

HTTP Working Group
INTERNET-DRAFT
<draft-ietf-http-v11-spec-03.html>

R. Fielding, UC Irvine
H. Frystyk, MIT/LCS
T. Berners-Lee, MIT/LCS
J. Gettys, DEC
J. C. Mogul, DEC
May 2, 1996

Expires October 2, 1996

Hypertext Transfer Protocol -- HTTP/1.1

1 Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress”.

To learn the current status of any Internet-Draft, please check the “`1id-abstracts.txt`” listing contained in the Internet-Drafts Shadow Directories on `ftp.is.co.za` (Africa), `nic.nordu.net` (Europe), `munnari.oz.au` (Pacific Rim), `ds.internic.net` (US East Coast), or `ftp.isi.edu` (US West Coast).

Distribution of this document is unlimited. Please send comments to the HTTP working group at `<http-wg@cuckoo.hp1.hp.com>`. Discussions of the working group are archived at `<URL:http://www.ics.uci.edu/pub/ietf/http/>`. General discussions about HTTP and the applications which use HTTP should take place on the `<www-talk@w3.org>` mailing list.

NOTE: This specification is for discussion purposes only. It is not claimed to represent the consensus of the HTTP working group, and contains a number of proposals that either have not been discussed or are controversial. The working group is discussing significant changes in many areas, including - support for caching, persistent connections, range retrieval, content negotiation, MIME compatibility, authentication, timing of the PUT operation.

2 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 (commands). 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”.

3 Note to Readers of This Document

We believe this draft to be very close to consensus of the working group in terms of functionality for HTTP/1.1, and the text substantially correct. One final technical change NOT reflected in this draft is to make persistent connections the default behavior for HTTP/1.1; editorial changes to reflect this in the next, and we hope final draft, are being circulated in the working group mailing list.

This draft has undergone extensive reorganization to improve presentation. Let us know if there are remaining problems.

The terminology used in this draft has changed to reduce confusion. While we are converging on a shared set of terminology and definitions, it is possible there will be a final set of terminology adopted in the next draft. Despite any terminology changes that may occur to improve the presentation of the specification, we do not expect to change the name of any header field or parameter name.

There are a very few remaining issues indicated by **Editor's Note: in bold font**.

4 Table of Contents

HYPERTEXT TRANSFER PROTOCOL -- HTTP/1.1	1
1 Status of this Memo	1
2 Abstract	1
3 Note to Readers of This Document	1
4 Table of Contents	3
5 Introduction	9
5.1 Purpose	9
5.2 Requirements	9
5.3 Terminology	10
5.4 Overall Operation	12
5.5 HTTP and MIME	13
6 Notational Conventions and Generic Grammar	13
6.1 Augmented BNF	13
6.2 Basic Rules	15
7 Protocol Parameters	16
7.1 HTTP Version	16
7.2 Uniform Resource Identifiers	16
7.2.1 General Syntax	17
7.2.2 http URL	17
7.2.3 URI Canonicalization	18
7.3 Date/Time Formats	18
7.3.1 Full Date	18
7.3.2 Delta Seconds	19
7.4 Character Sets	20
7.5 Content Codings	20
7.6 Transfer Codings	21
7.7 Media Types	22
7.7.1 Canonicalization and Text Defaults	22
7.7.2 Multipart Types	23
7.8 Product Tokens	23
7.9 Quality Values	24
7.10 Language Tags	24
7.11 Entity Tags	24
7.12 Variant IDs	25
7.13 Variant Sets	25
7.14 Range Protocol Parameters	25
7.14.1 Range Units	25
7.14.2 Byte Ranges	25
7.14.3 Content Ranges	27

8 HTTP Message	27
8.1 Message Types	27
8.2 Message Headers	27
8.3 General Header Fields	28
9 Request	28
9.1 Request-Line	29
9.1.1 Method	29
9.1.2 Request-URI	29
9.2 The Resource Identified by a Request	30
9.3 Request Header Fields	31
10 Response	31
10.1 Status-Line	31
10.1.1 Status Code and Reason Phrase	31
10.2 Response Header Fields	33
11 Entity	34
11.1 Entity Header Fields	34
11.2 Entity Body	34
11.2.1 Type	35
11.2.2 Length	35
12 Status Code Definitions	35
12.1 Informational 1xx	36
12.2 Successful 2xx	36
12.3 Redirection 3xx	37
12.4 Client Error 4xx	39
12.5 Server Error 5xx	41
13 Method Definitions	42
13.1 OPTIONS	42
13.2 GET	43
13.3 HEAD	43
13.4 POST	43
13.4.1 SLUSHY: Entity Transmission Requirements	44
13.5 PUT	45
13.6 DELETE	46
13.7 TRACE	46
14 Access Authentication	47
14.1 Basic Authentication Scheme	47
14.2 Digest Authentication Scheme	48
15 Content Negotiation	48
15.1 Negotiation Facilities Defined in this Specification	49
16 Caching in HTTP	49
16.1 Semantic Transparency	49
16.1.1 Cache Correctness	50
16.1.2 Cache-control Mechanisms	50
16.1.3 Warnings	50
16.1.4 Explicit User Agent Warnings	51
16.1.5 Exceptions to the Rules and Warnings	51

16.1.6 Client-controlled Behavior	51
16.2 Expiration Model	52
16.2.1 Server-Specified Expiration	52
16.2.2 Limitations on the Effect of Expiration Times	52
16.2.3 Heuristic Expiration	52
16.2.4 Age Calculations	53
16.2.5 Expiration Calculations	54
16.2.6 Scope of Expiration	55
16.2.7 Disambiguating Expiration Values	55
16.2.8 Disambiguating Multiple Responses	55
16.3 Validation Model	55
16.3.1 Last-modified Dates	56
16.3.2 Entity Tags	56
16.3.3 Weak and Strong Validators	57
16.3.4 Rules for When to Use Entity Tags and Last-modified Dates	58
16.3.5 Non-validating Conditionals	59
16.4 Constructing Responses From Caches	59
16.4.1 End-to-end and Hop-by-hop Headers	59
16.4.2 Non-modifiable Headers	60
16.4.3 Combining Headers	60
16.4.4 Combining Byte Ranges	61
16.5 Caching and Generic Resources	61
16.5.1 Vary Header Use	61
16.5.2 Alternates Header Use	61
16.5.3 Variant-ID Use	61
16.6 Shared and Non-Shared Caches	62
16.7 Selecting a Cached Response	62
16.7.1 Plain Resources	62
16.7.2 Generic Resources	63
16.8 Errors or Incomplete Response Cache Behavior	63
16.8.1 Caching and Status Codes	63
16.8.2 Handling of Retry-After	63
16.9 Side Effects of GET and HEAD	64
16.10 Invalidation After Updates or Deletions	64
16.11 Write-Through Mandatory	64
16.12 Generic Resources and HTTP/1.0 Proxy Caches	65
16.13 Cache Replacement	65
16.14 Caching of Negative Responses	65
16.15 History Lists	65
17 Persistent Connections	65
17.1 Purpose	65
17.2 Overall Operation	66
17.2.1 Negotiation	66
17.2.2 Pipe-lining	66
17.2.3 Delimiting Entity-Bodies	67
17.3 Proxy Servers	67
17.4 Interaction with Security Protocols	67
17.5 Practical Considerations	67
18 Header Field Definitions	68
18.1 Accept	68
18.2 Accept-Charset	69
18.3 Accept-Encoding	70

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.