HTTP
Internet-Draft
Intended status: Standards Track
Expires: 16 April 2022

M. Nottingham
Fastly
P. Sikora
Google

13 October 2021

The Proxy-Status HTTP Response Header Field
draft-ietf-httpbis-proxy-status-08

Abstract

This document defines the Proxy-Status HTTP field to convey the details of intermediary response handling, including generated errors.

Note to Readers

RFC EDITOR: please remove this section before publication

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/ (https://lists.w3.org/Archives/Public/ietf-http-wg/).

Working Group information can be found at https://httpwg.org/ (https://httpwg.org/); source code and issues list for this draft can be found at https://github.com/httpwg/http-extensions/labels/proxy-status (https://github.com/httpwg/http-extensions/labels/proxy-status).

Table of Contents

1. Introduction

HTTP intermediaries (see Section 3.7 of [HTTP]) -- including both forward proxies and gateways (also known as "reverse proxies") -- have become an increasingly significant part of HTTP deployments. In particular, reverse proxies and Content Delivery Networks (CDNs) form part of the critical infrastructure of many Web sites.

Typically, HTTP intermediaries forward requests towards the origin server (inbound) and then forward their responses back to clients (outbound). However, if an error occurs before a response is obtained from an inbound server, the response is often generated by the intermediary itself.

HTTP accommodates these types of errors with a few status codes; for example, 502 Bad Gateway and 504 Gateway Timeout. However, experience has shown that more information is necessary to aid debugging and communicate what's happened to the client. Additionally, intermediaries sometimes want to convey additional information about their handling of a response, even if they did not generate it.

To enable these uses, Section 2 defines a new HTTP response field to allow intermediaries to convey details of their handling of a response. Section 2.1 enumerates the information that can be added to the field by intermediaries, which can be extended as per Section 2.2. Section 2.3 defines a set of error types for use when a proxy encounters an issue when obtaining a response for the request; these can likewise be extended as per Section 2.4.

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 Structured Fields [STRUCTURED-FIELDS] to specify syntax and parsing, and ABNF [RFC5234] as a shorthand for that syntax. The terms sf-list, sf-item, sf-string, sf-token, sf-integer and key refer to the structured types defined therein.

Note that in this specification, "proxy" is used to indicate both forward and reverse proxies, otherwise known as gateways. "Next hop" indicates the connection in the direction leading to the origin server for the request.

2. The Proxy-Status HTTP Field

The Proxy-Status HTTP response field allows an intermediary to convey additional information about its handling of a response and its associated request. The syntax of this header field conforms to [STRUCTURED-FIELDS].

It is a List ([STRUCTURED-FIELDS], Section 3.1):

```
Proxy-Status   = sf-list
```

Each member of the list represents an intermediary that has handled the response. The first member of the list represents the intermediary closest to the origin server, and the last member of the list represents the intermediary closest to the user agent.

For example:

```
Proxy-Status: revproxy1.example.net, ExampleCDN
```

indicates that this response was handled first by revproxy1.example.net (a reverse proxy adjacent to the origin server) and then ExampleCDN.

Intermediaries determine when it is appropriate to add the Proxy-Status field to a response. Some might decide to append to it to all responses, whereas others might only do so when specifically configured to, or when the request contains a header field that activates a debugging mode.

Each member of the list identifies the intermediary that inserted the value, and MUST have a type of either sf-string or sf-token. Depending on the deployment, this might be a service name (but not a software or hardware product name; e.g., "Example CDN"is appropriate, but "ExampleProxy" is not, because it doesn't identify the deployment), a hostname ("proxy-3.example.com"), an IP address, or a generated string.

Parameters on each member (as per Section 3.1.2 of [STRUCTURED-FIELDS]) convey additional information about that intermediary's handling of the response and its associated request; see Section 2.1. While all of these parameters are OPTIONAL, intermediaries are encouraged to provide as much information as possible (but see Section 4 for security considerations in doing so).

When adding a value to the Proxy-Status field, intermediaries SHOULD preserve the existing members of the field to allow debugging of the entire chain of intermediaries handling the request, unless explicitly configured to remove them (e.g., to prevent internal network details from leaking; see Section 4).

Origin servers MUST NOT generate the Proxy-Status field.

Proxy-Status MAY be sent as a HTTP trailer field. For example, if an intermediary is streaming a response and the inbound connection suddenly terminates, Proxy-Status can only be appended to the trailer section of the outbound message, since the header section has already been sent. However, because it might be silently discarded along the path to the user agent (as is the case for all trailer fields; see Section 6.5 of [HTTP]), Proxy-Status SHOULD NOT be sent as a trailer field unless it is not possible to send it in the header section.

To allow recipients to reconstruct the relative ordering of Proxy-Status members conveyed in trailer fields with those conveyed in header fields, an intermediary MUST NOT send Proxy-Status as a trailer field unless it has also generated a Proxy-Status header field with the same member (although potentially different parameters) in that message.

For example, a proxy identified as 'ThisProxy' that receives a response bearing a header field:

Proxy-Status: SomeOtherProxy

would add its own entry to the header field:

Proxy-Status: SomeOtherProxy, ThisProxy

thus allowing it to append a trailer field:

Proxy-Status: ThisProxy; error=read_timeout

... which would thereby allow a downstream recipient to understand that processing by 'SomeOtherProxy' occurred before 'ThisProxy'.

A client MAY promote the Proxy-Status trailer field into a header field by following these steps:

1. For each member trailer_member of the Proxy-Status trailer field value:

    1. Let header_member be the first (left-most) value of the Proxy-Status header field value, comparing the sf-token or sf-string character-by-character and without consideration of parameters.

    2. If no matching header_member is found, continue processing the next trailer_member.

    3. Replace header_member with trailer_member in its entirety, including any parameters.

2. Remove the Proxy-Status trailer field, if empty.

2.1. Proxy-Status Parameters

This section lists parameters that can be used on the members of the Proxy-Status field. Unrecognised parameters MUST be ignored.

2.1.1. error

The error parameter's value is an sf-token that is a Proxy Error Type. When present, it indicates that the intermediary encountered an issue when obtaining this response.

The presence of some Proxy Error Types indicates that the response was generated by the intermediary itself, rather than being forwarded from the origin server. This is the case when, for example, the origin server can't be contacted, so the proxy has to create its own response.

Other Proxy Error Types can be added to (potentially partial) responses that were generated by the origin server or some other inbound server. For example, if the forward connection abruptly closes, an intermediary might add Proxy-Status with an appropriate error as a trailer field.

Proxy Error Types that are registered with a 'Response only generated by intermediaries' value of 'true' indicate that they can only occur on responses generated by the intermediary. If the value is 'false', the response might be generated by the intermediary or an inbound server.

Section 2.3 lists the Proxy Error Types defined in this document; new ones can be defined using the procedure outlined in Section 2.4.

For example:

```
HTTP/1.1 504 Gateway Timeout
Proxy-Status: ExampleCDN; error=connection_timeout
```

indicates that this 504 response was generated by ExampleCDN, due to a connection timeout when going forward.

Or:

```
HTTP/1.1 429 Too Many Requests
Proxy-Status: r34.example.net; error=http_request_error, ExampleCDN
```

indicates that this 429 Too Many Requests response was generated by r34.example.net, not the CDN or the origin.

When sending the error parameter, the most specific Proxy Error Type SHOULD be sent, provided that it accurately represents the error condition. If an appropriate Proxy Error Type is not defined, there are a number of generic error types (e.g., proxy_internal_error, http_protocol_error) that can be used. If they are not suitable, consider registering a new Proxy Error Type (see Section 2.4).

Each Proxy Error Type has a Recommended HTTP Status Code. When generating a HTTP response containing error, its HTTP status code SHOULD be set to the Recommended HTTP Status Code. However, there may be circumstances (e.g., for backwards compatibility with previous behaviours, a status code has already been sent) when another status code might be used.

Proxy Error Types can also define any number of extra parameters for use with that type. Their use, like all parameters, is optional. As a result, if an extra parameter is used with a Proxy Error Type for which it is not defined, it will be ignored.

2.1.2. next-hop

The next-hop parameter's value is an sf-string or sf-token that identifies the intermediary or origin server selected (and used, if contacted) to obtain this response. It might be a hostname, IP address, or alias.

For example:

Proxy-Status: cdn.example.org; next-hop=backend.example.org:8001

indicates that cdn.example.org used backend.example.org:8001 as the next hop for this request.

2.1.3. next-protocol

The next-protocol parameter's value indicates the ALPN protocol identifier [RFC7301] of the protocol used by the intermediary to connect to the next hop when obtaining this response.

The value MUST be either an sf-token or sf-binary, representing a TLS Application-Layer Protocol Negotiation (ALPN) Protocol ID (see https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#alpn-protocol-ids (https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#alpn-protocol-ids)). If the protocol identifier is able to be expressed as an sf-token using ASCII encoding, that form MUST be used.

For example:

Proxy-Status: "proxy.example.org"; next-protocol=h2

Note that the APLN identifier is being used here to identify the protocol in use; it may or may not have been actually used in the protocol negotiation.

2.1.4. received-status

The received-status parameter's value indicates the HTTP status code that the intermediary received from the next hop server when obtaining this response.

The value MUST be an sf-integer.

For example:

Proxy-Status: ExampleCDN; received-status=200

2.1.5. details

The details parameter's value is an sf-string containing additional information not captured anywhere else. This can include implementation-specific or deployment-specific information.

For example:

```
Proxy-Status: proxy.example.net; error="http_protocol_error";
              details="Malformed response header: space before colon"
```

2.2. Defining New Proxy-Status Parameters

New Proxy-Status Parameters can be defined by registering them in the HTTP Proxy-Status Parameters registry.

Registration requests are reviewed and approved by Expert Review, as per [RFC8126], Section 4.5. A specification document is appreciated, but not required.

The Expert(s) should consider the following factors when evaluating requests:

* Community feedback

* If the value is sufficiently well-defined

* Generic parameters are preferred over vendor-specific, application-specific or deployment-specific values. If a generic value cannot be agreed upon in the community, the parameter's name should be correspondingly specific (e.g., with a prefix that identifies the vendor, application or deployment).

* Parameter names should not conflict with registered extra parameters in the Proxy Error Type Registry.

Registration requests should use the following template:

* Name: [a name for the Proxy-Status Parameter that matches key]

* Description: [a description of the parameter semantics and value]

* Reference: [to a specification defining this parameter; optional]

See the registry at https://iana.org/assignments/http-proxy-status (https://iana.org/assignments/http-proxy-status) for details on where to send registration requests.

2.3. Proxy Error Types

This section lists the Proxy Error Types defined by this document. See Section 2.4 for information about defining new Proxy Error Types.

Note that implementations might not produce all Proxy Error Types. The set of types below is designed to map to existing states in implementations, and so may not be applicable to some.

2.3.1. DNS Timeout

* Name: dns_timeout
* Description: The intermediary encountered a timeout when trying to find an IP address for the next hop hostname.
* Extra Parameters: None.
* Recommended HTTP status code: 504
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.2. DNS Error

* Name: dns_error
* Description: The intermediary encountered a DNS error when trying to find an IP address for the next hop hostname.
* Extra Parameters:
  - rcode: A sf-string conveying the DNS RCODE that indicates the error type. See [RFC8499], Section 3.
  - info-code: A sf-integer conveying the Extended DNS Error Code info-code. See [RFC8914].
* Recommended HTTP status code: 502
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.3. Destination Not Found

* Name: destination_not_found
* Description: The intermediary cannot determine the appropriate next hop to use for this request; for example, it may not be configured. Note that this error is specific to gateways, which typically require specific configuration to identify the "backend" server; forward proxies use in-band information to identify the origin server.
* Extra Parameters: None.
* Recommended HTTP status code: 500
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.4. Destination Unavailable

* Name: destination_unavailable
* Description: The intermediary considers the next hop to be unavailable; e.g., recent attempts to communicate with it may have failed, or a health check may indicate that it is down.
* Extra Parameters: None.
* Recommended HTTP status code: 503
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.5. Destination IP Prohibited

* Name: destination_ip_prohibited
* Description: The intermediary is configured to prohibit connections to the next hop IP address.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: true

* Reference: [this document]

2.3.6. Destination IP Unroutable

* Name: destination_ip_unroutable
* Description: The intermediary cannot find a route to the next hop IP address.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.7. Connection Refused

* Name: connection_refused
* Description: The intermediary’s connection to the next hop was refused.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.8. Connection Terminated

* Name: connection_terminated
* Description: The intermediary’s connection to the next hop was closed before complete response was received.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.9. Connection Timeout

* Name: connection_timeout
* Description: The intermediary's attempt to open a connection to the next hop timed out.
* Extra Parameters: None.
* Recommended HTTP status code: 504
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.10. Connection Read Timeout

* Name: connection_read_timeout
* Description: The intermediary was expecting data on a connection (e.g., part of a response), but did not receive any new data in a configured time limit.
* Extra Parameters: None.
* Recommended HTTP status code: 504
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.11. Connection Write Timeout

* Name: connection_write_timeout
* Description: The intermediary was attempting to write data to a connection, but was not able to (e.g., because its buffers were full).
* Extra Parameters: None.
* Recommended HTTP status code: 504
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.12. Connection Limit Reached

* Name: connection_limit_reached
* Description: The intermediary is configured to limit the number of connections it has to the next hop, and that limit has been passed.
* Extra Parameters: None.
* Recommended HTTP status code: 503
* Response only generated by intermediaries: true
* Reference: [this document]

2.3.13. TLS Protocol Error

* Name: tls_protocol_error
* Description: The intermediary encountered a TLS error when communicating with the next hop, either during handshake or afterwards.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]
* Notes: Not appropriate when a TLS alert is received; see tls_alert_received

2.3.14. TLS Certificate Error

* Name: tls_certificate_error
* Description: The intermediary encountered an error when verifying the certificate presented by the next hop.
* Extra Parameters: None.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: true

* Reference: [this document]

2.3.15. TLS Alert Received

* Name: tls_alert_received
* Description: The intermediary received a TLS alert from the next hop.
* Extra Parameters:
  - alert-id: an sf-integer containing the applicable value from the TLS Alerts registry. See {!RFC8446}}.
  - alert-message: an sf-token or sf-string containing the applicable description string from the TLS Alerts registry. See [RFC8446].
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.16. HTTP Request Error

* Name: http_request_error
* Description: The intermediary is generating a client (4xx) response on the origin's behalf. Applicable status codes include (but are not limited to) 400, 403, 405, 406, 408, 411, 413, 414, 415, 416, 417, 429.
* Extra Parameters:
  - status-code: an sf-integer containing the generated status code.
  - status-phrase: an sf-string containing the generated status phrase.
* Recommended HTTP status code: The applicable 4xx status code
* Response only generated by intermediaries: true
* Reference: [this document]

* Notes: This type helps distinguish between responses generated by intermediaries from those generated by the origin.

2.3.17. HTTP Request Denied

* Name: http_request_denied

* Description: The intermediary rejected the HTTP request based on its configuration and/or policy settings. The request wasn't forwarded to the next hop.

* Extra Parameters: None.

* Recommended HTTP status code: 403

* Response only generated by intermediaries: true

* Reference: [this document]

2.3.18. HTTP Incomplete Response

* Name: http_response_incomplete

* Description: The intermediary received an incomplete response to the request from the next hop.

* Extra Parameters: None.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: false

* Reference: [this document]

2.3.19. HTTP Response Header Section Too Large

* Name: http_response_header_section_size

* Description: The intermediary received a response to the request whose header section was considered too large.

* Extra Parameters:

  - header-section-size: an sf-integer indicating how large the headers received were. Note that they might not be complete; i.e., the intermediary may have discarded or refused additional data.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: false

* Reference: [this document]

2.3.20. HTTP Response Header Field Line Too Large

* Name: http_response_header_size

* Description: The intermediary received a response to the request containing an individual header field line that was considered too large.

* Extra Parameters:

  - header-name: an sf-string indicating the name of the header field that triggered the error.

  - header-size: an sf-integer indicating the size of the header field that triggered the error.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: false

* Reference: [this document]

2.3.21. HTTP Response Body Too Large

* Name: http_response_body_size

* Description: The intermediary received a response to the request whose body was considered too large.

* Extra Parameters:

  - body-size: an sf-integer indicating how large the body received was. Note that it may not have been complete; i.e., the intermediary may have discarded or refused additional data.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: false

* Reference: [this document]

2.3.22. HTTP Response Trailer Section Too Large

* Name: http_response_trailer_section_size
* Description: The intermediary received a response to the request whose trailer section was considered too large.
* Extra Parameters:
  - trailer-section-size: an sf-integer indicating how large the trailers received were. Note that they might not be complete; i.e., the intermediary may have discarded or refused additional data.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.23. HTTP Response Trailer Field Line Too Large

* Name: http_response_trailer_size
* Description: The intermediary received a response to the request containing an individual trailer field line that was considered too large.
* Extra Parameters:
  - trailer-name: an sf-string indicating the name of the trailer field that triggered the error.
  - trailer-size: an sf-integer indicating the size of the trailer field that triggered the error.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.24. HTTP Response Transfer-Coding Error

* Name: http_response_transfer_coding
* Description: The intermediary encountered an error decoding the transfer-coding of the response.

* Extra Parameters:
  - coding: an sf-token containing the specific coding (from the HTTP Transfer Coding Registry) that caused the error.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.25. HTTP Response Content-Coding Error

* Name: http_response_content_coding
* Description: The intermediary encountered an error decoding the content-coding of the response.
* Extra Parameters:
  - coding: an sf-token containing the specific coding (from the HTTP Content Coding Registry) that caused the error.
* Recommended HTTP status code: 502
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.26. HTTP Response Timeout

* Name: http_response_timeout
* Description: The intermediary reached a configured time limit waiting for the complete response.
* Extra Parameters: None.
* Recommended HTTP status code: 504
* Response only generated by intermediaries: false
* Reference: [this document]

2.3.27. HTTP Upgrade Failed

* Name: http_upgrade_failed

* Description: The HTTP Upgrade between the intermediary and the next hop failed.

* Extra Parameters: None.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: true

* Reference: [this document]

2.3.28. HTTP Protocol Error

* Name: http_protocol_error

* Description: The intermediary encountered a HTTP protocol error when communicating with the next hop. This error should only be used when a more specific one is not defined.

* Extra Parameters: None.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: false

* Reference: [this document]

2.3.29. Proxy Internal Response

* Name: proxy_internal_response

* Description: The intermediary generated the response locally, without attempting to connect to the next hop (e.g. in response to a request to a debug endpoint terminated at the intermediary).

* Extra Parameters: None.

* Recommended HTTP status code: The most appropriate status code for the response

* Response only generated by intermediaries: true

* Reference: [this document]

2.3.30. Proxy Internal Error

* Name: proxy_internal_error

* Description: The intermediary encountered an internal error unrelated to the origin.

* Extra Parameters: None

* Recommended HTTP status code: 500

* Response only generated by intermediaries: true

* Reference: [this document]

2.3.31. Proxy Configuration Error

* Name: proxy_configuration_error

* Description: The intermediary encountered an error regarding its configuration.

* Extra Parameters: None

* Recommended HTTP status code: 500

* Response only generated by intermediaries: true

* Reference: [this document]

2.3.32. Proxy Loop Detected

* Name: proxy_loop_detected

* Description: The intermediary tried to forward the request to itself, or a loop has been detected using different means (e.g. [RFC8586]).

* Extra Parameters: None.

* Recommended HTTP status code: 502

* Response only generated by intermediaries: true

* Reference: [this document]

2.4. Defining New Proxy Error Types

New Proxy Error Types can be defined by registering them in the HTTP Proxy Error Types registry.

Registration requests are reviewed and approved by Expert Review, as per [RFC8126], Section 4.5. A specification document is appreciated, but not required.

The Expert(s) should consider the following factors when evaluating requests:

* Community feedback

* If the value is sufficiently well-defined

* Generic types are preferred over vendor-specific, application-specific or deployment-specific values. If a generic value cannot be agreed upon in the community, the types's name should be correspondingly specific (e.g., with a prefix that identifies the vendor, application or deployment).

* Extra Parameters should not conflict with registered Proxy-Status parameters.

Registration requests should use the following template:

* Name: [a name for the Proxy Error Type that matches sf-token]

* Description: [a description of the conditions that generate the Proxy Error Type]

* Extra Parameters: [zero or more optional parameters, along with their allowable type(s)]

* Recommended HTTP status code: [the appropriate HTTP status code for this entry]

* Response only generated by intermediaries: ['true' or 'false']

* Reference: [to a specification defining this error type; optional]

* Notes: [optional]

If the Proxy Error Type might occur in responses that are not generated by the intermediary -- for example, when an error is detected as the response is streamed from a forward connection, causing a Proxy-Status trailer field to be appended -- the 'Response only generated by intermediaries' should be 'false'. If the Proxy Error Type only occurs in responses that are generated by the intermediary, it should be 'true'.

See the registry at https://iana.org/assignments/http-proxy-status (https://iana.org/assignments/http-proxy-status) for details on where to send registration requests.

3. IANA Considerations

Upon publication, please create the HTTP Proxy-Status Parameters registry and the HTTP Proxy Error Types registry at https://iana.org/assignments/http-proxy-status (https://iana.org/assignments/http-proxy-status) and populate them with the types defined in Section 2.1 and Section 2.3 respectively; see Section 2.2 and Section 2.4 for its associated procedures.

Additionally, please register the following entry in the Hypertext Transfer Protocol (HTTP) Field Name Registry:

* Field name: Proxy-Status
* Status: permanent
* Specification document(s): [this document]
* Comments:

4. Security Considerations

One of the primary security concerns when using Proxy-Status is leaking information that might aid an attacker. For example, information about the intermediary's configuration and back-end topology can be exposed, allowing attackers to directly target back-end services that are not prepared for high traffic volume or malformed inputs. Some information might only be suitable to reveal to authorized parties.

As a result, care needs to be taken when deciding to generate a Proxy-Status field and what information to include in it. Note that intermediaries are not required to generate a Proxy-Status field in any response, and can conditionally generate them based upon request attributes (e.g., authentication tokens, IP address).

Likewise, generation of all parameters is optional, as is generation of the field itself. Also, the field's content is not verified; an intermediary can claim certain actions (e.g., sending a request over an encrypted channel) but fail to actually do that.

5. References

5.1. Normative References

[HTTP] Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-19, 12 September 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-19>.

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

[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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/rfc/rfc8499>.

[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>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/rfc/rfc7301>.

[RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <https://www.rfc-editor.org/rfc/rfc8914>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

5.2. Informative References

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/rfc/rfc5234>.

[RFC8586] Ludin, S., Nottingham, M., and N. Sullivan, "Loop Detection in Content Delivery Networks (CDNs)", RFC 8586, DOI 10.17487/RFC8586, April 2019, <https://www.rfc-editor.org/rfc/rfc8586>.

Authors' Addresses

Mark Nottingham
Fastly
Prahran
Australia

Email: mnot@mnot.net
URI: https://www.mnot.net/

Piotr Sikora
Google

Email: piotrsikora@google.com