

HTTPAPI
Internet-Draft
Intended status: Standards Track
Expires: 26 June 2021

S. Dalal
E. Wilde
23 December 2020

The Deprecation HTTP Header
draft-ietf-httpapi-deprecation-header-00

Abstract

The HTTP Deprecation Response Header can be used to signal to consumers of a URI-identified resource that the resource has been deprecated. Additionally, the deprecation link relation can be used to link to a resource that provides additional context for the deprecation, and possibly ways in which clients can find a replacement for the deprecated 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 26 June 2021.

Copyright Notice

Copyright (c) 2020 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

- [1. Introduction](#) [2](#)
- [1.1. Notational Conventions](#) [3](#)
- [2. The Deprecation HTTP Response Header](#) [3](#)
- [2.1. Syntax](#) [3](#)
- [3. The Deprecation Link Relation Type](#) [4](#)
- [3.1. Documentation](#) [4](#)
- [4. Recommend Replacement](#) [5](#)
- [5. Sunset](#) [6](#)
- [6. Resource Behavior](#) [6](#)
- [7. IANA Considerations](#) [6](#)
- [7.1. The Deprecation HTTP Response Header](#) [6](#)
- [7.2. The Deprecation Link Relation Type](#) [7](#)
- [8. Implementation Status](#) [7](#)
- [8.1. Implementing the Deprecation Header](#) [8](#)
- [8.2. Implementing the Concept](#) [9](#)
- [9. Security Considerations](#) [10](#)
- [10. Examples](#) [11](#)
- [11. References](#) [11](#)
- [11.1. Normative References](#) [11](#)
- [11.2. Informative References](#) [12](#)
- [Appendix A. Acknowledgments](#) [13](#)
- [Authors' Addresses](#) [13](#)

1. Introduction

Deprecation of an HTTP resource as defined in [Section 2 of \[RFC7231\]](#) is a technique to communicate information about the lifecycle of a resource. It encourages applications to migrate away from the resource, discourages applications from forming new dependencies on the resource, and informs applications about the risk of continuing dependence upon the resource.

The act of deprecation does not change any behavior of the resource. It just informs client of the fact that a resource is deprecated. The Deprecation HTTP response header field MAY be used to convey this fact at runtime to clients. The header field can carry information indicating since when the deprecation is in effect.

In addition to the Deprecation header field the resource provider can use other header fields to convey additional information related to deprecation. For example, information such as where to find documentation related to the deprecation or what should be used as an alternate and when the deprecated resource would be unreachable, etc. Alternates of a resource can be similar resource(s) or a newer version of the same resource.

1.1. Notational Conventions

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 Augmented Backus-Naur Form (ABNF) notation of [[RFC5234](#)] and includes, by reference, the IMF-fixdate rule as defined in [Section 7.1.1.1 of \[RFC7231\]](#).

The term "resource" is to be interpreted as defined in [Section 2 of \[RFC7231\]](#), that is identified by an URI.

2. The Deprecation HTTP Response Header

The "Deprecation" HTTP response header field allows a server to communicate to a client that the resource in context of the message is or will be deprecated.

2.1. Syntax

The "Deprecation" response header field describes the deprecation. It either shows the deprecation date, which may be in the future (the resource context will be deprecated at that date) or in the past (the resource context has been deprecated at that date), or it simply flags the resource context as being deprecated:

```
Deprecation = IMF-fixdate / "true"
```

Servers MUST NOT include more than one "Deprecation" header field in the same response.

The date, if present, is the date when the resource context was or will be deprecated. It is in the form of an IMF-fixdate timestamp.

The following example shows that the resource context has been deprecated on Friday, November 11, 2018 at 23:59:59 GMT:

```
Deprecation: Sun, 11 Nov 2018 23:59:59 GMT
```

The deprecation date can be in the future. If the value of "date" is in the future, it means that the resource will be deprecated at the given date in future.

If the deprecation date is not known, the header field can carry the simple string "true", indicating that the resource context is deprecated, without indicating when that happened:

```
Deprecation: true
```

3. The Deprecation Link Relation Type

In addition to the Deprecation HTTP header field, the server can use links with the "deprecation" link relation type to communicate to the client where to find more information about deprecation of the context. This can happen before the actual deprecation, to make a deprecation policy discoverable, or after deprecation, when there may be documentation about the deprecation, and possibly documentation of how to manage it.

This specification places no restrictions on the representation of the interlinked deprecation policy. In particular, the deprecation policy may be available as human-readable documentation or as machine-readable description.

3.1. Documentation

For a resource, deprecation could involve one or more parts of request, response or both. These parts could be one or more of the following.

- * URI - deprecation of one or more query parameter(s) or path element(s)
- * method - HTTP method for the resource is deprecated
- * request header - one or more HTTP request header(s) is deprecated
- * response header - HTTP response header(s) is deprecated
- * request body - request body contains one or more deprecated element(s)
- * response body - response body contains one or more deprecated element(s)

The purpose of the "Deprecation" header is to provide just enough "hints" about the deprecation to the client application developer. It is safe to assume that on reception of the "Deprecation" header, the client developer would look up the resource's documentation in order to find deprecation related semantics. The resource developer could provide a link to the resource documentation using a "Link" header with relation type "deprecation" as shown below.

```
Link: <https://developer.example.com/deprecation>;  
      rel="deprecation"; type="text/html"
```

In this example the interlinked content provides additional information about the deprecation of the resource context. There is no Deprecation header field in the response, and thus the resource is not deprecated. However, the resource already exposes a link where information is available how deprecation is managed for the context. This may be documentation explaining the use of the Deprecation header field, and also explaining under which circumstances and with which policies (announcement before deprecation; continued operation after deprecation) deprecation might be happening.

The following example uses the same link header, but also announces a deprecation date using a Deprecation header field.

```
Deprecation: Sun, 11 Nov 2018 23:59:59 GMT  
Link: <https://developer.example.com/deprecation>;  
      rel="deprecation"; type="text/html"
```

Given that the deprecation date is in the past, the linked resource may have been updated to include information about the deprecation, allowing clients to discover information about the deprecation that happened.

4. Recommend Replacement

The "Link" [[RFC8288](#)] header field can be used in addition to the "Deprecation" header field to inform the client about available alternatives to the deprecated resource. The following relation types as defined in [[RFC8288](#)] are RECOMMENDED to use for this purpose:

- * "successor-version": Points to a resource containing the successor version. [[RFC5829](#)]
- * "latest-version": Points to a resource containing the latest (e.g., current) version. [[RFC5829](#)]
- * "alternate": Designates a substitute. [W3C.REC-html401-19991224]

The following example provides link to the successor version of the requested resource that is deprecated.

Deprecation: Sun, 11 Nov 2018 23:59:59 GMT

Link: <https://api.example.com/v2/customers>; rel="successor-version"

This example provides link to an alternate resource to the requested resource that is deprecated.

Deprecation: Sun, 11 Nov 2018 23:59:59 GMT

Link: <https://api.example.com/v1/clients>; rel="alternate"

5. Sunset

In addition to the deprecation related information, if the resource provider wants to convey to the client application that the deprecated resource is expected to become unresponsive at a specific point in time, the Sunset HTTP header field [[RFC8594](#)] can be used in addition to the "Deprecation" header.

The timestamp given in the "Sunset" header field MUST be the later or the same as the one given in the "Deprecation" header field.

The following example shows that the resource in context has been deprecated since Sunday, November 11, 2018 at 23:59:59 GMT and its sunset date is Wednesday, November 11, 2020 at 23:59:59 GMT.

Deprecation: Sun, 11 Nov 2018 23:59:59 GMT

Sunset: Wed, 11 Nov 2020 23:59:59 GMT

6. Resource Behavior

The act of deprecation does not change any behavior of the resource. Deprecated resources SHOULD keep functioning as before, allowing consumers to still use the resources in the same way as they did before the resources were declared deprecated.

7. IANA Considerations

7.1. The Deprecation HTTP Response Header

The "Deprecation" response header should be added to the permanent registry of message header fields (see [[RFC3864](#)]), taking into account the guidelines given by HTTP/1.1 [[RFC7231](#)].

Header Field Name: Deprecation

Applicable Protocol: Hypertext Transfer Protocol (HTTP)

Status: Standard

Author: Sanjay Dalal <sanjay.dalal@cal.berkeley.edu>,
Erik Wilde <erik.wilde@dret.net>

Change controller: IETF

Specification document: this specification,
[Section 2](#) "The Deprecation HTTP Response Header"

7.2. The Deprecation Link Relation Type

The "deprecation" link relation type should be added to the permanent registry of link relation types according to [Section 4.2 of \[RFC8288\]](#):

Relation Type: deprecation

Applicable Protocol: Hypertext Transfer Protocol (HTTP)

Status: Standard

Author: Sanjay Dalal <sanjay.dalal@cal.berkeley.edu>,
Erik Wilde <erik.wilde@dret.net>

Change controller: IETF

Specification document: this specification,
[Section 3](#) "The Deprecation Link Relation Type"

8. 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 [RFC7942]. 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 7942](#), "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".

8.1. Implementing the Deprecation Header

This is a list of implementations that implement the deprecation header field:

Organization: Apollo

- * Description: Deprecation header is returned when deprecated functionality (as declared in the GraphQL schema) is accessed
- * Reference: <https://www.npmjs.com/package/apollo-server-tools>

Organization: Zalando

- * Description: Deprecation header is recommended as the preferred way to communicate API deprecation in Zalando API designs.
- * Reference: <https://opensource.zalando.com/restful-api-guidelines/#deprecation>

Organization: Palantir Technologies

- * Description: Deprecation header is incorporated in code generated by `conjure-java`, a CLI to generate Java POJOs and interfaces from Conjure API definitions
- * Reference: <https://github.com/palantir/conjure-java>

Organization: IETF Internet Draft, Registration Protocols Extensions

- * Description: Deprecation link relation is returned in Registration Data Access Protocol (RDAP) notices to indicate deprecation of jCard in favor of JSContact.
- * Reference: <https://tools.ietf.org/html/draft-loffredo-regext-rdap-jcard-deprecation>

Organization: E-Voyageurs Technologies

- * Description: Deprecation header is incorporated in Hesperides, a configuration management tool providing universal text file templating and properties editing through a REST API or a webapp.
- * Reference: https://github.com/voyages-sncf-technologies/hesperides/blob/master/documentation/lightweight-architecture-decision-records/deprecated_endpoints.md

Organization: Open-Xchange

- * Description: Deprecation header is used in Open-Xchange appsuite-middleware
- * Reference: <https://github.com/open-xchange/appsuite-middleware>

Organization: MediaWiki

- * Description: Core REST API of MediaWiki would use Deprecation header for endpoints that have been deprecated because a new endpoint provides the same or better functionality.
- * Reference: <https://phabricator.wikimedia.org/T232485>

8.2. Implementing the Concept

This is a list of implementations that implement the general concept, but do so using different mechanisms:

Organization: Zapier

- * Description: Zapier uses two custom HTTP headers named "X-API-Deprecation-Date" and "X-API-Deprecation-Info"
- * Reference: <https://zapier.com/engineering/api-geriatrics/>

Organization: IBM

* Description: IBM uses a custom HTTP header named "Deprecated"

* Reference:

https://www.ibm.com/support/knowledgecenter/en/SS42VS_7.3.1/com.ibm.qradar.doc/c_rest_api_getting_started.html

Organization: Ultipro

* Description: Ultipro uses the HTTP "Warning" header as described in [Section 5.5 of \[RFC7234\]](#) with code "299"

* Reference: <https://connect.ultipro.com/api-deprecation>

Organization: Clearbit

* Description: Clearbit uses a custom HTTP header named "X-API-Warn"

* Reference: <https://blog.clearbit.com/dealing-with-deprecation/>

Organization: PayPal

* Description: PayPal uses a custom HTTP header named "PayPal-Deprecated"

* Reference: <https://github.com/paypal/api-standards/blob/master/api-style-guide.md#runtime>

9. Security Considerations

The Deprecation header field SHOULD be treated as a hint, meaning that the resource is indicating (and not guaranteeing with certainty) that it is deprecated. Applications consuming the resource SHOULD check the resource documentation to verify authenticity and accuracy. Resource documentation SHOULD provide additional information about the deprecation including recommendation(s) for replacement.

In cases, where the Deprecation header field value is a date in future, it can lead to information that otherwise might not be available. Therefore, applications consuming the resource SHOULD verify the resource documentation and if possible, consult the resource developer to discuss potential impact due to deprecation and plan for possible transition to recommended resource.

In cases where "Link" header is used to provide more documentation and/or recommendation for replacement, one should assume that the content of the "Link" header field may not be secure, private or integrity-guaranteed, and due caution should be exercised when using it. Applications consuming the resource SHOULD check the referred resource documentation to verify authenticity and accuracy.

The suggested "Link" header fields make extensive use of IRIs and URIs. See [RFC3987] for security considerations relating to IRIs. See [RFC3986] for security considerations relating to URIs. See [RFC7230] for security considerations relating to HTTP headers.

Applications that take advantage of typed links should consider the attack vectors opened by automatically following, trusting, or otherwise using links gathered from the HTTP headers. In particular, Link headers that use the "successor-version", "latest-version" or "alternate" relation types should be treated with due caution. See [RFC5829] for security considerations relating to these link relation types.

10. Examples

The first example shows a deprecation header field without date information:

```
Deprecation: true
```

The second example shows a deprecation header with date information and a link to the successor version:

```
Deprecation: Sun, 11 Nov 2018 23:59:59 GMT
Link: <https://api.example.com/v2/customers>; rel="successor-version"
```

The third example shows a deprecation header field with links for the successor version and for the API's deprecation policy. In addition, it shows the sunset date for the deprecated resource:

```
Deprecation: Sun, 11 Nov 2018 23:59:59 GMT
Sunset: Wed, 11 Nov 2020 23:59:59 GMT
Link: <https://api.example.com/v2/customers>; rel="successor-version",
      <https://developer.example.com/deprecation>; rel="deprecation"
```

11. References

11.1. 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", [BCP 90](#), [RFC 3864](#), DOI 10.17487/RFC3864, September 2004, <<https://www.rfc-editor.org/info/rfc3864>>.
- [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>>.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", [RFC 3987](#), DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](#), DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [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>>.
- [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, <<https://www.rfc-editor.org/info/rfc7234>>.
- [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>>.
- [RFC8288] Nottingham, M., "Web Linking", [RFC 8288](#), DOI 10.17487/RFC8288, October 2017, <<https://www.rfc-editor.org/info/rfc8288>>.

11.2. Informative References

- [RFC5829] Brown, A., Clemm, G., and J. Reschke, Ed., "Link Relation Types for Simple Version Navigation between Web Resources", [RFC 5829](#), DOI 10.17487/RFC5829, April 2010, <<https://www.rfc-editor.org/info/rfc5829>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [BCP 205](#), [RFC 7942](#), DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8594] Wilde, E., "The Sunset HTTP Header Field", [RFC 8594](#), DOI 10.17487/RFC8594, May 2019, <<https://www.rfc-editor.org/info/rfc8594>>.

Appendix A. Acknowledgments

The authors would like to thank Nikhil Kolekar, Mark Nottingham, and Roberto Polli for their contributions.

The authors take all responsibility for errors and omissions.

Authors' Addresses

Sanjay Dalal

Email: sanjay.dalal@cal.berkeley.edu

URI: <https://github.com/sdatspun2>

Erik Wilde

Email: erik.wilde@dret.net

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

