

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

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. C. 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

HYPERTEXT TRANSFER PROTOCOL -- HTTP/1.1.....	1
Status of this Memo	1
Copyright Notice.....	1
Abstract	1
Table of Contents.....	2
1 Introduction	7
1.1 Purpose	7
1.2 Requirements	7
1.3 Terminology	8
1.4 Overall Operation	10
2 Notational Conventions and Generic Grammar	11
2.1 Augmented BNF	11
2.2 Basic Rules	12
3 Protocol Parameters	13
3.1 HTTP Version	13
3.2 Uniform Resource Identifiers.....	14
3.2.1 General Syntax.....	14
3.2.2 http URL	14
3.2.3 URI Comparison.....	15
3.3 Date/Time Formats	15
3.3.1 Full Date	15
3.3.2 Delta Seconds	16
3.4 Character Sets	16
3.4.1 Missing Charset	16
3.5 Content Codings	16
3.6 Transfer Codings	17
3.6.1 Chunked Transfer Coding.....	18
3.7 Media Types	18
3.7.1 Canonicalization and Text Defaults	19
3.7.2 Multipart Types.....	19
3.8 Product Tokens.....	20
3.9 Quality Values	20
3.10 Language Tags.....	20
3.11 Entity Tags.....	20
3.12 Range Units	21
4 HTTP Message.....	21
4.1 Message Types.....	21
4.2 Message Headers	21
4.3 Message Body.....	22
4.4 Message Length	23
4.5 General Header Fields	23

5	Request	24
5.1	Request-Line	24
5.1.1	Method	24
5.1.2	Request-URI	24
5.2	The Resource Identified by a Request	25
5.3	Request Header Fields	26
6	Response	26
6.1	Status-Line	26
6.1.1	Status Code and Reason Phrase	26
6.2	Response Header Fields	28
7	Entity	28
7.1	Entity Header Fields	28
7.2	Entity Body	29
7.2.1	Type	29
7.2.2	Entity Length	29
8	Connections	29
8.1	Persistent Connections	29
8.1.1	Purpose	29
8.1.2	Overall Operation	30
8.1.3	Proxy Servers	31
8.1.4	Practical Considerations	31
8.2	Message Transmission Requirements	31
8.2.1	Persistent Connections and Flow Control	31
8.2.2	Monitoring Connections for Error Status Messages	31
8.2.3	Use of the 100 (Continue) Status	32
8.2.4	Client Behavior if Server Prematurely Closes Connection	33
9	Method Definitions	33
9.1	Safe and Idempotent Methods	33
9.1.1	Safe Methods	33
9.1.2	Idempotent Methods	34
9.2	OPTIONS	34
9.3	GET	35
9.4	HEAD	35
9.5	POST	35
9.6	PUT	36
9.7	DELETE	36
9.8	TRACE	37
9.9	CONNECT	37
10	Status Code Definitions	37
10.1	Informational 1xx	37
10.1.1	100 Continue	37
10.1.2	101 Switching Protocols	38
10.2	Successful 2xx	38
10.2.1	200 OK	38
10.2.2	201 Created	38
10.2.3	202 Accepted	38
10.2.4	203 Non-Authoritative Information	39
10.2.5	204 No Content	39
10.2.6	205 Reset Content	39
10.2.7	206 Partial Content	39

10.3	Redirection 3xx.....	40
10.3.1	300 Multiple Choices.....	40
10.3.2	301 Moved Permanently	40
10.3.3	302 Found.....	40
10.3.4	303 See Other	41
10.3.5	304 Not Modified	41
10.3.6	305 Use Proxy.....	41
10.3.7	306 (Unused)	41
10.3.8	307 Temporary Redirect	42
10.4	Client Error 4xx	42
10.4.1	400 Bad Request.....	42
10.4.2	401 Unauthorized.....	42
10.4.3	402 Payment Required.....	42
10.4.4	403 Forbidden.....	42
10.4.5	404 Not Found	43
10.4.6	405 Method Not Allowed	43
10.4.7	406 Not Acceptable	43
10.4.8	407 Proxy Authentication Required.....	43
10.4.9	408 Request Timeout	43
10.4.10	409 Conflict	43
10.4.11	410 Gone	44
10.4.12	411 Length Required	44
10.4.13	412 Precondition Failed.....	44
10.4.14	413 Request Entity Too Large	44
10.4.15	414 Request-URI Too Long	44
10.4.16	415 Unsupported Media Type	44
10.4.17	416 Requested Range Not Satisfiable.....	44
10.4.18	417 Expectation Failed	45
10.5	Server Error 5xx	45
10.5.1	500 Internal Server Error	45
10.5.2	501 Not Implemented	45
10.5.3	502 Bad Gateway.....	45
10.5.4	503 Service Unavailable	45
10.5.5	504 Gateway Timeout.....	45
10.5.6	505 HTTP Version Not Supported	45
11	Access Authentication.....	46
12	Content Negotiation.....	46
12.1	Server-driven Negotiation.....	46
12.2	Agent-driven Negotiation	47
12.3	Transparent Negotiation	47
13	Caching in HTTP	47
13.1.1	Cache Correctness.....	48
13.1.2	Warnings.....	49
13.1.3	Cache-control Mechanisms.....	49
13.1.4	Explicit User Agent Warnings	49
13.1.5	Exceptions to the Rules and Warnings.....	50
13.1.6	Client-controlled Behavior.....	50
13.2	Expiration Model	50
13.2.1	Server-Specified Expiration.....	50
13.2.2	Heuristic Expiration.....	51
13.2.3	Age Calculations.....	51
13.2.4	Expiration Calculations.....	52

13.2.5	Disambiguating Expiration Values	53
13.2.6	Disambiguating Multiple Responses	53
13.3	Validation Model	53
13.3.1	Last-Modified Dates	54
13.3.2	Entity Tag Cache Validators	54
13.3.3	Weak and Strong Validators	54
13.3.4	Rules for When to Use Entity Tags and Last-Modified Dates	56
13.3.5	Non-validating Conditionals	57
13.4	Response Cacheability	57
13.5	Constructing Responses From Caches	57
13.5.1	End-to-end and Hop-by-hop Headers	58
13.5.2	Non-modifiable Headers	58
13.5.3	Combining Headers	59
13.5.4	Combining Byte Ranges	59
13.6	Caching Negotiated Responses	60
13.7	Shared and Non-Shared Caches	60
13.8	Errors or Incomplete Response Cache Behavior	61
13.9	Side Effects of GET and HEAD	61
13.10	Invalidation After Updates or Deletions	61
13.11	Write-Through Mandatory	61
13.12	Cache Replacement	62
13.13	History Lists	62
14	Header Field Definitions	62
14.1	Accept	62
14.2	Accept-Charset	64
14.3	Accept-Encoding	64
14.4	Accept-Language	65
14.5	Accept-Ranges	66
14.6	Age	66
14.7	Allow	66
14.8	Authorization	66
14.9	Cache-Control	67
14.9.1	What is Cacheable	68
14.9.2	What May be Stored by Caches	69
14.9.3	Modifications of the Basic Expiration Mechanism	69
14.9.4	Cache Revalidation and Reload Controls	70
14.9.5	No-Transform Directive	72
14.9.6	Cache Control Extensions	72
14.10	Connection	72
14.11	Content-Encoding	73
14.12	Content-Language	73
14.13	Content-Length	74
14.14	Content-Location	74
14.15	Content-MD5	75
14.16	Content-Range	75
14.17	Content-Type	77
14.18	Date	77
14.18.1	Clockless Origin Server Operation	78
14.19	ETag	78
14.20	Expect	78
14.21	Expires	78
14.22	From	79
14.23	Host	79
14.24	If-Match	80

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.