Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 September 2025

Y. Weiss
Shopify

20 March 2025

Delete-Cookie
draft-deletecookie-weiss-http-00

Abstract

This document specifies a Delete-Cookie HTTP header that instructs clients to delete cookies of certain names, without requiring the server to know more details about them.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://yoavweiss.github.io/delete-cookie/draft-deletecookie-weiss-http.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-deletecookie-weiss-http/.

Source for this draft and an issue tracker can be found at https://github.com/yoavweiss/delete-cookie.

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 21 September 2025.

Table of Contents

1. Introduction

Long-operating web sites can often find themselves dealing with "cookie cruft" - cookies that no longer have backend logic that corresponds with them.

Such cookies may have been set at some point in the past with far-reaching expiration dates, and are now causing useless cookie bloat at best, or using up quotas at the expense of relevant cookies at worst

Deleting cookies is possible today by setting their expiry date to one in the past, but that requires one to know the "domain" and "path" parameters with which the cookies were set. That is not something that can be passively observed on the server side by default.

This draft proposes a new Delete-Cookie header which will enable servers to instruct clients to delete cookies of a certain name from their cookie stores.

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 the following terminology from Section 3 of [STRUCTURED-FIELDS] to specify syntax and parsing: Lists, strings.

2. Delete-Cookie

The Delete-Cookie response header is a Structured Field [STRUCTURED-FIELDS] List of strings. Each string represents a cookie [COOKIES] name to be deleted.

A user agent that receives a Delete-Cookie response header MUST process it before any Set-Cookie headers received as part of the same response.

Note: The relevant processing needs to happen before step 15 of the network fetch (https://fetch.spec.whatwg.org/#http-network-fetch) algorithm.

When receiving a Delete-Cookie response header, the user agent MUST:

1. If the response was not delivered over a "secure" connection (as defined by the user agent) [COOKIES], return.

2. Let host be the canonicalized response URL’s host name.

3. Let parsed be the result of parsing the header’s value as a List, as per [STRUCTURED-FIELDS], Section 4.2.1.

4. If parsing fails, return.

5. For each name in parsed:

    1. If name is not a String [STRUCTURED-FIELDS], continue.

        Note: The above ignores any parameters in the header’s value List.

    2. Let matching cookies be the set of cookies from the user agent’s cookie store that meet all the following requirements:

1. The cookie’s name is name.

Note: This ensures that cookies with an empty name can be deleted using the empty string. It also ensures that multiple cookies with the same name (and different paths, domains or other attributes) will all be deleted.

1. Either of the following is true:

    1. The cookie’s host-only-flag is true, and the cookie’s domain is identical to host.

    Note: The ensures that a server can’t delete host-only cookies that aren’t destined to its host.

    1. The cookie’s host-only-flag is false, and the cookie’s domain domain-matches host, as per [COOKIES], Section 5.1.3.

3. For each cookie in matching cookies:

    1. Remove cookie from the user agent’s cookie store.

2.1. Example

MegaCorp servers receive a request from a client with the following header:

Cookie: foo=bar, fizz=baz

The server operators know that cookies named "foo" and "fizz" are no longer meaningful for their service, and are unaware when and how those cookies were set. In order to clear that user’s cookie store, they send the following header:

Delete-Cookie: "foo", "fizz"

That clears the client’s cookie store of "foo" and "fizz" and ensures that these cookies won’t be sent again.

3. Security Considerations

The Delete-Cookie header enables servers to delete cookies from user agents on their own registrable domains. These servers could have already deleted these same cookies by setting cookies with identical name, path and domain with an expiration date of 0. As such the header does not provide servers any new capabilities, beyond the convenience of not having to know the path and domain of a cookie in

order to delete it.

4. IANA Considerations

4.1. Header Field Registration

IANA is asked to update the "Hypertext Transfer Protocol (HTTP) Field Name Registry" registry maintained at <https://www.iana.org/assignments/http-fields/http-fields.xhtml> according to the table below:

Field Name	Status	Reference
Delete-Cookie	permanent	Section 2 of this document

Table 1

5. Normative References

[COOKIES] "Cookies HTTP State Management Mechanism", February 2025, <https://datatracker.ietf.org/doc/draft-ietf-httpbis-rfc6265bis/>.

[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/rfc/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/rfc/rfc8174>.

[STRUCTURED-FIELDS] "Structured Field Values for HTTP", May 2024, <https://datatracker.ietf.org/doc/draft-ietf-httpbis-sfbis/>.

Acknowledgments

Thanks to Anne van Kesteren and Pat Meenan for their feedback on early versions of this proposal.

Author’s Address

Yoav Weiss
Shopify

Email: yoav@yoav.ws