "Link" HTTP header fields in responses to HEAD and GET requests to provide RDAP clients with the URL of related RDAP records, without the need for a signalling mechanism for the client to tell the server that it is only interested in retrieving those URLs.

2. RDAP Link Objects

RDAP link objects, described in Section 4.2 of [RFC9083], establish unidirectional relationships between an RDAP resource and other web resources, which may also be RDAP resources. The "rel" property indicates the nature of the relationship, and its possible values are described in [RFC8288].

If a link object has a "type" property which contains the value "application/rdap+json", then clients can assume that the linked resource is also an RDAP resource.

In the domain name space, this allows clients to discover the URL of the sponsoring registrar’s RDAP record for a given domain name, if the "rel" property has the value "related", while in the IP address space, the "up" and "down" values allow RDAP clients to navigate the hierarchy of address space allocations.

3. HTTP "Link" Header Field

"Link" header fields, described in Section 3 of [RFC8288], provide a means for describing a relationship between two resources, generally between the requested resource and some other resource. The "Link" header field is semantically equivalent to the <link> element in HTML, and multiple "Link" headers may be present in the header of an HTTP response.

"Link" header fields may contain most of the parameters that are also present in Link objects in RDAP responses (See Section 4.2 of [RFC9083]). So for example, an RDAP link object which has the following JSON representation:

```
{
  "value" : "https://example.com/context_uri",
  "rel" : "self",
  "href" : "https://example.com/target_uri",
  "hreflang" : [ "en", "ch" ],
  "title" : "title",
  "media" : "screen",
  "type" : "application/json"
}
```

may be represented in an HTTP response header as follows:

```
Link: <https://example.com/target_uri>;
  rel="self";
  hreflang="en,ch";
  title="title";
  media="screen";
  type="application/json"
```

In this example, the context URI is the URI that was requested by the user agent.

3.1. Registrar RDAP "Link" Header

Following on from the above, the following RDAP link object, which represents the RDAP URL of the sponsoring registrar of a resource:

```
{
  "value": "https://rdap.example.com/domain/example.com",
  "title": "URL of Sponsoring Registrar's RDAP Record",
  "rel": "related",
  "href": "https://rdap.example.com/domain/example.com",
  "type": "application/rdap+json"
}
```

may be represented as follows:

```
Link: <https://rdap.example.com/domain/example.com>;
  title="URL of Sponsoring Registrar's RDAP Record";
  rel="related";
  type="application/rdap+json"
```

4. RDAP Responses

In response to GET and HEAD RDAP requests, RDAP servers which implement this specification MUST include a "Link" header field for each link object which refers to an RDAP resource that is present in the "links" array of the object in question. The server MAY also include "Link" headers for link objects which refer to other types of resource. In all cases, the link attributes MUST be the same in both places.

4.1. RDAP HEAD requests

The HTTP HEAD method can be used for obtaining metadata about a resource without transferring that resource (see Section 4.3.2 of [RFC7231]).

An RDAP client which only wishes to obtain the URLs of related RDAP resources can issue a HEAD request for an RDAP resource and check the response for the presence of an appropriate "Link" header field. If the link is absent, it may then fall back to performing a GET request.

An RDAP client interested in both the server's record and related records can use the traditional method of performing a GET request and extracting the link objects from the response. To improve performance, RDAP clients MAY inspect the header of a response, extract the link headers, and issue requests for the related record in parallel while the request to the server is still in flight. As an example, the cURL library provides the CURLOPT_HEADERFUNCTION (https://curl.se/libcurl/c/CURLOPT_HEADERFUNCTION.html) configuration option to provide a callback that is invoked as soon as it has received header data.

5. RDAP Conformance

Servers which implement this specification MUST include the string "link_headers" in the "rdapConformance" array in all RDAP responses.

6. IANA Considerations

IANA is requested to register the following value in the RDAP Extensions Registry:

Extension identifier: link_headers
Registry operator: any.
Published specification: this document.
Contact: the authors of this document.
Intended usage: this extension indicates that the server will provide links to related resources using "Link" headers in responses to RDAP queries.

7. Change log

7.1. Changes from draft-brown-rdap-referrals-02 to draft-ietf-regext-rdap-referrals-00

* Nothing apart from the name.

7.2. Changes from 01 to 02

* add this change log.

7.3. Changes from 00 to 01

* change extension identifer from registrar_link_header to link_headers.

8. Normative References

[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>.

[RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", STD 95, RFC 7480, DOI 10.17487/RFC7480, March 2015, <https://www.rfc-editor.org/info/rfc7480>.

[RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", STD 95, RFC 7481, DOI 10.17487/RFC7481, March 2015, <https://www.rfc-editor.org/info/rfc7481>.

[RFC8288] Nottingham, M., "Web Linking", RFC 8288, DOI 10.17487/RFC8288, October 2017, <https://www.rfc-editor.org/info/rfc8288>.

[RFC9082] Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

[RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

Authors' Addresses

Gavin Brown
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90292
United States of America
Email: gavin.brown@icann.org
URI: https://icann.org

Andy Newton
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90292
United States of America
Email: andy.newton@icann.org
URI: https://icann.org