Network Working Group   E. Wilde
Internet-Draft   February 3, 2016
Intended status: Standards Track
Expires: August 6, 2016

The Sunset HTTP Header
draft-wilde-sunset-header-01

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.

Note to Readers

This draft should be discussed on the apps-discuss mailing list [1].

Online access to all versions and files is available on GitHub [2].

Status of This Memo

Copyright Notice

Table of Contents

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 some point in the future.

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

> 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.

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.

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.

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 [1].

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 [3].

```
Sunset = HTTP-date
```

For example

```
Sunset: Sat, 31 Dec 2016 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 either results in client-side errors (HTTP 4xx status codes), or redirect responses (HTTP 3xx status codes). The Sunset header does not expose any information about which of those two 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 [4]. 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. IANA Considerations

5.1. The Sunset Response Header

The Sunset response header should be added to the permanent registry of message header fields (see [2]), taking into account the guidelines given by HTTP/1.1 [3].

Header Field Name: Sunset

Applicable Protocol: Hypertext Transfer Protocol (HTTP)

Status: Standard

Author/Change controller: IETF

Specification document(s): RFC XXXX

6. Security Considerations

...

7. Example

Assuming that a resource has been created in an archive that for management or compliance reason 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, 2014, then the following header field can be included in responses:

```
Sunset: Fri, 11 Nov 2016 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 this point in time. Clients can decide to ignore this information, adjust their own behavior accordingly, or even alert 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 it has assigned; if not, the response may be a more generic "404 Not Found".)

8. Open Issues

Note to RFC Editor: Please remove this section before publication.

- Human readable explanation: It would be possible to include human-readable information about the reason for the URI's disappearance. This could either be done inline (probably just a simple string), and/or by reference to a URI that will dereference to human-readable information.

- Machine-readable explanation: It would be possible to include machine-readable information about the reason for the URI's disappearance. This could either be done inline (maybe name/value pairs), and/or by reference to a URI that will dereference to machine-readable information.

- New URI: If the resource's new URI is known (for example when a service is migrating in a way that invalidates existing URIs, but has a plan of how these will map to new URIs), then that URI could be advertised in advance (i.e., it's not yet a 3xx, but clients might want to start working with the new URI anyway).

- Timestamp in the past: Should it be allowed to have timestamps in the past? Maybe either disallow them, or specifically allow them and explain what they mean, for example the fact that a resource that now is redirecting can expose information about when it changed from being the primary resource to being the redirection.

9. Change Log

Note to RFC Editor: Please remove this section before publication.

9.1. From -00 to -01

- Fixing Typos.

10. References

10.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[2] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[3] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

10.2. Non-Normative References

[4] Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June 2014.

Author's Address

Erik Wilde

Email: erik.wilde@dret.net
URI: http://dret.net/netdret/