Stream: Internet Engineering Task Force (IETF)
RFC: 10036
Category: Standards Track
Published: August 2026
ISSN: 2070-1721
Authors: K. Oku *Fastly*, T. Pauly *Apple*, M. Thomson *Mozilla*

RFC 10036
Incremental Forwarding of HTTP Messages

Abstract

This document specifies the "Incremental" HTTP header field, which instructs HTTP intermediaries to forward the HTTP message incrementally.

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/rfc10036.

Copyright Notice

Table of Contents

1. Introduction

HTTP [HTTP] permits receivers to begin processing portions of HTTP messages as they arrive, rather than requiring them to wait for the entire HTTP message to be received before acting.

Some applications are specifically designed to take advantage of this capability.

For example, Server-Sent Events [SSE] uses a long-running HTTP response, where the server continually sends notifications as they become available.

In the case of Chunked Oblivious HTTP Messages [CHUNKED-OHTTP], the client opens an HTTP request and incrementally sends application data, while the server can start responding even before the HTTP request is fully complete. In this way, the HTTP request-response pair could create what is, in effect, a bidirectional communication channel.

Applications that rely on incremental delivery of data are fragile when HTTP intermediaries are involved. This is because HTTP intermediaries are not only permitted but are frequently deployed to buffer complete HTTP messages before forwarding them downstream (Section 7.6 of [HTTP]).

If such a buffering HTTP intermediary exists between the client and the server, these applications may fail to function as intended.

In the case of Server-Sent Events, an intermediary that tries to buffer the HTTP response completely before forwarding it could be left waiting indefinitely. A client might never receive any portion of the response.

In the case of requests that involve any bidirectional exchange, an intermediary that tries to buffer entire messages -- either request or response -- prevents any data from being delivered.

To help avoid such behavior, this document specifies the "Incremental" HTTP header field, which requests that HTTP intermediaries begin forwarding the HTTP message downstream before receiving the complete message.

This indication might not be supported by intermediaries. Intermediaries that are unaware of this field will not change their behavior. Intermediaries that support the field might choose instead to reject a request; see Section 4.

2. Conventions and Definitions

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 relies on structured field definitions of Item and Boolean [STRUCTURED-FIELDS].

3. The Incremental Header Field

The Incremental HTTP header field expresses the sender's intent for HTTP intermediaries to start forwarding the message downstream before the entire message is received.

The Incremental header field is defined as a structured field [STRUCTURED-FIELDS] of type Item. Only Boolean values (Section 3.3.6 of [STRUCTURED-FIELDS]) are valid; a recipient ignores the field if it contains any other type.

```
Incremental: ?1
```

A true value ("?1") indicates that the sender requests intermediaries to forward the message incrementally, as described below.

```
Incremental: ?0
```

A false value ("?0") indicates the default behavior defined in [HTTP], where intermediaries might buffer the entire message before forwarding it. However, this explicit signal might give an intermediary greater confidence in choosing to buffer.

The Incremental HTTP header field applies to each HTTP message. Therefore, if both the HTTP request and response need to be forwarded incrementally, the Incremental HTTP header field **MUST** be set for both the HTTP request and the response.

Upon receiving a header section that includes an Incremental header field with a true value, HTTP intermediaries **SHOULD NOT** buffer the entire message before forwarding it. Instead, intermediaries **SHOULD** transmit the header section downstream and continuously forward the bytes of the message content as they arrive. As the Incremental header field indicates only how the message content is to be forwarded, intermediaries can still buffer the entire header and trailer sections of the message before forwarding them downstream.

If an intermediary decides outright to refuse forwarding the message body incrementally, the intermediary **MUST** generate an error response rather than buffering an entire message before forwarding. Typical scenarios under which an intermediary might refuse are discussed in Section 4.

The request to use incremental forwarding also applies to HTTP implementations. Though most HTTP APIs provide the ability to incrementally transfer message content, those that do not for any reason **SHOULD** use the presence of the Incremental header field to reduce or disable buffering.

The Incremental field might not be supported by intermediaries. Intermediaries that are unaware of the field or that do not support the field might buffer messages, even when explicitly requested otherwise. Clients and servers therefore cannot expect all intermediaries to understand and respect a request to deliver messages incrementally. Clients that depend on support for incremental forwarding can rely on prior knowledge or probe for support on individual resources.

The Incremental header field facilitates the establishment of a bidirectional byte channel over HTTP, as its presence in both requests and responses requests that intermediaries forward early responses (Section 7.5 of [HTTP]) and transmit message contents incrementally in both directions. However, when developing bidirectional protocols over HTTP, Extended CONNECT [RFC8441][RFC9220] is generally more consistent with HTTP's architecture.

This document does not define any parameters for the Incremental header field value, but future documents might define parameters. Receivers **MUST** ignore unknown parameters.

4. Security Considerations

When receiving a request or response that asks for incremental forwarding, intermediaries might reject the HTTP request due to security concerns. The following subsections explore typical scenarios under which the intermediaries might reject requests.

Note that rejecting requests based on the value of the Incremental field only occurs when an intermediary understands the field.

4.1. Permanent Rejection

Some intermediaries inspect the content of HTTP messages and forward them only if their content is deemed safe. Any feature that depends on seeing the entirety of the message in this way is incompatible with incremental delivery.

When an intermediary is asked to incrementally forward a message and cannot -- whether that message is a request or a response -- due to security concerns about the message content, the intermediary **SHOULD** respond with a 501 (Not Implemented) error with an incremental_refused Proxy-Status response header field (Section 5).

4.2. Temporary Rejection

To conserve resources required to handle HTTP requests or connections, it is common for intermediaries to impose limits on the maximum number of concurrent HTTP requests that they forward, while buffering requests that exceed this limit.

Such intermediaries could apply a more restrictive concurrency limit to requests marked as incremental to ensure that capacity remains available for non-incremental requests, even when the maximum number of incremental requests is reached. This approach helps balance the processing of different types of requests and maintains service availability across all requests.

When rejecting incremental requests due to reaching the concurrency limit, intermediaries **SHOULD** respond with a 429 (Too Many Requests) error (Section 4 of [EXTRA-STATUS]), accompanied by a connection_limit_reached Proxy-Status response header field (Section 2.3.12 of [PROXY-STATUS]).

4.3. Handling of Small Packets

For performance and efficiency reasons, a small amount of buffering might be used by intermediaries, even for incremental messages. Immediate forwarding might be exploited to cause an intermediary to waste effort on many small packets. Enabling incremental delivery might instead set limits on the number of bytes that are buffered or the length of time that buffers are held before forwarding. Any buffering could adversely affect application latency, even if it improves efficiency. In all cases, intermediaries cannot hold data in buffers indefinitely, so data needs to be forwarded when either the time limit or the byte limit is reached.

5. IANA Considerations

An HTTP field named Incremental has been registered in the "Hypertext Transfer Protocol (HTTP) Field Name Registry" following the procedures in Section 18.4 of [HTTP]. The following values are registered:

Field Name: Incremental

Status: permanent

Structured Type: Item

Reference: This document

Comments: None

An HTTP Proxy Error Type has been registered in the "HTTP Proxy Error Types" registry as shown below:

Name: incremental_refused

Description: The HTTP message contained the Incremental HTTP header field, but the intermediary refused to forward the message incrementally.

Extra Parameters: none

Recommended HTTP Status Code: 501

Response Only Generated By Intermediaries: true

Reference: This document

6. References

6.1. Normative References

[EXTRA-STATUS] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <https://www.rfc-editor.org/info/rfc6585>.

[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>.

[PROXY-STATUS] Nottingham, M. and P. Sikora, "The Proxy-Status HTTP Response Header Field", RFC 9209, DOI 10.17487/RFC9209, June 2022, <https://www.rfc-editor.org/info/rfc9209>.

[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

[CHUNKED-OHTTP] Pauly, T. and M. Thomson, "Chunked Oblivious HTTP Messages", Work in Progress, Internet-Draft, draft-ietf-ohai-chunked-ohttp-08, 18 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ohai-chunked-ohttp-08>.

[RFC8441] McManus, P., "Bootstrapping WebSockets with HTTP/2", RFC 8441, DOI 10.17487/RFC8441, September 2018, <https://www.rfc-editor.org/info/rfc8441>.

[RFC9220] Hamilton, R., "Bootstrapping WebSockets with HTTP/3", RFC 9220, DOI 10.17487/RFC9220, June 2022, <https://www.rfc-editor.org/info/rfc9220>.

[SSE] WHATWG, "HTML - Server-Sent Events", WHATWG Living Standard, <https://html.spec.whatwg.org/multipage/server-sent-events.html>. Commit snapshot: <https://html.spec.whatwg.org/commit-snapshots/6f84b26bd6eb8bd0e0e8df9819e43e901867166b/>

Acknowledgments

The authors would like to thank many members of the IETF HTTP Working Group for their discussions and feedback on this specification. In particular, the authors would like to thank Mark Thomas, Piotr Sikora, Thibault Meunier, Marius Kleidl, Ben Schwartz, Willy Tarreau, Will Hawkins, Mark Nottingham, and Lucas Pardue for close review and suggested changes.

Authors' Addresses

Kazuho Oku
Fastly
Email: kazuhooku@gmail.com

Additional contact information:

奥 一穂
Fastly

Tommy Pauly
Apple
Email: tpauly@apple.com

Martin Thomson
Mozilla
Email: mt@lowentropy.net