- HTTP
- HTTP Request

S. Prabhavathi
AP/IT

Hypertext Transport Protocol (HTTP)

- HTTP is based on the request-response communication model:
  - – Client sends a request
  - – Server sends a response
- HTTP is a stateless protocol:
  - – The protocol does not require the server to remember anything about the client between requests.

HTTP

- Normally implemented over a TCP connection (80 is standard port number for HTTP)
- Typical browser-server interaction:
  - User enters Web address in browser
  - Browser uses DNS to locate IP address
  - Browser opens TCP connection to server
  - Browser sends HTTP request over connection
  - Server sends HTTP response to browser over connection
  - Browser displays body of response in the client area of the browser window

HTTP

- The information transmitted using HTTP is often entirely text
- Can use the Internet’s Telnet protocol to simulate browser request and view server response

HTTP

Connect

```
$ telnet www.example.org 80
Trying 192.0.34.166...
Connected to www.example.com
(192.0.34.166).
Escape character is ’^]’.
```

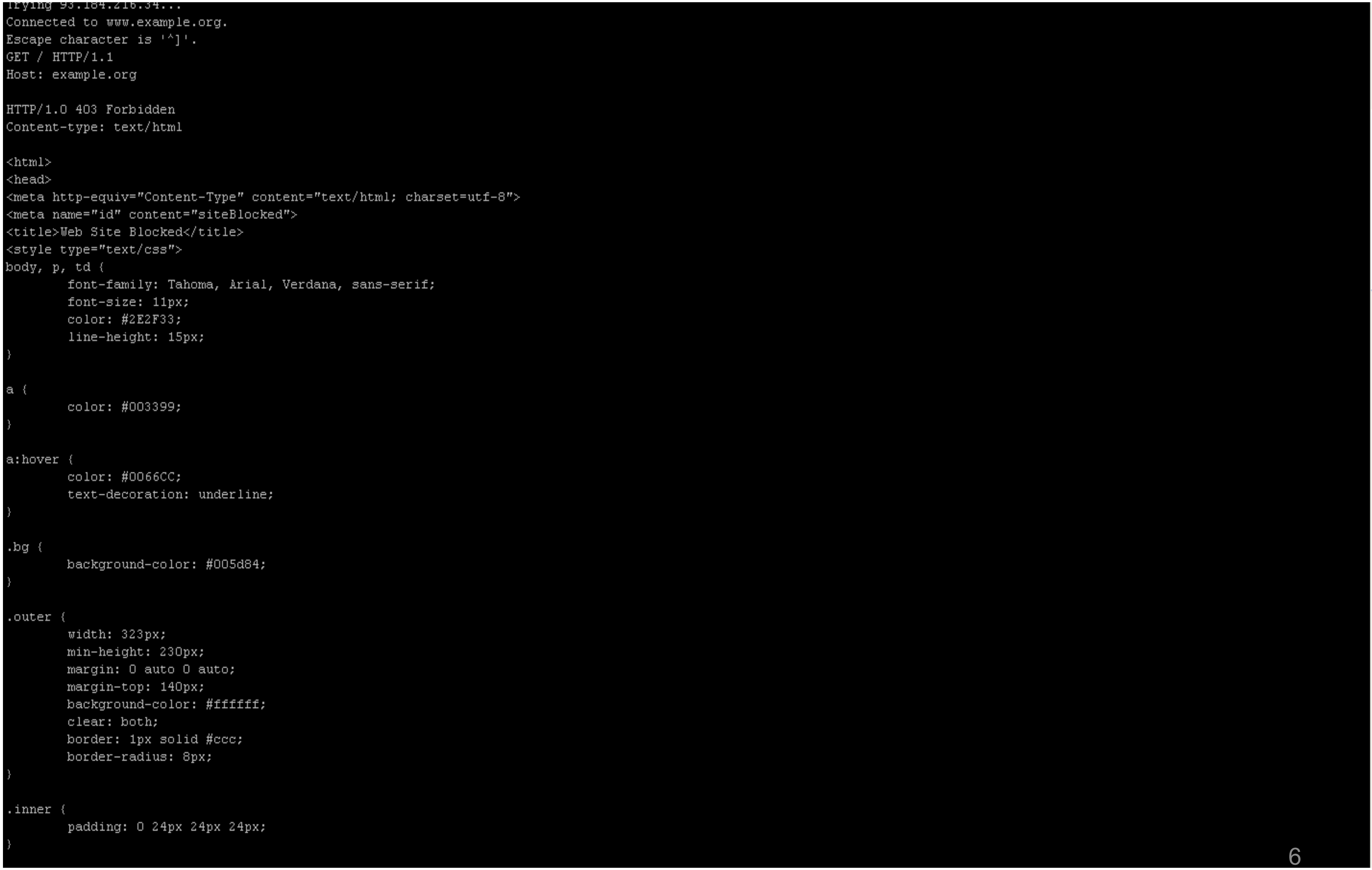
Send Request

```
GET / HTTP/1.1
Host: www.example.org

```

Receive Response

```
HTTP/1.1 200 OK
Date: Thu, 09 Oct 2003 20:30:49 GMT
…
```



```
Trying 93.184.216.34...
Connected to www.example.org.
Escape character is '^]'.
GET / HTTP/1.1
Host: example.org

HTTP/1.0 403 Forbidden
Content-type: text/html

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="id" content="siteBlocked">
<title>Web Site Blocked</title>
<style type="text/css">
body, p, td {
        font-family: Tahoma, Arial, Verdana, sans-serif;
        font-size: 11px;
        color: #2E2F33;
        line-height: 15px;
}

a {
        color: #003399;
}

a:hover {
        color: #0066CC;
        text-decoration: underline;
}

.bg {
        background-color: #005d84;
}

.outer {
        width: 323px;
        min-height: 230px;
        margin: 0 auto 0 auto;
        margin-top: 140px;
        background-color: #ffffff;
        clear: both;
        border: 1px solid #ccc;
        border-radius: 8px;
}

.inner {
        padding: 0 24px 24px 24px;
}
```

HTTP Request Message

Overall structure of the request

- Start line
- Header field(s) (one or more)
- Blank line
- Message body (optional)

HTTP Request Message

Structure of the request

- 1. Start line
- Header field(s) (one or more)
- Blank line
- Message body (optional)

HTTP Request

- Start line
  - Example: `GET / HTTP/1.1`
- Three space-separated parts:
  - HTTP request method
  - Request-URI
  - HTTP version

HTTP Version

- The initial version of HTTP was referred to as HTTP/0.9
- HTTP/1.0, HTTP/1.1

HTTP Request- URI

- Uniform Resource Identifier (URI)

  The concatenation of the string http://, the value of the Host header field (www.example.org, in this example), and the Request-URI (/ in this example) forms a string known as a *Uniform Resource Identifier* (URI).

- A URI is an identifier that is intended to be associated with a particular resource (such as a web page or graphics image) on the World Wide Web.
  - Syntax: *scheme* : *scheme-depend-part*
    - Ex: In http://www.example.com/
      the scheme is http

- URIs using the http scheme are both URIs and URLs.

URI

- URI's are of two types:
  - Uniform Resource Name (URN)
    - Can be used to identify resources with unique names, such as books (which have unique ISBN's)
    - Scheme is `urn`
  - Uniform Resource Locator (URL)
    - Specifies location at which a resource can be found
    - In addition to `http`, some other URL schemes are `https`, `ftp`, `mailto`, and `file`

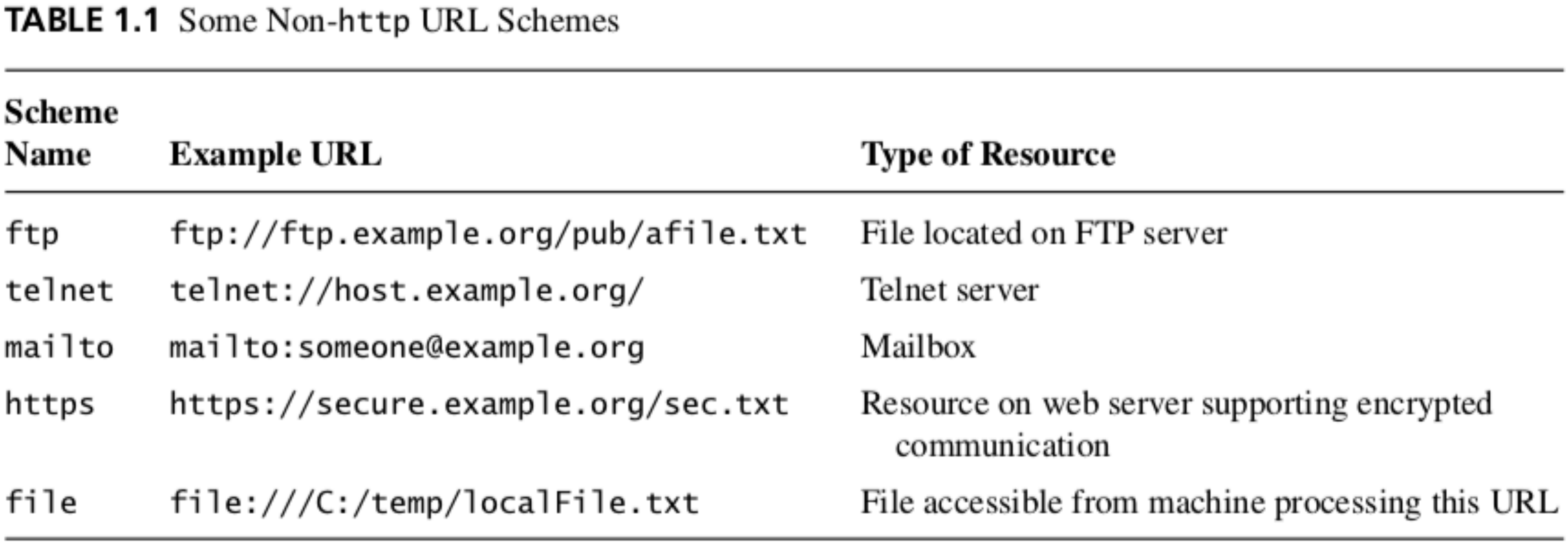


TABLE 1.1 Some Non-http URL Schemes

Scheme Name	Example URL	Type of Resource
`ftp`	`ftp://ftp.example.org/pub/afile.txt`	File located on FTP server
`telnet`	`telnet://host.example.org/`	Telnet server
`mailto`	`mailto:someone@example.org`	Mailbox
`https`	`https://secure.example.org/sec.txt`	Resource on web server supporting encrypted communication
`file`	`file:///C:/temp/localFile.txt`	File accessible from machine processing this URL

HTTP Request

- Start line
  - Example: `GET / HTTP/1.1`
- Three space-separated parts:
  - **HTTP request method**
  - Request-URI
  - HTTP version

HTTP Request Method

- GET method used when you type a URL into the Location bar of your browser.
- GET is used by default when you click on a link in a document
- POST method is typically used to send information collected from a form displayed within a browser, back to the web server.

Request Method

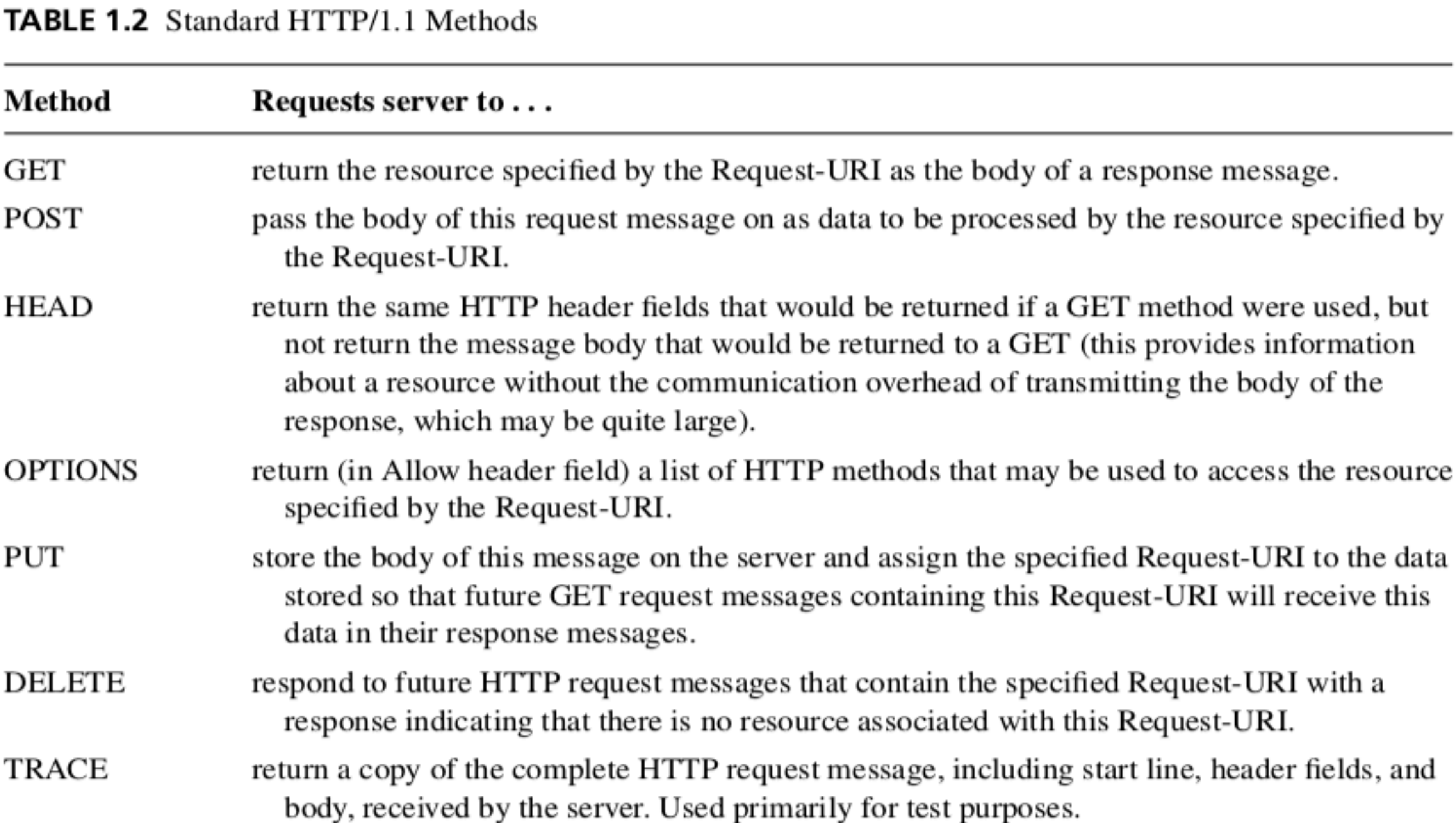


TABLE 1.2 Standard HTTP/1.1 Methods

Method	Requests server to . . .
GET	return the resource specified by the Request-URI as the body of a response message.
POST	pass the body of this request message on as data to be processed by the resource specified by the Request-URI.
HEAD	return the same HTTP header fields that would be returned if a GET method were used, but not return the message body that would be returned to a GET (this provides information about a resource without the communication overhead of transmitting the body of the response, which may be quite large).
OPTIONS	return (in Allow header field) a list of HTTP methods that may be used to access the resource specified by the Request-URI.
PUT	store the body of this message on the server and assign the specified Request-URI to the data stored so that future GET request messages containing this Request-URI will receive this data in their response messages.
DELETE	respond to future HTTP request messages that contain the specified Request-URI with a response indicating that there is no resource associated with this Request-URI.
TRACE	return a copy of the complete HTTP request message, including start line, header fields, and body, received by the server. Used primarily for test purposes.

HTTP Request

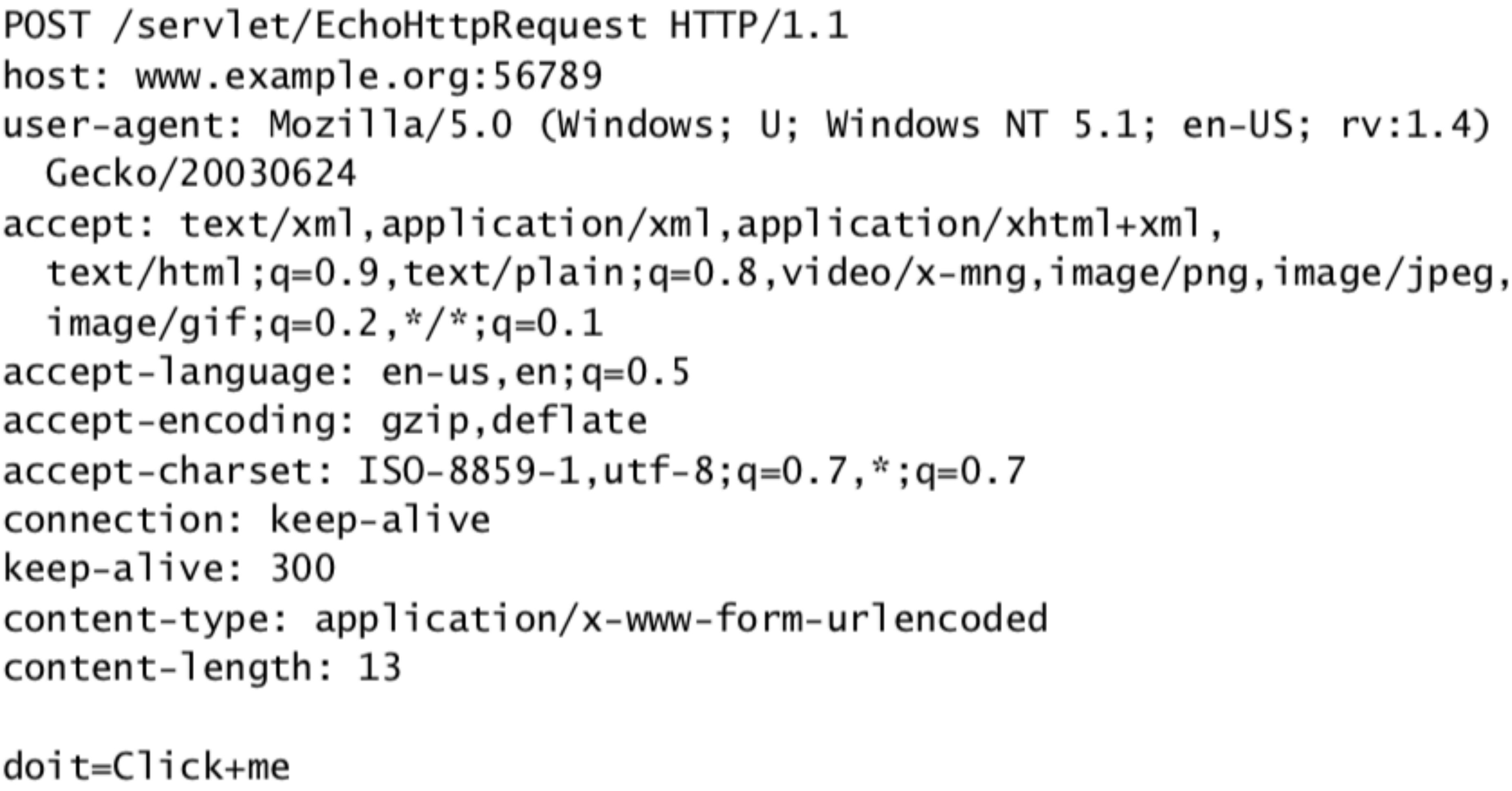
- Structure of the request:
  - start line
  - **2. header field(s)**
  - blank line
  - optional body

Header Fields and MIME Types

Host header field is used when forming the URI associated with an HTTP request.

- Each header field begins with a *field name*, such as Host, followed by a colon and then a *field value*.
- EX: host: www.example.org:56789

HTTP Request



```
POST /servlet/EchoHttpRequest HTTP/1.1
host: www.example.org:56789
user-agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.4)
  Gecko/20030624
accept: text/xml,application/xml,application/xhtml+xml,
  text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,
  image/gif;q=0.2,*/*;q=0.1
accept-language: en-us,en;q=0.5
accept-encoding: gzip,deflate
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
connection: keep-alive
keep-alive: 300
content-type: application/x-www-form-urlencoded
content-length: 13

doit=Click+me
```

- Actual HTTP request sent by a browser consists of a start line, 10 header fields and a short message body

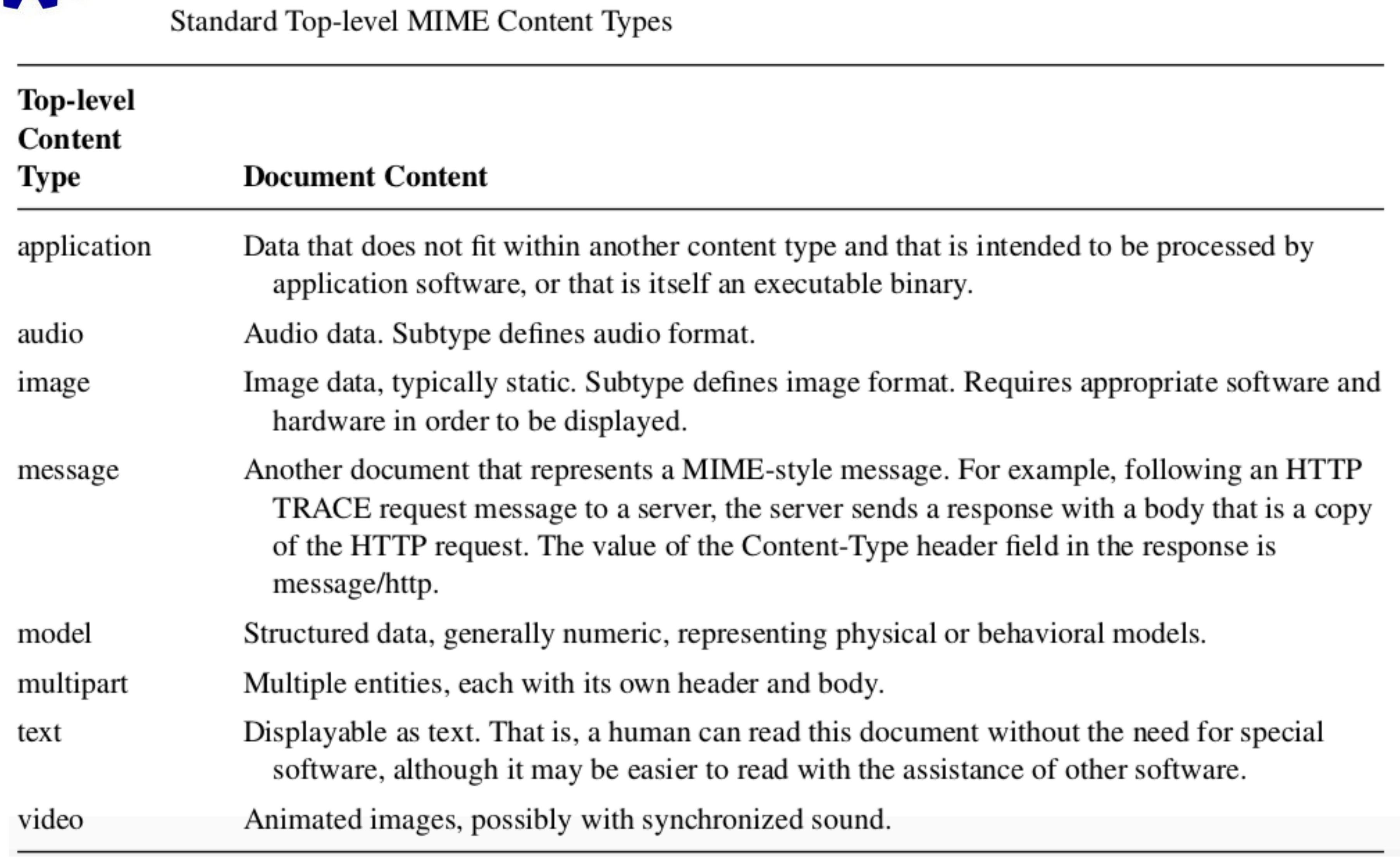
Multipurpose Internet Mail Extensions (MIME)

- MIME is an Internet e-mail protocol that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds

- The content of a MIME message is specified using a two-part, is known as the *content type* of the message.

- MIME content type syntax:
  - top–level type / subtype
  - Example: text/html and image/jpeg.

MIME



Standard Top-level MIME Content Types

Top-level Content Type	Document Content
application	Data that does not fit within another content type and that is intended to be processed by application software, or that is itself an executable binary.
audio	Audio data. Subtype defines audio format.
image	Image data, typically static. Subtype defines image format. Requires appropriate software and hardware in order to be displayed.
message	Another document that represents a MIME-style message. For example, following an HTTP TRACE request message to a server, the server sends a response with a body that is a copy of the HTTP request. The value of the Content-Type header field in the response is message/http.
model	Structured data, generally numeric, representing physical or behavioral models.
multipart	Multiple entities, each with its own header and body.
text	Displayable as text. That is, a human can read this document without the need for special software, although it may be easier to read with the assistance of other software.
video	Animated images, possibly with synchronized sound.

MIME

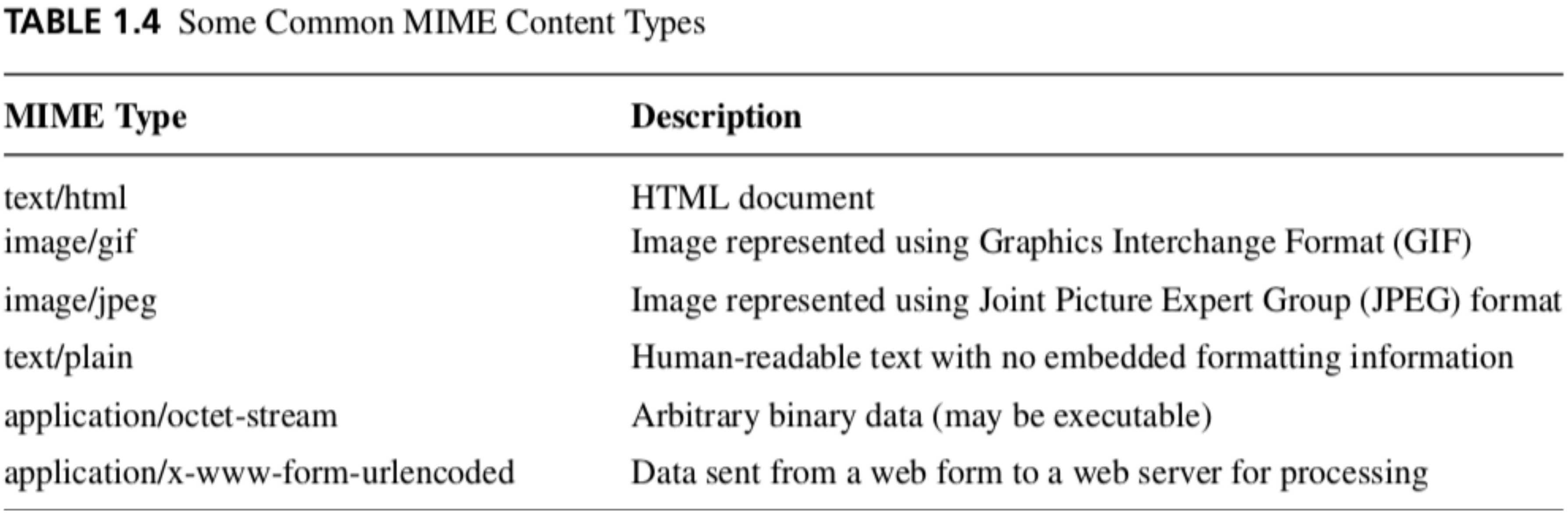


TABLE 1.4 Some Common MIME Content Types

MIME Type	Description
text/html	HTML document
image/gif	Image represented using Graphics Interchange Format (GIF)
image/jpeg	Image represented using Joint Picture Expert Group (JPEG) format
text/plain	Human-readable text with no embedded formatting information
application/octet-stream	Arbitrary binary data (may be executable)
application/x-www-form-urlencoded	Data sent from a web form to a web server for processing

- *private* (unregistered) MIME top-level types and subtypes may be used.
- A private type or subtype is indicated by an “x-” (or “X-”) prefix.

HTTP Quality values and Wildcards

- quality value, q=*num*, where *num* is a decimal number between 0 and 1, to indicate preferences.

- Higher number representing greater preference.

- * character in a header field value as a wildcard character

- the string */* in the Accept header field value represents all possible MIME types.

Header Fields

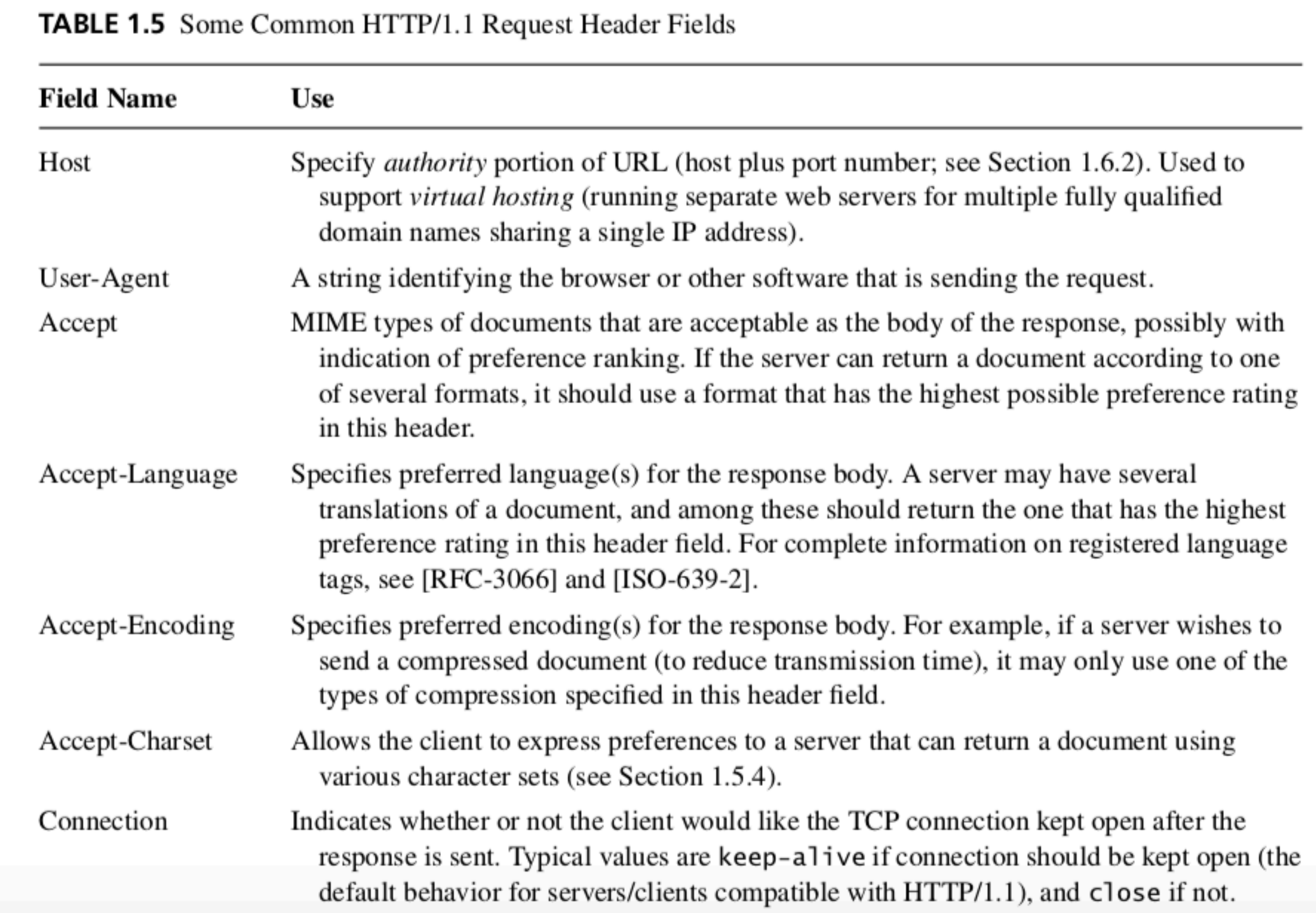
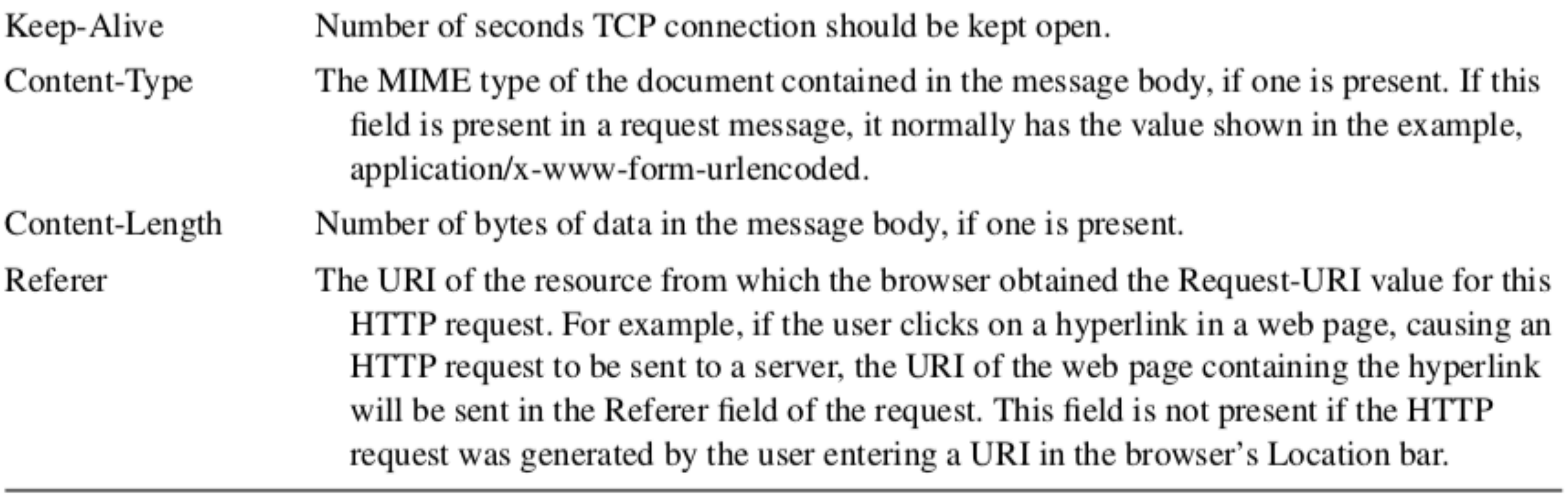


TABLE 1.5 Some Common HTTP/1.1 Request Header Fields

Field Name	Use
Host	Specify *authority* portion of URL (host plus port number; see Section 1.6.2). Used to support *virtual hosting* (running separate web servers for multiple fully qualified domain names sharing a single IP address).
User-Agent	A string identifying the browser or other software that is sending the request.
Accept	MIME types of documents that are acceptable as the body of the response, possibly with indication of preference ranking. If the server can return a document according to one of several formats, it should use a format that has the highest possible preference rating in this header.
Accept-Language	Specifies preferred language(s) for the response body. A server may have several translations of a document, and among these should return the one that has the highest preference rating in this header field. For complete information on registered language tags, see [RFC-3066] and [ISO-639-2].
Accept-Encoding	Specifies preferred encoding(s) for the response body. For example, if a server wishes to send a compressed document (to reduce transmission time), it may only use one of the types of compression specified in this header field.
Accept-Charset	Allows the client to express preferences to a server that can return a document using various character sets (see Section 1.5.4).
Connection	Indicates whether or not the client would like the TCP connection kept open after the response is sent. Typical values are `keep-alive` if connection should be kept open (the default behavior for servers/clients compatible with HTTP/1.1), and `close` if not.

Header Fields(cntd)



Keep-Alive	Number of seconds TCP connection should be kept open.
Content-Type	The MIME type of the document contained in the message body, if one is present. If this field is present in a request message, it normally has the value shown in the example, application/x-www-form-urlencoded.
Content-Length	Number of bytes of data in the message body, if one is present.
Referer	The URI of the resource from which the browser obtained the Request-URI value for this HTTP request. For example, if the user clicks on a hyperlink in a web page, causing an HTTP request to be sent to a server, the URI of the web page containing the hyperlink will be sent in the Referer field of the request. This field is not present if the HTTP request was generated by the user entering a URI in the browser's Location bar.

End of HTTP Request