

httpbis
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

奥 一穗 (K. Oku)
Fastly
T. Pauly
Apple
M. Thomson
Mozilla
20 October 2025

Incremental HTTP Messages
draft-ietf-httpbis-incremental-02

Abstract

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

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 23 April 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Conventions and Definitions	3
3. The Incremental Header Field	3
4. Security Considerations	4
4.1. Permanent Rejection	5
4.2. Temporary Rejection	5
4.3. Handling of Small Packets	5
5. IANA Considerations	6
6. References	6
6.1. Normative References	6
6.2. Informative References	7
Acknowledgments	7
Authors' Addresses	7

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 bi-directional 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 bi-directional 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 instructs HTTP intermediaries to begin forwarding the HTTP message downstream before receiving the complete message.

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 = sf-boolean

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

A false value ("?0") indicates the default behavior defined in [HTTP], where intermediaries might buffer the entire message before forwarding it.

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 body 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.

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 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.

The Incremental field is advisory. 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 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 instructs intermediaries to forward early responses (Section 7.5 of [HTTP]) and to 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 an incremental request, intermediaries might reject the request due to security concerns. The following subsections explore typical scenarios under which the intermediaries might reject requests.

4.1. Permanent Rejection

Some intermediaries inspect the payload 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, so these intermediaries need to reject requests unless the entire message is received.

When an intermediary rejects an incremental message -- either a request or a response -- due to security concerns with regard to the payload that the message might convey, 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 bytes that are buffered or the 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 is 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 is 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

- [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/rfc/rfc9110>>.
- [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]

Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <<https://www.rfc-editor.org/rfc/rfc8941>>.

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-06, 13 September 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-ohai-chunked-ohttp-06>>.

[EXTRA-STATUS]

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

[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/rfc/rfc9209>>.

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

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

[SSE] WHATWG, "Server-Sent Events", n.d., <<https://html.spec.whatwg.org/multipage/server-sent-events.html>>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Kazuho Oku
Fastly
Email: kazuhooku@gmail.com

Additional contact information:

奥 一 穗
Fastly

Tommy Pauly
Apple
Email: tpauly@apple.com

Martin Thomson
Mozilla
Email: mt@lowentropy.net