

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June 1999

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers [47]. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification defines the protocol referred to as "HTTP/1.1", and is an update to RFC 2068 [33].

- 1 Introduction7
- 1.1 Purpose.....7
- 1.2 Requirements8
- 1.3 Terminology8
- 1.4 Overall Operation12
- 2 Notational Conventions and Generic Grammar14
- 2.1 Augmented BNF14
- 2.2 Basic Rules15
- 3 Protocol Parameters17
- 3.1 HTTP Version17
- 3.2 Uniform Resource Identifiers18
- 3.2.1 General Syntax19
- 3.2.2 http URL19
- 3.2.3 URI Comparison20
- 3.3 Date/Time Formats20
- 3.3.1 Full Date20
- 3.3.2 Delta Seconds21
- 3.4 Character Sets21
- 3.4.1 Missing Charset22
- 3.5 Content Codings23
- 3.6 Transfer Codings24
- 3.6.1 Chunked Transfer Coding25
- 3.7 Media Types26
- 3.7.1 Canonicalization and Text Defaults27
- 3.7.2 Multipart Types27
- 3.8 Product Tokens28
- 3.9 Quality Values29
- 3.10 Language Tags29
- 3.11 Entity Tags30
- 3.12 Range Units30
- 4 HTTP Message31
- 4.1 Message Types31
- 4.2 Message Headers31
- 4.3 Message Body32
- 4.4 Message Length33
- 4.5 General Header Fields34
- 5 Request35
- 5.1 Request-Line35
- 5.1.1 Method36
- 5.1.2 Request-URI36
- 5.2 The Resource Identified by a Request38
- 5.3 Request Header Fields38
- 6 Response39
- 6.1 Status-Line39
- 6.1.1 Status Code and Reason Phrase39
- 6.2 Response Header Fields41

Fielding, et al. Standards Track [Page 2]

RFC 2616 HTTP/1.1 June 1999

- 7 Entity42
- 7.1 Entity Header Fields42
- 7.2 Entity Body43
- 7.2.1 Type43
- 7.2.2 Entity Length43
- 8 Connections44
- 8.1 Persistent Connections44
- 8.1.1 Purpose44
- 8.1.2 Overall Operation45

- 8.2 Message Transmission Requirements47
 - 8.2.1 Persistent Connections and Flow Control47
 - 8.2.2 Monitoring Connections for Error Status Messages48
 - 8.2.3 Use of the 100 (Continue) Status48
 - 8.2.4 Client Behavior if Server Prematurely Closes Connection ..50
- 9 Method Definitions51
 - 9.1 Safe and Idempotent Methods51
 - 9.1.1 Safe Methods51
 - 9.1.2 Idempotent Methods51
 - 9.2 OPTIONS52
 - 9.3 GET53
 - 9.4 HEAD54
 - 9.5 POST54
 - 9.6 PUT55
 - 9.7 DELETE56
 - 9.8 TRACE56
 - 9.9 CONNECT57
- 10 Status Code Definitions57
 - 10.1 Informational 1xx57
 - 10.1.1 100 Continue58
 - 10.1.2 101 Switching Protocols58
 - 10.2 Successful 2xx58
 - 10.2.1 200 OK58
 - 10.2.2 201 Created59
 - 10.2.3 202 Accepted59
 - 10.2.4 203 Non-Authoritative Information59
 - 10.2.5 204 No Content60
 - 10.2.6 205 Reset Content60
 - 10.2.7 206 Partial Content60
 - 10.3 Redirection 3xx61
 - 10.3.1 300 Multiple Choices61
 - 10.3.2 301 Moved Permanently62
 - 10.3.3 302 Found62
 - 10.3.4 303 See Other63
 - 10.3.5 304 Not Modified63
 - 10.3.6 305 Use Proxy64
 - 10.3.7 306 (Unused)64

- 10.3.8 307 Temporary Redirect65
- 10.4 Client Error 4xx65
 - 10.4.1 400 Bad Request65
 - 10.4.2 401 Unauthorized66
 - 10.4.3 402 Payment Required66
 - 10.4.4 403 Forbidden66
 - 10.4.5 404 Not Found66
 - 10.4.6 405 Method Not Allowed66
 - 10.4.7 406 Not Acceptable67
 - 10.4.8 407 Proxy Authentication Required67
 - 10.4.9 408 Request Timeout67
 - 10.4.10 409 Conflict67
 - 10.4.11 410 Gone68
 - 10.4.12 411 Length Required68
 - 10.4.13 412 Precondition Failed68
 - 10.4.14 413 Request Entity Too Large69
 - 10.4.15 414 Request-URI Too Long69
 - 10.4.16 415 Unsupported Media Type69



- 10.5 Server Error 5xx70
- 10.5.1 500 Internal Server Error70
- 10.5.2 501 Not Implemented70
- 10.5.3 502 Bad Gateway70
- 10.5.4 503 Service Unavailable70
- 10.5.5 504 Gateway Timeout71
- 10.5.6 505 HTTP Version Not Supported71
- 11 Access Authentication71
- 12 Content Negotiation71
- 12.1 Server-driven Negotiation72
- 12.2 Agent-driven Negotiation73
- 12.3 Transparent Negotiation74
- 13 Caching in HTTP74
- 13.1.1 Cache Correctness75
- 13.1.2 Warnings76
- 13.1.3 Cache-control Mechanisms77
- 13.1.4 Explicit User Agent Warnings78
- 13.1.5 Exceptions to the Rules and Warnings78
- 13.1.6 Client-controlled Behavior79
- 13.2 Expiration Model79
- 13.2.1 Server-Specified Expiration79
- 13.2.2 Heuristic Expiration80
- 13.2.3 Age Calculations80
- 13.2.4 Expiration Calculations83
- 13.2.5 Disambiguating Expiration Values84
- 13.2.6 Disambiguating Multiple Responses84
- 13.3 Validation Model85
- 13.3.1 Last-Modified Dates86

- 13.3.2 Entity Tag Cache Validators86
- 13.3.3 Weak and Strong Validators86
- 13.3.4 Rules for When to Use Entity Tags and Last-Modified Dates.....89
- 13.3.5 Non-validating Conditionals90
- 13.4 Response Cacheability91
- 13.5 Constructing Responses From Caches92
- 13.5.1 End-to-end and Hop-by-hop Headers92
- 13.5.2 Non-modifiable Headers92
- 13.5.3 Combining Headers94
- 13.5.4 Combining Byte Ranges95
- 13.6 Caching Negotiated Responses95
- 13.7 Shared and Non-Shared Caches96
- 13.8 Errors or Incomplete Response Cache Behavior97
- 13.9 Side Effects of GET and HEAD97
- 13.10 Invalidation After Updates or Deletions97
- 13.11 Write-Through Mandatory98
- 13.12 Cache Replacement99
- 13.13 History Lists99
- 14 Header Field Definitions100
- 14.1 Accept100
- 14.2 Accept-Charset102
- 14.3 Accept-Encoding102
- 14.4 Accept-Language104
- 14.5 Accept-Ranges105
- 14.6 Age106
- 14.7 Allow106
- 14.8 Authorization107



- 14.9.2 What May be Stored by Caches110
- 14.9.3 Modifications of the Basic Expiration Mechanism111
- 14.9.4 Cache Revalidation and Reload Controls113
- 14.9.5 No-Transform Directive115
- 14.9.6 Cache Control Extensions116
- 14.10 Connection117
- 14.11 Content-Encoding118
- 14.12 Content-Language118
- 14.13 Content-Length119
- 14.14 Content-Location120
- 14.15 Content-MD5121
- 14.16 Content-Range122
- 14.17 Content-Type124
- 14.18 Date124
- 14.18.1 Clockless Origin Server Operation125
- 14.19 ETag126
- 14.20 Expect126
- 14.21 Expires127
- 14.22 From128

Fielding, et al. Standards Track [Page 5]

RFC 2616 HTTP/1.1 June 1999

- 14.23 Host128
- 14.24 If-Match129
- 14.25 If-Modified-Since130
- 14.26 If-None-Match132
- 14.27 If-Range133
- 14.28 If-Unmodified-Since134
- 14.29 Last-Modified134
- 14.30 Location135
- 14.31 Max-Forwards136
- 14.32 Pragma136
- 14.33 Proxy-Authenticate137
- 14.34 Proxy-Authorization137
- 14.35 Range138
- 14.35.1 Byte Ranges138
- 14.35.2 Range Retrieval Requests139
- 14.36 Referer140
- 14.37 Retry-After141
- 14.38 Server141
- 14.39 TE142
- 14.40 Trailer143
- 14.41 Transfer-Encoding.....143
- 14.42 Upgrade144
- 14.43 User-Agent145
- 14.44 Vary145
- 14.45 Via146
- 14.46 Warning148
- 14.47 WWW-Authenticate150
- 15 Security Considerations150
- 15.1 Personal Information.....151
- 15.1.1 Abuse of Server Log Information151
- 15.1.2 Transfer of Sensitive Information151
- 15.1.3 Encoding Sensitive Information in URI's152
- 15.1.4 Privacy Issues Connected to Accept Headers152
- 15.2 Attacks Based On File and Path Names153
- 15.3 DNS Spoofing154
- 15.4 Location Headers and Spoofing154

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.