                 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

RFC 1945   HTTP/1.0   May 1996

Berners-Lee, et al   Informational   [Page 3]

RFC 1945   HTTP/1.0   May 1996

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 too 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

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.

Berners-Lee, et al   Informational   [Page 4]

RFC 1945   HTTP/1.0   May 1996

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

        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

Berners-Lee, et al   Informational   [Page 5]

RFC 1945   HTTP/1.0   May 1996

        forwarding it. Proxies are often used as client-side portals
        through network firewalls and as helper applications for
        handling requests via protocols not implemented by the user
        agent.

    gateway

        A server which acts as an intermediary for some other server.
        Unlike a proxy, a gateway receives requests as if it were the
        origin server for the requested resource; the requesting client
        may not be aware that it is communicating with a gateway.
        Gateways are often used as server-side portals through network
        firewalls and as protocol translators for access to resources
        stored on non-HTTP systems.

    tunnel

        A tunnel is an intermediary program which is acting as a blind
        relay between two connections. Once active, a tunnel is not
        considered a party to the HTTP communication, though the tunnel
        may have been initiated by an HTTP request. The tunnel ceases to
        exist when both ends of the relayed connections are closed.
        Tunnels are used when a portal is necessary and the intermediary
        cannot, or should not, interpret the relayed communication.

    cache

        A program's local store of response messages and the subsystem
        that controls its message storage, retrieval, and deletion. A
        cache stores cachable responses in order to reduce the response
        time and network bandwidth consumption on future, equivalent
        requests. Any client or server may include a cache, though a
        cache cannot be used by a server while it is acting as a tunnel.

    Any given program may be capable of being both a client and a server;
    our use of these terms refers only to the role being performed by the

DOCKET ALARM

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.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase
Smarter legal research.