Stream: Internet Engineering Task Force (IETF)
RFC: 9875
Category: Standards Track
Published: October 2025
ISSN: 2070-1721
Author: M. Nottingham
*Cloudflare*

# RFC 9875
# HTTP Cache Groups

## Abstract

This specification introduces a means of describing the relationships between stored responses in HTTP caches, grouping them by associating a stored response with one or more strings.

## 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/rfc9875.

## Copyright Notice

## Table of Contents



# 1. Introduction

HTTP caching [HTTP-CACHING] operates at the granularity of a single resource; the freshness of one stored response does not affect that of others. This granularity can make caching more efficient -- for example, when a page is composed of many assets that have different requirements for caching.

However, there are also cases where the relationship between stored responses could be used to improve cache efficiency.

For example, it is often necessary to invalidate a set of related resources. This might be because a state-changing request has side effects on other resources, or it might be purely for administrative convenience (e.g., "invalidate this part of the site"). Grouping responses together provides a dedicated way to express these relationships, instead of relying on things like URL structure.

In addition to sharing invalidation events, the relationships indicated by grouping can also be used by caches to optimise their operation (e.g., to inform the operation of cache eviction algorithms).

Section 2 introduces a means of describing the relationships between stored responses in HTTP caches, by associating those responses with one or more groups that reflect those relationships. It also describes how caches can use that information to apply invalidation events to members of a group.

Section 3 introduces one new source of such events: an HTTP response header field that allows a state-changing response to trigger a group invalidation.

These mechanisms operate within a single cache, across the stored responses associated with a single origin server (see Section 2.1). They do not address the issues of synchronising state between multiple caches (e.g., in a hierarchy or mesh), nor do they facilitate association of stored responses from disparate origins.

### 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 specification uses the following terminology from [STRUCTURED-FIELDS]: List, String, and Parameter.

## 2. The Cache-Groups Response Header Field

The Cache-Groups response header field is a List of Strings (Sections 3.1 and 3.3.3 of [STRUCTURED-FIELDS]). Each member of the List is a value that identifies a group that the response belongs to. These Strings are opaque -- while they might have some meaning to the server that creates them, the cache does not have any insight into their structure or content (beyond uniquely identifying a group).

```
HTTP/1.1 200 OK
Content-Type: application/javascript
Cache-Control: max-age=3600
Cache-Groups: "scripts"
```

The ordering of members is not significant. Unrecognised Parameters are to be ignored.

Implementations **MUST** support at least 32 groups in a field value, with up to at least 32 characters in each member. Note that generic limitations on HTTP field lengths may constrain the size of this field value in practice.

### 2.1. Identifying Grouped Responses

Two responses stored in the same cache are considered to belong to the same group when all of the following conditions are met:

1. They both contain a Cache-Groups response header field that contains the same String (in any position in the List), when compared character-by-character (case sensitive).
2. They both share the same URI origin (per Section 4.3.1 of [HTTP]).

### 2.2. Cache Behaviour

#### 2.2.1. Invalidation

A cache that invalidates a stored response **MAY** invalidate any stored responses that share groups (per Section 2.1) with that response. Note that further grouped invalidations are not triggered by a grouped invalidation; i.e., this mechanism does not cascade.

Cache extensions can explicitly strengthen the requirement above. For example, a targeted cache control header field [TARGETED] might specify that caches processing it are required to invalidate such responses.

## 3. The Cache-Group-Invalidation Response Header Field

The Cache-Group-Invalidation response header field is a List of Strings (Sections 3.1 and 3.3.3 of [STRUCTURED-FIELDS]). Each member of the List is a value that identifies a group that the response invalidates, per Section 2.2.1.

For example, following a POST request that has side effects on two cache groups, the corresponding response could indicate that stored responses associated with either or both of those groups should be invalidated with:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Group-Invalidation: "eurovision-results", "australia"
```

The Cache-Group-Invalidation header field **MUST** be ignored on responses to requests that have a safe method (e.g., GET; see Section 9.2.1 of [HTTP]).

A cache that receives a Cache-Group-Invalidation header field on a response to an unsafe request **MAY** invalidate any stored responses that share groups (per Section 2.1) with any of the listed groups.

Cache extensions can explicitly strengthen the requirement above. For example, a targeted cache control header field [TARGETED] might specify that caches processing it are required to respect the Cache-Group-Invalidation signal.

The ordering of members is not significant. Unrecognised Parameters are to be ignored.

Implementations **MUST** support at least 32 groups in a field value, with up to at least 32 characters in each member. Note that generic limitations on HTTP field lengths may constrain the size of this field value in practice.

# 4. IANA Considerations

IANA has added the following entries to the "Hypertext Transfer Protocol (HTTP) Field Name Registry":

Field Name: Cache-Groups
Status: permanent
Reference: RFC 9875

Field Name: Cache-Group-Invalidation
Status: permanent
Reference: RFC 9875

# 5. Security Considerations

This mechanism allows resources that share an origin to invalidate each other. Because of this, origins that represent multiple parties (sometimes referred to as "shared hosting") might allow one party to group its resources with those of others or to send signals that have side effects upon them.

Shared hosts that wish to mitigate these risks can control access to the header fields defined in this specification.

# 6. References

## 6.1. 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>.

**[HTTP-CACHING]** Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/info/rfc9111>.

**[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>.

**[STRUCTURED-FIELDS]** 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>.

### 6.2. Informative References

**[TARGETED]** Ludin, S., Nottingham, M., and Y. Wu, "Targeted HTTP Cache Control", RFC 9213, DOI 10.17487/RFC9213, June 2022, <https://www.rfc-editor.org/info/rfc9213>.

## Acknowledgements

Thanks to Stephen Ludin for his review and suggestions.

## Author's Address

**Mark Nottingham**
Cloudflare
Melbourne
Australia
Email: mnot@mnot.net
URI: https://www.mnot.net/