RFC 9745
The Deprecation HTTP Header Field

Abstract

The Deprecation HTTP response header field is used to signal to consumers of a resource (in the sense of URI) that the resource will be or has been deprecated. Additionally, the deprecation link relation can be used to link to a resource that provides additional information about planned or existing deprecation and possibly ways in which client application developers can best manage deprecation.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9745.

Copyright Notice

Table of Contents

1. Introduction

Deprecation of an HTTP resource (Section 3.1 of [HTTP]) communicates information about the lifecycle of a resource. It encourages client applications to migrate away from the resource, discourages applications from forming new dependencies on the resource, and informs applications about the risk of continued dependence upon the resource.

The act of deprecation does not change any behavior of the resource. It informs client applications of the fact that a resource will be or has been deprecated. The Deprecation HTTP response header field can be used to convey this information at runtime and indicates when the deprecation will be in effect.

In addition to the Deprecation header field, the resource provider can use other header fields such as the Link header field [LINK] to convey additional information related to deprecation. This can be information such as where to find documentation related to the deprecation, what can be used as a replacement, and when a deprecated resource becomes non-operational.

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 document uses "Structured Field Values for HTTP" [RFC9651] to specify syntax and parsing of date values.

The term "resource" is to be interpreted as defined in Section 3.1 of [HTTP].

2. The Deprecation HTTP Response Header Field

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

2.1. Syntax

The `Deprecation` response header field describes the deprecation of the resource identified with the response it occurred within (see Section 6.4.2 of [HTTP]). It conveys 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 was deprecated at that date).

`Deprecation` is an Item Structured Header Field; its value **MUST** be a Date as per Section 3.3.7 of [RFC9651].

The following example shows that the resource context was deprecated on Friday, June 30, 2023 at 23:59:59 UTC:

```
Deprecation: @1688169599
```

2.2. Scope

The Deprecation header field applies to the resource identified with the response it occurred within (see Section 6.4.2 of [HTTP]), meaning that it announces the upcoming deprecation of that specific resource. However, there may be scenarios where the scope of the announced deprecation is larger than just the single resource where it appears.

Resources are free to define such an increased scope, and usually this scope will be documented by the resource so that consumers of the resource know about the increased scope and can behave accordingly. When doing so, it is important to take into account that such increased

scoping is invisible for consumers who are unaware of the increased scoping rules. This means that these consumers will not be aware of the increased scope, and they will not interpret deprecation information differently from its standard meaning (i.e., it applies to the resource only).

Using such an increased scope still may make sense, as deprecation information is only a hint anyway. It is optional information that cannot be depended on, and client applications should always be implemented in ways that allow them to function without deprecation information. Increased scope information may help client application developers to glean additional hints from related resources and thus might allow them to implement behavior that enables them to make educated guesses about resources becoming deprecated.

For example, an API might not use Deprecation header fields on all of its resources but only on designated resources such as the API's home document. This means that deprecation information is available, but in order to get it, client application developers have to periodically inspect the home document. In this example, the extended context of the Deprecation header field would be all resources provided by the API, while the visibility of the information would only be on the home document.

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 application developer 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 linked deprecation policy. In particular, the deprecation policy may be available as human-readable documentation or as a machine-readable description.

3.1. Documentation

The purpose of the `Deprecation` header field is to provide a hint about deprecation to the resource consumer. Upon reception of the `Deprecation` header field, the client application developer can look up the resource's documentation in order to find deprecation-related information. The documentation **MAY** provide a guide and timeline for migrating away from the deprecated resource to a new resource(s) that replaces the deprecated resource, if applicable. The resource provider can provide a link to the resource documentation using a `Link` header field with the relation type `deprecation` as shown below:

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

In this example, the linked content provides additional information about deprecation of the resource context. There is no Deprecation header field in the response; thus, the resource is not (yet) deprecated. However, the resource already exposes a link where information describing how deprecation is managed for the resource is available. This may be the documentation explaining under which circumstances and with which policies deprecation might take place. For example, a policy may indicate that deprecation of a resource(s) will always be signaled in the dedicated places at least N days ahead of the planned deprecation date and then the resource(s) would be deprecated on the planned date. Or a policy may indicate that the resource(s) would be deprecated first and then be signaled as deprecated at dedicated places. The documentation, in addition to the deprecation policy, may also provide a migration guide explaining to consumers of the resource how to migrate to a new or alternate resource(s) before the deprecation date. Such policy and documentation would be very useful to consumers of the resource to plan ahead and migrate successfully.

The following example uses the same Link header field but also announces a deprecation date using a Deprecation header field:

```
Deprecation: @1688169599
Link: <https://developer.example.com/deprecation>;
      rel="deprecation"; type="text/html"
```

Given that the deprecation date is in the past, the linked information resource may have been updated to include information about the deprecation, allowing consumers to discover information about the deprecation and how to best manage it.

4. 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 field.

The timestamp given in the `Sunset` header field **MUST NOT** be earlier than the one given in the `Deprecation` header field. If that happens (for example, due to misconfiguration of deployment of the resource or an error), the client application developer **SHOULD** consult the resource developer to get clarification.

The following example shows that the resource in context was deprecated on Friday, June 30, 2023 at 23:59:59 UTC and its sunset date is Sunday, June 30, 2024 at 23:59:59 UTC. Please note that for historical reasons the Sunset HTTP header field uses a different data format for date.

```
Deprecation: @1688169599
Sunset: Sun, 30 Jun 2024 23:59:59 UTC
```

5. Resource Behavior

The act of deprecation does not change any behavior of the resource. The presence of a Deprecation header field in a response is not meant to signal a change in the meaning or function of a resource in the context, allowing consumers to still use the resource in the same way as they did before the resource was declared deprecated.

6. IANA Considerations

6.1. The Deprecation HTTP Response Header Field

The `Deprecation` response header field has been added to the "Hypertext Transfer Protocol (HTTP) Field Name Registry" (Section 16.3.1 of [HTTP]) as follows:

Field Name: Deprecation

Status: permanent

Structured Type: Item

Reference: RFC 9745, Section 2: The Deprecation HTTP Header Field

6.2. The Deprecation Link Relation Type

The `deprecation` link relation type has been added to the "Link Relation Types" registry (Section 4.2 of [LINK]) as follows:

Relation Name: deprecation

Description: Refers to a resource that is documentation (intended for human consumption) about the deprecation of the link's context.

Reference: RFC 9745, Section 3

7. Security Considerations

The Deprecation header field should be treated as a hint, meaning that the resource is indicating (but not guaranteeing with certainty) that it will be or has been deprecated. Deprecated resources function as they would have without sending the Deprecation header field, even though one might consider non-functional details such as making them progressively less efficient with longer response time, for example.

Resource documentation should provide additional information about the deprecation, such as recommendations for replacement. Developers of client applications consuming the resource **SHOULD** always check the referred resource documentation to verify authenticity and accuracy.

In cases where a `Link` header field is used to provide documentation, one should assume (unless served over HTTPS) that the content of the `Link` header field may not be secure, private, or integrity-guaranteed, so due caution should be exercised when using it (see Section 5 of [LINK] for more details). In cases where the Deprecation header field value is in the past, the client application developers **MUST** no longer assume that the behavior of the resource will remain the same as before the deprecation date. In cases where the Deprecation header field value is a date in the future, it can lead to information that otherwise might not be available. Therefore, client application developers consuming the resource **SHOULD**, if possible, consult the resource developer to discuss potential impact due to deprecation and plan for possible transition to a recommended resource(s).

8. Normative References

[HTTP]   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>.

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

[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>.

[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>.

[RFC8594]   Wilde, E., "The Sunset HTTP Header Field", RFC 8594, DOI 10.17487/RFC8594, May 2019, <https://www.rfc-editor.org/info/rfc8594>.

[RFC9651]   Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 9651, DOI 10.17487/RFC9651, September 2024, <https://www.rfc-editor.org/info/rfc9651>.

Acknowledgments

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