

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].

## Table of Contents

1	Introduction .....	7
1.1	Purpose.....	7
1.2	Requirements .....	8
1.3	Terminology .....	8
1.4	Overall Operation .....	12
2	Notational Conventions and Generic Grammar .....	14
2.1	Augmented BNF .....	14
2.2	Basic Rules .....	15
3	Protocol Parameters .....	17
3.1	HTTP Version .....	17
3.2	Uniform Resource Identifiers .....	18
3.2.1	General Syntax .....	19
3.2.2	http URL .....	19
3.2.3	URI Comparison .....	20
3.3	Date/Time Formats .....	20
3.3.1	Full Date .....	20
3.3.2	Delta Seconds .....	21
3.4	Character Sets .....	21
3.4.1	Missing Charset .....	22
3.5	Content Codings .....	23
3.6	Transfer Codings .....	24
3.6.1	Chunked Transfer Coding .....	25
3.7	Media Types .....	26
3.7.1	Canonicalization and Text Defaults .....	27
3.7.2	Multipart Types .....	27
3.8	Product Tokens .....	28
3.9	Quality Values .....	29
3.10	Language Tags .....	29
3.11	Entity Tags .....	30
3.12	Range Units .....	30
4	HTTP Message .....	31
4.1	Message Types .....	31
4.2	Message Headers .....	31
4.3	Message Body .....	32
4.4	Message Length .....	33
4.5	General Header Fields .....	34
5	Request .....	35
5.1	Request-Line .....	35
5.1.1	Method .....	36
5.1.2	Request-URI .....	36
5.2	The Resource Identified by a Request .....	38
5.3	Request Header Fields .....	38
6	Response .....	39
6.1	Status-Line .....	39
6.1.1	Status Code and Reason Phrase .....	39
6.2	Response Header Fields .....	41

- 7 Entity .....42
  - 7.1 Entity Header Fields .....42
  - 7.2 Entity Body .....43
    - 7.2.1 Type .....43
    - 7.2.2 Entity Length .....43
- 8 Connections .....44
  - 8.1 Persistent Connections .....44
    - 8.1.1 Purpose .....44
    - 8.1.2 Overall Operation .....45
    - 8.1.3 Proxy Servers .....46
    - 8.1.4 Practical Considerations .....46
  - 8.2 Message Transmission Requirements .....47
    - 8.2.1 Persistent Connections and Flow Control .....47
    - 8.2.2 Monitoring Connections for Error Status Messages .....48
    - 8.2.3 Use of the 100 (Continue) Status .....48
    - 8.2.4 Client Behavior if Server Prematurely Closes Connection ..50
- 9 Method Definitions .....51
  - 9.1 Safe and Idempotent Methods .....51
    - 9.1.1 Safe Methods .....51
    - 9.1.2 Idempotent Methods .....51
  - 9.2 OPTIONS .....52
  - 9.3 GET .....53
  - 9.4 HEAD .....54
  - 9.5 POST .....54
  - 9.6 PUT .....55
  - 9.7 DELETE .....56
  - 9.8 TRACE .....56
  - 9.9 CONNECT .....57
- 10 Status Code Definitions .....57
  - 10.1 Informational 1xx .....57
    - 10.1.1 100 Continue .....58
    - 10.1.2 101 Switching Protocols .....58
  - 10.2 Successful 2xx .....58
    - 10.2.1 200 OK .....58
    - 10.2.2 201 Created .....59
    - 10.2.3 202 Accepted .....59
    - 10.2.4 203 Non-Authoritative Information .....59
    - 10.2.5 204 No Content .....60
    - 10.2.6 205 Reset Content .....60
    - 10.2.7 206 Partial Content .....60
  - 10.3 Redirection 3xx .....61
    - 10.3.1 300 Multiple Choices .....61
    - 10.3.2 301 Moved Permanently .....62
    - 10.3.3 302 Found .....62
    - 10.3.4 303 See Other .....63
    - 10.3.5 304 Not Modified .....63
    - 10.3.6 305 Use Proxy .....64
    - 10.3.7 306 (Unused) .....64

- 10.4.1 400 Bad Request .....65
- 10.4.2 401 Unauthorized .....66
- 10.4.3 402 Payment Required .....66
- 10.4.4 403 Forbidden .....66
- 10.4.5 404 Not Found .....66
- 10.4.6 405 Method Not Allowed .....66
- 10.4.7 406 Not Acceptable .....67
- 10.4.8 407 Proxy Authentication Required .....67
- 10.4.9 408 Request Timeout .....67
- 10.4.10 409 Conflict .....67
- 10.4.11 410 Gone .....68
- 10.4.12 411 Length Required .....68
- 10.4.13 412 Precondition Failed .....68
- 10.4.14 413 Request Entity Too Large .....69
- 10.4.15 414 Request-URI Too Long .....69
- 10.4.16 415 Unsupported Media Type .....69
- 10.4.17 416 Requested Range Not Satisfiable .....69
- 10.4.18 417 Expectation Failed .....70
- 10.5 Server Error 5xx .....70
- 10.5.1 500 Internal Server Error .....70
- 10.5.2 501 Not Implemented .....70
- 10.5.3 502 Bad Gateway .....70
- 10.5.4 503 Service Unavailable .....70
- 10.5.5 504 Gateway Timeout .....71
- 10.5.6 505 HTTP Version Not Supported .....71
- 11 Access Authentication .....71
- 12 Content Negotiation .....71
- 12.1 Server-driven Negotiation .....72
- 12.2 Agent-driven Negotiation .....73
- 12.3 Transparent Negotiation .....74
- 13 Caching in HTTP .....74
- 13.1.1 Cache Correctness .....75
- 13.1.2 Warnings .....76
- 13.1.3 Cache-control Mechanisms .....77
- 13.1.4 Explicit User Agent Warnings .....78
- 13.1.5 Exceptions to the Rules and Warnings .....78
- 13.1.6 Client-controlled Behavior .....79
- 13.2 Expiration Model .....79
- 13.2.1 Server-Specified Expiration .....79
- 13.2.2 Heuristic Expiration .....80
- 13.2.3 Age Calculations .....80
- 13.2.4 Expiration Calculations .....83
- 13.2.5 Disambiguating Expiration Values .....84
- 13.2.6 Disambiguating Multiple Responses .....84
- 13.3 Validation Model .....85
- 13.3.1 Last-Modified Dates .....86

- 13.3.2 Entity Tag Cache Validators .....86
- 13.3.3 Weak and Strong Validators .....86
- 13.3.4 Rules for When to Use Entity Tags and Last-Modified Dates.89

13.5	Constructing Responses From Caches	92
13.5.1	End-to-end and Hop-by-hop Headers	92
13.5.2	Non-modifiable Headers	92
13.5.3	Combining Headers	94
13.5.4	Combining Byte Ranges	95
13.6	Caching Negotiated Responses	95
13.7	Shared and Non-Shared Caches	96
13.8	Errors or Incomplete Response Cache Behavior	97
13.9	Side Effects of GET and HEAD	97
13.10	Invalidation After Updates or Deletions	97
13.11	Write-Through Mandatory	98
13.12	Cache Replacement	99
13.13	History Lists	99
14	Header Field Definitions	100
14.1	Accept	100
14.2	Accept-Charset	102
14.3	Accept-Encoding	102
14.4	Accept-Language	104
14.5	Accept-Ranges	105
14.6	Age	106
14.7	Allow	106
14.8	Authorization	107
14.9	Cache-Control	108
14.9.1	What is Cacheable	109
14.9.2	What May be Stored by Caches	110
14.9.3	Modifications of the Basic Expiration Mechanism	111
14.9.4	Cache Revalidation and Reload Controls	113
14.9.5	No-Transform Directive	115
14.9.6	Cache Control Extensions	116
14.10	Connection	117
14.11	Content-Encoding	118
14.12	Content-Language	118
14.13	Content-Length	119
14.14	Content-Location	120
14.15	Content-MD5	121
14.16	Content-Range	122
14.17	Content-Type	124
14.18	Date	124
14.18.1	Clockless Origin Server Operation	125
14.19	ETag	126
14.20	Expect	126
14.21	Expires	127
14.22	From	128

Fielding, et al.

Standards Track

[Page 5]

RFC 2616

HTTP/1.1

June 1999

14.23	Host	128
14.24	If-Match	129
14.25	If-Modified-Since	130
14.26	If-None-Match	132
14.27	If-Range	133
14.28	If-Unmodified-Since	134

# 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.