

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

R. Fielding
UC Irvine
J. Gettys
J. Mogul
DEC
H. Frystyk
T. Berners-Lee
MIT/LCS
January 1997

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.

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods. 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".

Table of Contents

1 Introduction.....	7
1.1 Purpose	7
1.2 Requirements	7
1.3 Terminology	8
1.4 Overall Operation	11
2 Notational Conventions and Generic Grammar.....	13
2.1 Augmented BNF	13
2.2 Basic Rules	15
3 Protocol Parameters.....	17
3.1 HTTP Version	17

Fielding, et. al.

Standards Track

[Page 1]

RFC 2068

HTTP/1.1

January 1997

3.2 Uniform Resource Identifiers	18
3.2.1 General Syntax	18
3.2.2 http URL	19
3.2.3 URI Comparison	20
3.3 Date/Time Formats	21
3.3.1 Full Date	21
3.3.2 Delta Seconds	22
3.4 Character Sets	22
3.5 Content Codings	23
3.6 Transfer Codings	24
3.7 Media Types	25
3.7.1 Canonicalization and Text Defaults	26
3.7.2 Multipart Types	27
3.8 Product Tokens	28
3.9 Quality Values	28
3.10 Language Tags	28
3.11 Entity Tags	29
3.12 Range Units	30

4.2	Message Headers	31
4.3	Message Body	32
4.4	Message Length	32
4.5	General Header Fields	34
5	Request	34
5.1	Request-Line	34
5.1.1	Method	35
5.1.2	Request-URI	35
5.2	The Resource Identified by a Request	37
5.3	Request Header Fields	37
6	Response	38
6.1	Status-Line	38
6.1.1	Status Code and Reason Phrase	39
6.2	Response Header Fields	41
7	Entity	41
7.1	Entity Header Fields	41
7.2	Entity Body	42
7.2.1	Type	42
7.2.2	Length	43
8	Connections	43
8.1	Persistent Connections	43
8.1.1	Purpose	43
8.1.2	Overall Operation	44
8.1.3	Proxy Servers	45
8.1.4	Practical Considerations	45
8.2	Message Transmission Requirements	46
9	Method Definitions	48
9.1	Safe and Idempotent Methods	48

9.1.1	Safe Methods	48
9.1.2	Idempotent Methods	49
9.2	OPTIONS	49
9.3	GET	50
9.4	HEAD	50
9.5	POST	51
9.6	PUT	52
9.7	DELETE	53
9.8	TRACE	53
10	Status Code Definitions	53
10.1	Informational 1xx	54
10.1.1	100 Continue	54
10.1.2	101 Switching Protocols	54
10.2	Successful 2xx	54
10.2.1	200 OK	54
10.2.2	201 Created	55
10.2.3	202 Accepted	55
10.2.4	203 Non-Authoritative Information	55
10.2.5	204 No Content	55
10.2.6	205 Reset Content	56
10.2.7	206 Partial Content	56
10.3	Redirection 3xx	56
10.3.1	300 Multiple Choices	57
10.3.2	301 Moved Permanently	57
10.3.3	302 Moved Temporarily	58
10.3.4	303 See Other	58
10.3.5	304 Not Modified	58
10.3.6	305 Use Proxy	59
10.4	Client Error 4xx	59
10.4.1	400 Bad Request	60
10.4.2	401 Unauthorized	60
10.4.3	402 Payment Required	60
10.4.4	403 Forbidden	60
10.4.5	404 Not Found	60
10.4.6	405 Method Not Allowed	61
10.4.7	406 Not Acceptable	61
10.4.8	407 Proxy Authentication Required	61
10.4.9	408 Request Timeout	62
10.4.10	409 Conflict	62
10.4.11	410 Gone	62
10.4.12	411 Length Required	63
10.4.13	412 Precondition Failed	63
10.4.14	413 Request Entity Too Large	63

10.5 Server Error 5xx	64
10.5.1 500 Internal Server Error	64
10.5.2 501 Not Implemented	64

10.5.3 502 Bad Gateway	64
10.5.4 503 Service Unavailable	64
10.5.5 504 Gateway Timeout	64
10.5.6 505 HTTP Version Not Supported	65
11 Access Authentication.....	65
11.1 Basic Authentication Scheme	66
11.2 Digest Authentication Scheme	67
12 Content Negotiation.....	67
12.1 Server-driven Negotiation	68
12.2 Agent-driven Negotiation	69
12.3 Transparent Negotiation	70
13 Caching in HTTP.....	70
13.1.1 Cache Correctness	72
13.1.2 Warnings	73
13.1.3 Cache-control Mechanisms	74
13.1.4 Explicit User Agent Warnings	74
13.1.5 Exceptions to the Rules and Warnings	75
13.1.6 Client-controlled Behavior	75
13.2 Expiration Model	75
13.2.1 Server-Specified Expiration	75
13.2.2 Heuristic Expiration	76
13.2.3 Age Calculations	77
13.2.4 Expiration Calculations	79
13.2.5 Disambiguating Expiration Values	80
13.2.6 Disambiguating Multiple Responses	80
13.3 Validation Model	81
13.3.1 Last-modified Dates	82
13.3.2 Entity Tag Cache Validators	82
13.3.3 Weak and Strong Validators	82
13.3.4 Rules for When to Use Entity Tags and Last- modified Dates.....	85
13.3.5 Non-validating Conditionals	86
13.4 Response Cachability	86
13.5 Constructing Responses From Caches	87
13.5.1 End-to-end and Hop-by-hop Headers	88
13.5.2 Non-modifiable Headers	88
13.5.3 Combining Headers	89
13.5.4 Combining Byte Ranges	90
13.6 Caching Negotiated Responses	90
13.7 Shared and Non-Shared Caches	91
13.8 Errors or Incomplete Response Cache Behavior	91
13.9 Side Effects of GET and HEAD	92
13.10 Invalidation After Updates or Deletions	92
13.11 Write-Through Mandatory	93
13.12 Cache Replacement	93
13.13 History Lists	93
14 Header Field Definitions.....	94
14.1 Accept	95

14.2 Accept-Charset	97
14.3 Accept-Encoding	97
14.4 Accept-Language	98
14.5 Accept-Ranges	99
14.6 Age	99
14.7 Allow	100
14.8 Authorization	100
14.9 Cache-Control	101
14.9.1 What is Cachable	103
14.9.2 What May be Stored by Caches	103
14.9.3 Modifications of the Basic Expiration Mechanism	104
14.9.4 Cache Revalidation and Reload Controls	105

14.10	Connection	109
14.11	Content-Base	109
14.12	Content-Encoding	110
14.13	Content-Language	110
14.14	Content-Length	111
14.15	Content-Location	112
14.16	Content-MD5	113
14.17	Content-Range	114
14.18	Content-Type	116
14.19	Date	116
14.20	ETag	117
14.21	Expires	117
14.22	From	118
14.23	Host	119
14.24	If-Modified-Since	119
14.25	If-Match	121
14.26	If-None-Match	122
14.27	If-Range	123
14.28	If-Unmodified-Since	124
14.29	Last-Modified	124
14.30	Location	125
14.31	Max-Forwards	125
14.32	Pragma	126
14.33	Proxy-Authenticate	127
14.34	Proxy-Authorization	127
14.35	Public	127
14.36	Range	128
14.36.1	Byte Ranges	128
14.36.2	Range Retrieval Requests	130
14.37	Referer	131
14.38	Retry-After	131
14.39	Server	132
14.40	Transfer-Encoding	132
14.41	Upgrade	132

14.42	User-Agent	134
14.43	Vary	134
14.44	Via	135
14.45	Warning	137
14.46	WWW-Authenticate	139
15	Security Considerations	139
15.1	Authentication of Clients	139
15.2	Offering a Choice of Authentication Schemes	140
15.3	Abuse of Server Log Information	141
15.4	Transfer of Sensitive Information	141
15.5	Attacks Based On File and Path Names	142
15.6	Personal Information	143
15.7	Privacy Issues Connected to Accept Headers	143
15.8	DNS Spoofing	144
15.9	Location Headers and Spoofing	144
16	Acknowledgments	144
17	References	146
18	Authors' Addresses	149
19	Appendices	150
19.1	Internet Media Type message/http	150
19.2	Internet Media Type multipart/byteranges	150
19.3	Tolerant Applications	151
19.4	Differences Between HTTP Entities and MIME Entities	152
19.4.1	Conversion to Canonical Form	152
19.4.2	Conversion of Date Formats	153
19.4.3	Introduction of Content-Encoding	153
19.4.4	No Content-Transfer-Encoding	153
19.4.5	HTTP Header Fields in Multipart Body-Parts	153
19.4.6	Introduction of Transfer-Encoding	154
19.4.7	MIME-Version	154
19.5	Changes from HTTP/1.0	154
19.5.1	Changes to Simplify Multi-homed Web Servers and Conserve IP Addresses	155
19.6	Additional Features	156
19.6.1	Additional Request Methods	156
19.6.2	Additional Header Field Definitions	156

1 Introduction

1.1 Purpose

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. The first version of HTTP, referred to as HTTP/0.9, was a simple protocol for raw data transfer across the Internet. HTTP/1.0, as defined by RFC 1945 [6], improved the protocol by allowing messages to be in the format of MIME-like messages, containing meta-information about the data transferred and modifiers on the request/response semantics. However, HTTP/1.0 does not sufficiently take into consideration the effects of hierarchical proxies, caching, the need for persistent connections, and virtual hosts. In addition, the proliferation of incompletely-implemented applications calling themselves "HTTP/1.0" has necessitated a protocol version change in order for two communicating applications to determine each other's true capabilities.

This specification defines the protocol referred to as "HTTP/1.1". This protocol includes more stringent requirements than HTTP/1.0 in order to ensure reliable implementation of its features.

Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation. HTTP allows an open-ended set of methods that indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI) [3][20], as a location (URL) [4] or name (URN) , for indicating the resource to which a method is to be applied. Messages are passed in a format similar to that used by Internet mail as defined by the Multipurpose Internet Mail Extensions (MIME).

HTTP is also used as a generic protocol for communication between user agents and proxies/gateways to other Internet systems, including those supported by the SMTP [16], NNTP [13], FTP [18], Gopher [2], and WAIS [10] protocols. In this way, HTTP allows basic hypermedia access to resources available from diverse applications.

1.2 Requirements

This specification uses the same words as RFC 1123 [8] for defining the significance of each particular requirement. These words are:

MUST

This word or the adjective "required" means that the item is an absolute requirement of the specification.

SHOULD

This word or the adjective "recommended" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

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.