

The Sunset HTTP Header
draft-wilde-sunset-header-07

Abstract

This specification defines the Sunset HTTP response header field, which indicates that a URI is likely to become unresponsive at a specified point in the future. It also defines a sunset link relation type that allows linking to resources providing information about an upcoming resource or service sunset.

Note to Readers

This draft should be discussed 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/sunset-header>).

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 April 19, 2019.

Copyright Notice

Copyright (c) 2018 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.	Temporary Resources	3
1.2.	Migration	3
1.3.	Retention	3
1.4.	Deprecation	3
2.	Terminology	4
3.	The Sunset HTTP Response Header	4
4.	Sunset and Caching	5
5.	Sunset Scope	5
6.	The Sunset Link Relation Type	6
7.	IANA Considerations	6
7.1.	The Sunset Response Header	6
7.2.	The Sunset Link Relation Type	7
8.	Implementation Status	7
9.	Security Considerations	8
10.	Example	9
11.	References	9
11.1.	Normative References	9
11.2.	Informative References	10
11.3.	URIs	10
Appendix A.	Acknowledgements	10
	Author's Address	10

[1.](#) Introduction

As a general rule, URIs should be stable and persistent, so that applications can use them as stable and persistent identifiers for resources. However, there are many scenarios where for a variety of reasons, URIs have a limited lifetime. In some of these scenarios, this limited lifetime is known in advance. In this case, it can be useful for clients if resources make this information about their limited lifetime known. This specification defines the Sunset HTTP response header field, which indicates that a URI is likely to become unresponsive at a specified point in the future.

Wilde

Expires April 19, 2019

[Page 2]

This specification also defines a link relation type "sunset" that allows to provide information about a resource's or a service's sunset policy, and/or upcoming sunsets, and/or possible mitigation scenarios for resource/service users. This specification does not place any constraints on the nature of the linked resource, which can be targeted at humans, at machines, or a combination of both.

Possible scenarios for known lifetimes of resources include, but are not limited to the following scenarios.

1.1. Temporary Resources

Some resources may have a limited lifetime by definition. For example, a pending order represented by a resource may already list all the details of the order, but may only exist for a limited time unless it is confirmed and only then becomes permanent. In such a case, the service managing the pending order can make this limited lifetime explicit, allowing clients to understand that the pending order, unless confirmed, will disappear at some point in time.

1.2. Migration

If resources are changing identity because a service migrates them, then this may be known in advance. While it may not yet be appropriate to use HTTP redirect status codes (3xx), it may be interesting for clients to learn about the service's plan to take down the original resource.

1.3. Retention

There are many cases where regulation or legislation require that resources are kept available for a certain amount of time. However, in many cases there also is a requirement for those resources to be permanently deleted after some period of time. Since the deletion of the resource in this scenario is governed by well-defined rules, it could be made explicit for clients interacting with the resource.

1.4. Deprecation

For Web APIs one standard scenario is that an API or specific subsets of an API may get deprecated. If this is planned in advance, then for the time before the actual deprecation is rolled out, the resources that will be affected by the deprecation can make the date of their deprecation known. This allows consumers of the API to be notified of the upcoming deprecation.

In this scenario, the announced sunset date typically affects the whole API or parts of it (i.e., sets of resources), and not just a

single resource. In this case, it makes sense for the API to define rules how an announced sunset on a specific resource (such as the API's home/start resource) implies the sunseting of the whole API or parts of it (i.e., sets of resources), and not just the resource returning the sunset header field. Section [Section 5](#) discusses how the scope of the Sunset header field may change because of how a resource is using it.

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](#)].

3. The Sunset HTTP Response Header

The Sunset HTTP response header field allows a server to communicate the fact that a resource is expected to become unresponsive at a specific point in time. It provides information for clients which they can use to control their usage of the resource.

The Sunset header contains a single timestamp which advertises the point in time when the resource is expected to become unresponsive. The Sunset value is an HTTP-date timestamp, as defined in [Section 7.1.1.1 of \[RFC7231\]](#), and SHOULD be a timestamp in the future.

Timestamps in the past SHOULD be considered to mean the present time, meaning that the resource is expected to become unavailable at any point in time.

Sunset = HTTP-date

For example

Sunset: Sat, 31 Dec 2018 23:59:59 GMT

Clients SHOULD treat Sunset timestamps as hints: It is not guaranteed that the resource will in fact be available until that time, and will not be available after that time. However, since this information is provided by the resource itself, it does have some credibility.

After the Sunset time has arrived, it is likely that interactions with the resource will result in client-side errors (HTTP 4xx status codes), redirect responses (HTTP 3xx status codes), or the client might not be able to interact with the resource at all. The Sunset header does not expose any information about which of those behaviors can be expected.

Clients not interpreting an existing Sunset header field can operate as usual and simply may experience the resource becoming unavailable without getting any notification about it beforehand.

4. Sunset and Caching

It should be noted that the Sunset HTTP response header field serves a different purpose than HTTP caching [[RFC7234](#)]. HTTP caching is concerned with making resource representations (i.e., represented resource state) reusable, so that they can be more efficiently used. This is achieved by using header fields that allow clients and intermediaries to better understand when a resource representation can be reused, or when resource state (and thus the representation) may have changed.

The Sunset header field is not concerned with resource state at all. It only signals that a resource is expected to become unavailable at a specific point in time. There are no assumptions about if, when, or how often a resource may change state in the meantime.

For these reasons, the Sunset header field and HTTP caching should be seen as complementary, and not as overlapping in scope and functionality.

5. Sunset Scope

The Sunset header field applies to the resource that returns it, meaning that it announces the upcoming sunset of that specific resources. However, as discussed in [Section 1.4](#), there may be scenarios where the scope of the announced Sunset information 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. However, 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 will not interpret Sunset information different from its standard meaning (i.e., it applies to the resource only).

Using such an increased scope still may make sense, as Sunset information is only a hint anyway, and thus is optional information that cannot be depended on, and clients should always be implemented in ways that allow them to function without Sunset information. Increased scope information may help clients to glean additional hints from resources (e.g., concluding that an API is being

deprecated because its home/start resource announces a Sunset), and thus might allow them to implement behavior that allows them to make educated guesses about resources becoming unavailable.

6. The Sunset Link Relation Type

The Sunset HTTP header field indicates the upcoming retirement of a resource or a service. In addition, resource may want to make information available that provides additional information about how retirement will be handled for resources or services. This information can be broadly described by the following three topics:

Sunset policy: The policy for which resources and in which way sunsets may occur may be published as part of service's description. Sunsets may only/mostly affect a subset of a service's resources, and may be exposed according to a certain policy (e.g., one week in advance).

Upcoming sunset: There may be additional information about an upcoming sunset, which can be published as a resource that can be consumed by those looking for this additional information.

Sunset mitigation: There may be information about possible mitigation/migration strategies, such as possible ways how resource users can switch to alternative resources/services.

Any information regarding the above issues (and possibly additional ones) can be made available through a URI that then can be linked to using the sunset link relation type. This specification places no constraints on the scope or the type of the linked resource. The scope can be for a resource or for a service. The type is determined by the media type of the linked resource, and can be targeted at humans, at machines, or a combination of both.

7. IANA Considerations

7.1. The Sunset Response Header

The Sunset 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: Sunset

Applicable Protocol: Hypertext Transfer Protocol (HTTP)

Status: Standard

Author/Change controller: IETF

Specification document(s): RFC XXXX

7.2. The Sunset Link Relation Type

The sunset link relation type should be added to the permanent registry of link relation types according to [Section 4.2 of RFC 8288](#) [[RFC8288](#)]:

Relation Name: sunset

Description: Linking to a resource providing information about a resource's or service's retirement policy and/or information.

Reference: RFC XXXX

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 [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".

Name: <https://github.com/hskrasek/guzzle-sunset>

Licensing: MIT

Organization: WeWork

Name: <https://github.com/wework/faraday-sunset>

Licensing: MIT

Description: A Ruby gem adding HTTP client middleware for Sunset to Faraday

Organization: WeWork

Name: <https://github.com/wework/rails-sunset>

Licensing: MIT

Description: Create shortcuts for backend developers to use Sunset (The Rails Way).

Organization: Tyk Technologies

Name: <https://github.com/TykTechnologies/tyk>

Licensing: Mozilla Public License Version 2.0

Description: Configurable HTTP header for API version expiry. (GitHub issue [[1](#)])

9. Security Considerations

The Sunset header field should be treated as a resource hint, meaning that the resource is indicating its potential retirement. The definitive test whether or not the resource in fact is available or not will be to attempt to interact with it. Applications should never treat an advertised Sunset date as a definitive prediction that is going to happen at the specified point in time. The Sunset indication may have been inserted by an intermediary, or the advertised date may get changed or withdrawn by the resource owner.

The main purpose of the Sunset header field is to signal intent, so that applications using resources may get a warning ahead of time and can react accordingly. What an appropriate reaction is (such as switching to a different resource or service), what it will be based on (such as machine-readable formats that allow the switching to be done automatically), and when it will happen (such as ahead of the

advertised date or only when the resource in fact becomes unavailable) is outside the scope of this specification.

10. Example

Assuming that a resource has been created in an archive that for management or compliance reasons only stores resources for two years, and permanently deletes them afterwards, then the Sunset header field can be used to expose this information. If such a resource has been created on November 11, 2016, then the following header field can be included in responses:

Sunset: Fri, 11 Nov 2018 11:11:11 GMT

This allows clients that are aware of the Sunset header field to understand that the resource likely will become unavailable at the specified point in time. Clients can decide to ignore this information, adjust their own behavior accordingly, or alert applications or users about this timestamp.

Even though the Sunset header information is made available by the resource itself, there is no guarantee that the resource indeed will become unavailable, and if so, how the response will look like for requests made after that timestamp. In case of the archive used as an example here, the resource indeed may be permanently deleted, and requests for the URI after the Sunset timestamp may receive a "410 Gone" HTTP response. (This is assuming that the archive keeps track of the URIs that it had previously assigned; if not, the response may be a more generic "404 Not Found".)

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), March 1997.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", [BCP 90](#), [RFC 3864](#), September 2004.
- [RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [RFC 7231](#), June 2014.
- [RFC8288] Nottingham, M., "Web Linking", [RFC 8288](#), DOI 10.17487/RFC8288, October 2017, <<https://www.rfc-editor.org/info/rfc8288>>.

11.2. Informative References

- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", [RFC 6982](#), July 2013.
- [RFC7234] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", [RFC 7234](#), June 2014.

11.3. URIs

- [1] <https://github.com/TykTechnologies/tyk/issues/1626>

Appendix A. Acknowledgements

Thanks for comments and suggestions provided by Phil Sturgeon and Asbjørn Ulsberg.

Author's Address

Erik Wilde
CA Technologies

Email: erik.wilde@dret.net

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

