

Internet Engineering Task Force (IETF)
Request for Comments: 7230
Obsoletes: 2145, 2616
Updates: 2817, 2818
Category: Standards Track
ISSN: 2070-1721

R. Fielding, Ed.
Adobe
J. Reschke, Ed.
greenbytes
June 2014

Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing

Abstract

The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems. This document provides an overview of HTTP architecture and its associated terminology, defines the "http" and "https" Uniform Resource Identifier (URI) schemes, defines the HTTP/1.1 message syntax and parsing requirements, and describes related security concerns for implementations.

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 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7230>.

Copyright Notice

Copyright (c) 2014 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	5
1.1. Requirements Notation	6
1.2. Syntax Notation	6
2. Architecture	6
2.1. Client/Server Messaging	7
2.2. Implementation Diversity	8
2.3. Intermediaries	9
2.4. Caches	11
2.5. Conformance and Error Handling	12
2.6. Protocol Versioning	13
2.7. Uniform Resource Identifiers	16
2.7.1. http URI Scheme	17
2.7.2. https URI Scheme	18
2.7.3. http and https URI Normalization and Comparison	19
3. Message Format	19
3.1. Start Line	20
3.1.1. Request Line	21
3.1.2. Status Line	22
3.2. Header Fields	22

3.2.1. Field Extensibility	23
3.2.2. Field Order	23
3.2.3. Whitespace	24
3.2.4. Field Parsing	25
3.2.5. Field Limits	26
3.2.6. Field Value Components	27
3.3. Message Body	28
3.3.1. Transfer-Encoding	28
3.3.2. Content-Length	30
3.3.3. Message Body Length	32
3.4. Handling Incomplete Messages	34
3.5. Message Parsing Robustness	34
4. Transfer Codings	35
4.1. Chunked Transfer Coding	36
4.1.1. Chunk Extensions	36
4.1.2. Chunked Trailer Part	37
4.1.3. Decoding Chunked	38
4.2. Compression Codings	38
4.2.1. Compress Coding	38
4.2.2. Deflate Coding	38
4.2.3. Gzip Coding	39
4.3. TE	39
4.4. Trailer	40
5. Message Routing	40
5.1. Identifying a Target Resource	40
5.2. Connecting Inbound	41
5.3. Request Target	41
5.3.1. origin-form	42
5.3.2. absolute-form	42
5.3.3. authority-form	43
5.3.4. asterisk-form	43
5.4. Host	44
5.5. Effective Request URI	45
5.6. Associating a Response to a Request	46
5.7. Message Forwarding	47
5.7.1. Via	47
5.7.2. Transformations	49
6. Connection Management	50
6.1. Connection	51
6.2. Establishment	52
6.3. Persistence	52
6.3.1. Retrying Requests	53
6.3.2. Pipelining	54
6.4. Concurrency	55
6.5. Failures and Timeouts	55
6.6. Tear-down	56
6.7. Upgrade	57
7. ABNF List Extension: #rule	59

8. IANA Considerations	61
8.1. Header Field Registration	61
8.2. URI Scheme Registration	62
8.3. Internet Media Type Registration	62
8.3.1. Internet Media Type message/http	62
8.3.2. Internet Media Type application/http	63
8.4. Transfer Coding Registry	64
8.4.1. Procedure	65
8.4.2. Registration	65
8.5. Content Coding Registration	66
8.6. Upgrade Token Registry	66
8.6.1. Procedure	66
8.6.2. Upgrade Token Registration	67
9. Security Considerations	67
9.1. Establishing Authority	67
9.2. Risks of Intermediaries	68
9.3. Attacks via Protocol Element Length	69
9.4. Response Splitting	69
9.5. Request Smuggling	70
9.6. Message Integrity	70
9.7. Message Confidentiality	71
9.8. Privacy of Server Log Information	71
10. Acknowledgments	72
11. References	74
11.1. Normative References	74
11.2. Informative References	75
Appendix A. HTTP Version History	78
A.1. Changes from HTTP/1.0	78
A.1.1. Multihomed Web Servers	78
A.1.2. Keep-Alive Connections	79
A.1.3. Introduction of Transfer-Encoding	79
A.2. Changes from RFC 2616	80
Appendix B. Collected ABNF	82
Index	85

1. Introduction

The Hypertext Transfer Protocol (HTTP) is a stateless application-level request/response protocol that uses extensible semantics and self-descriptive message payloads for flexible interaction with network-based hypertext information systems. This document is the first in a series of documents that collectively form the HTTP/1.1 specification:

1. "Message Syntax and Routing" (this document)
2. "Semantics and Content" [RFC7231]
3. "Conditional Requests" [RFC7232]
4. "Range Requests" [RFC7233]
5. "Caching" [RFC7234]
6. "Authentication" [RFC7235]

This HTTP/1.1 specification obsoletes RFC 2616 and RFC 2145 (on HTTP versioning). This specification also updates the use of CONNECT to establish a tunnel, previously defined in RFC 2817, and defines the "https" URI scheme that was described informally in RFC 2818.

HTTP is a generic interface protocol for information systems. It is designed to hide the details of how a service is implemented by presenting a uniform interface to clients that is independent of the types of resources provided. Likewise, servers do not need to be aware of each client's purpose: an HTTP request can be considered in isolation rather than being associated with a specific type of client or a predetermined sequence of application steps. The result is a protocol that can be used effectively in many different contexts and for which implementations can evolve independently over time.

HTTP is also designed for use as an intermediation protocol for translating communication to and from non-HTTP information systems. HTTP proxies and gateways can provide access to alternative information services by translating their diverse protocols into a hypertext format that can be viewed and manipulated by clients in the same way as HTTP services.

One consequence of this flexibility is that the protocol cannot be defined in terms of what occurs behind the interface. Instead, we are limited to defining the syntax of communication, the intent of received communication, and the expected behavior of recipients. If the communication is considered in isolation, then successful actions

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.