

## The Sunset HTTP Header Field

### 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.

### Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are candidates for any level of Internet Standard; see [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/rfc8594>.

### 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

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 Field . . . . .	4
4. Sunset and Caching . . . . .	5
5. Sunset Scope . . . . .	6
6. The Sunset Link Relation Type . . . . .	6
7. IANA Considerations . . . . .	7
7.1. The Sunset Response Header Field . . . . .	7
7.2. The Sunset Link Relation Type . . . . .	8
8. Security Considerations . . . . .	8
9. Example . . . . .	9
10. References . . . . .	10
10.1. Normative References . . . . .	10
10.2. Informative References . . . . .	10
Acknowledgements . . . . .	10
Author's Address . . . . .	11

## 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.

This specification also defines a sunset link relation type that allows information to be provided about 1) the sunset policy of a resource or a service, and/or 2) upcoming sunsets, and/or 3) 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 to humans, machines, or 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 shopping order represented by a resource may already list all order details, but it may only exist for a limited time unless it is confirmed and only then becomes an acknowledged shopping order. In such a case, the service managing the pending shopping 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 is also 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. Deprecation often happens in two stages: the first stage being that the API is not the preferred or recommended version anymore and the second stage being that the API or a specific version of the API gets decommissioned.

For the first stage (the API is not the preferred or recommended version anymore), the Sunset header field is not appropriate: at this stage, the API remains operational and can still be used. Other mechanisms can be used for signaling that first stage that might help with more visible deprecation management, but the Sunset header field does not aim to represent that information.

For the second stage (the API or a specific version of the API gets decommissioned), the Sunset header field is appropriate: that is when the API or a version does become unresponsive. From the Sunset header field's point of view, it does not matter that the API may not

have been the preferred or recommended version anymore. The only thing that matters is that it will become unresponsive and that this time can be advertised using the Sunset header field.

In this scenario, the announced sunset date typically affects all of the deprecated API or parts of it (i.e., just deprecated sets of resources), and not just a single resource. In this case, it makes sense for the API to define rules about how an announced sunset on a specific resource (such as the API's home/start resource) implies the sunsetting of the whole API or parts of it (i.e., sets of resources), and not just the resource returning the sunset header field.

[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", "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.

## 3. The Sunset HTTP Response Header Field

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 that they can use to control their usage of the resource.

The Sunset header field contains a single timestamp that 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.

It is safest to consider timestamps in the past mean the present time, meaning that the resource is expected to become unavailable at any 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 field 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 recognizing 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 used more efficiently. 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.

This also means that applications acting as intermediaries, such as search engines or archives that make resources discoverable, should treat Sunset information differently from caching information. These applications may use Sunset information for signaling to users that a resource may become unavailable. But they still have to account for the fact that resource state can change in the meantime and that Sunset information is a hint and, thus, future resource availability may differ from the advertised timestamp.

## 5. Sunset Scope

The Sunset header field applies to the resource that returns it, meaning that it announces the upcoming sunset of that specific resource. 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 they 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; thus, it 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, a 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 they 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 to humans, machines, or both.

If the linked resource does provide machine-readable information, consumers should be careful before acting on this information. Such information may, for example, instruct consumers to use a migration rule so that sunset resources can be accessed at new URIs. However, this kind of information amounts to a possibly large-scale identity migration of resources, so it is crucial that the migration information is authentic and accurate.

## 7. IANA Considerations

### 7.1. The Sunset Response Header Field

The Sunset response header field has been added to the "Permanent Message Header Field Names" registry (see [RFC3864]), taking into account the guidelines given by HTTP/1.1 [RFC7231].

Header Field Name: Sunset

Protocol: http

Status: informational

Author/Change controller: IETF

Reference: [RFC 8594](#)

## 7.2. The Sunset Link Relation Type

The sunset link relation type has been added to the permanent "Link Relation Types" registry according to [Section 4.2 of \[RFC8288\]](#):

Relation Name: sunset

Description: Identifies a resource that provides information about the context's retirement policy.

Reference: [RFC 8594](#)

## 8. Security Considerations

Generally speaking, information about upcoming sunsets can leak information that otherwise might not be available. For example, a resource representing a registration can leak information about the expiration date when it exposes sunset information. For this reason, any use of sunset information where the sunset represents an expiration or allows the calculation of another date (such as calculating a creation date because it is known that resources expire after one year) should be treated in the same way as if this information would be made available directly in the resource's representation.

The Sunset header field SHOULD be treated as a resource hint, meaning that the resource is indicating (and not guaranteeing with certainty) its potential retirement. The definitive test whether or not the resource in fact is available will be to attempt to interact with it. Applications should never treat an advertised Sunset date as a definitive prediction of what 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.

In cases where a sunset policy is linked by using the sunset link relation type, clients SHOULD be careful about taking any actions based on this information. It SHOULD be verified that the information is authentic and accurate. Furthermore, it SHOULD be



tested that this information is only applied to resources that are within the scope of the policy, making sure that sunset policies cannot "hijack" resources by for example providing migration information for them.

## 9. Example

If a resource has been created in an archive that, for management or compliance reasons, stores resources for ten years and permanently deletes them afterwards, 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: Wed, 11 Nov 2026 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 field 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".)

Before the Sunset header field even appears for the first time (it may not appear from the very beginning), it is possible that the resource (or possibly just the "home" resource of the service context) communicates its sunset policy by using the sunset link relation type. If communicated as an HTTP header field, it might look as follows:

```
Link: <http://example.net/sunset>;rel="sunset";type="text/html"
```

In this case, the linked resource provides sunset policy information about the service context. It may be documentation aimed at developers, for example, informing them that the lifetime of a certain class of resources is ten years after creation and that Sunset header fields will be served as soon as the sunset date is

less than some given period of time. It may also inform developers whether the service will respond with 410 or 404 after the sunset time, as discussed above.

## 10. References

### 10.1. 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>>.
- [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>>.
- [RFC8288] Nottingham, M., "Web Linking", [RFC 8288](#), DOI 10.17487/RFC8288, October 2017, <<https://www.rfc-editor.org/info/rfc8288>>.

### 10.2. Informative References

- [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>>.

## Acknowledgements

Thanks for comments and suggestions provided by Ben Campbell, Alissa Cooper, Benjamin Kaduk, Mirja Kuhlewind, Adam Roach, Phil Sturgeon, and Asbjorn Ulsberg.

Author's Address

Erik Wilde

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

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