

## Hypertext Transfer Protocol -- HTTP/1.0

### Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### IESG Note:

The IESG has concerns about this protocol, and expects this document to be replaced relatively soon by a standards track document.

### Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary 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 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 reflects common usage of the protocol referred to as "HTTP/1.0".

### Table of Contents

1.	Introduction .....	4
1.1	Purpose .....	4
1.2	Terminology .....	4
1.3	Overall Operation .....	6
1.4	HTTP and MIME .....	8
2.	Notational Conventions and Generic Grammar .....	8
2.1	Augmented BNF .....	8
2.2	Basic Rules .....	10
3.	Protocol Parameters .....	12

3.1	HTTP Version .....	12
3.2	Uniform Resource Identifiers .....	14
3.2.1	General Syntax .....	14
3.2.2	http URL .....	15
3.3	Date/Time Formats .....	15
3.4	Character Sets .....	17
3.5	Content Codings .....	18
3.6	Media Types .....	19
3.6.1	Canonicalization and Text Defaults .....	19
3.6.2	Multipart Types .....	20
3.7	Product Tokens .....	20
4.	HTTP Message .....	21
4.1	Message Types .....	21
4.2	Message Headers .....	22
4.3	General Header Fields .....	23
5.	Request .....	23
5.1	Request-Line .....	23
5.1.1	Method .....	24
5.1.2	Request-URI .....	24
5.2	Request Header Fields .....	25
6.	Response .....	25
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 .....	29
7.2	Entity Body .....	29
7.2.1	Type .....	29
7.2.2	Length .....	30
8.	Method Definitions .....	30
8.1	GET .....	31
8.2	HEAD .....	31
8.3	POST .....	31
9.	Status Code Definitions .....	32
9.1	Informational 1xx .....	32
9.2	Successful 2xx .....	32
9.3	Redirection 3xx .....	34
9.4	Client Error 4xx .....	35
9.5	Server Error 5xx .....	37
10.	Header Field Definitions .....	37
10.1	Allow .....	38
10.2	Authorization .....	38
10.3	Content-Encoding .....	39
10.4	Content-Length .....	39
10.5	Content-Type .....	40
10.6	Date .....	40
10.7	Expires .....	41
10.8	From .....	42

10.9	If-Modified-Since	42
10.10	Last-Modified	43
10.11	Location	44
10.12	Pragma	44
10.13	Referer	44
10.14	Server	45
10.15	User-Agent	46
10.16	WWW-Authenticate	46
11.	Access Authentication	47
11.1	Basic Authentication Scheme	48
12.	Security Considerations	49
12.1	Authentication of Clients	49
12.2	Safe Methods	49
12.3	Abuse of Server Log Information	50
12.4	Transfer of Sensitive Information	50
12.5	Attacks Based On File and Path Names	51
13.	Acknowledgments	51
14.	References	52
15.	Authors' Addresses	54
Appendix A.	Internet Media Type message/http	55
Appendix B.	Tolerant Applications	55
Appendix C.	Relationship to MIME	56
C.1	Conversion to Canonical Form	56
C.2	Conversion of Date Formats	57
C.3	Introduction of Content-Encoding	57
C.4	No Content-Transfer-Encoding	57
C.5	HTTP Header Fields in Multipart Body-Parts	57
Appendix D.	Additional Features	57
D.1	Additional Request Methods	58
D.1.1	PUT	58
D.1.2	DELETE	58
D.1.3	LINK	58
D.1.4	UNLINK	58
D.2	Additional Header Field Definitions	58
D.2.1	Accept	58
D.2.2	Accept-Charset	59
D.2.3	Accept-Encoding	59
D.2.4	Accept-Language	59
D.2.5	Content-Language	59
D.2.6	Link	59
D.2.7	MIME-Version	59
D.2.8	Retry-After	60
D.2.9	Title	60
D.2.10	URI	60

## 1. Introduction

### 1.1 Purpose

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification reflects common usage of the protocol referred to as "HTTP/1.0". This specification describes the features that seem to be consistently implemented in most HTTP/1.0 clients and servers. The specification is split into two sections. Those features of HTTP for which implementations are usually consistent are described in the main body of this document. Those features which have few or inconsistent implementations are listed in [Appendix D](#).

Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation. HTTP allows an open-ended set of methods to be used to indicate the purpose of a request. It builds on the discipline of reference provided by the Uniform Resource Identifier (URI) [2], as a location (URL) [4] or name (URN) [16], for indicating the resource on which a method is to be applied. Messages are passed in a format similar to that used by Internet Mail [7] and the Multipurpose Internet Mail Extensions (MIME) [5].

HTTP is also used as a generic protocol for communication between user agents and proxies/gateways to other Internet protocols, such as SMTP [12], NNTP [11], FTP [14], Gopher [1], and WAIS [8], allowing basic hypermedia access to resources available from diverse applications and simplifying the implementation of user agents.

### 1.2 Terminology

This specification uses a number of terms to refer to the roles played by participants in, and objects of, the HTTP communication.

connection

A transport layer virtual circuit established between two application programs for the purpose of communication.

message

The basic unit of HTTP communication, consisting of a structured sequence of octets matching the syntax defined in [Section 4](#) and transmitted via the connection.

**request**

An HTTP request message (as defined in [Section 5](#)).

**response**

An HTTP response message (as defined in [Section 6](#)).

**resource**

A network data object or service which can be identified by a URI ([Section 3.2](#)).

**entity**

A particular representation or rendition of a data resource, or reply from a service resource, that may be enclosed within a request or response message. An entity consists of metainformation in the form of entity headers and content in the form of an entity body.

**client**

An application program that establishes connections for the purpose of sending requests.

**user agent**

The client which initiates a request. These are often browsers, editors, spiders (web-traversing robots), or other end user tools.

**server**

An application program that accepts connections in order to service requests by sending back responses.

**origin server**

The server on which a given resource resides or is to be created.

**proxy**

An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them, with possible translation, on to other servers. A proxy must interpret and, if necessary, rewrite a request message before

# 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.