COMPUTER NETWORKS

PRESENTATION

WWW and HTTP

Telugu Renuka-22B81A0529
B.Sandhya-22B81A0538
CSE-A

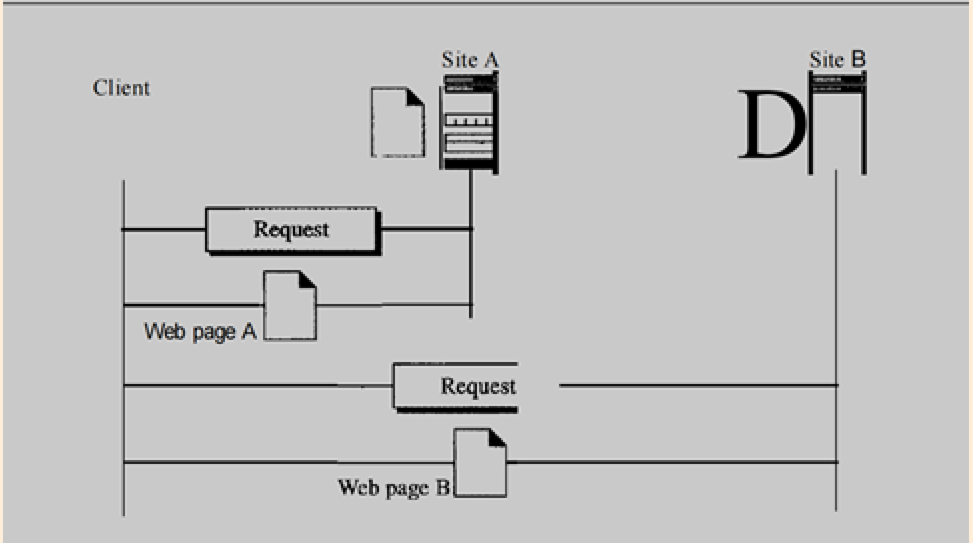
World Wide Web (WWW)

- **Definition**: The World Wide Web (WWW), commonly referred to as the web, is a system of interlinked hypertext documents accessed through the internet.
- Initiated by **CERN** for scientific research, the WWW combines flexibility, portability, and user-friendliness.

Architecture

The **WWW** operates as a **client/server** model:

- **Client**: Uses a browser to request information.
- **Server**: Hosts **Web pages** that are retrieved by the client.



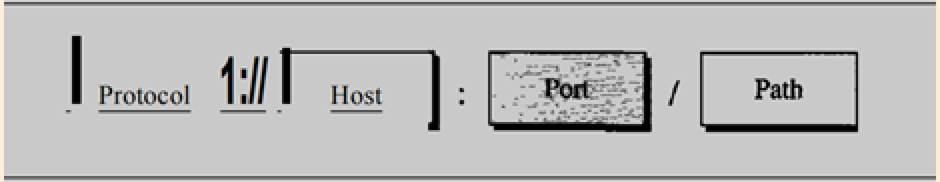
Process:

1. A client requests information from **Site A** via a URL.
2. The server at Site A retrieves the document and sends it to the client.
1. If the document contains links to other sites (like **Site B**), the client can request those too.

Browser Structure:

- **Controller**: Manages user input.
- **Client Protocol**: Retrieves documents (e.g., HTTP, FTP).
- **Interpreters**: Renders documents (e.g., HTML, Java).

Uniform Resource Locator (URL)



A **URL** is the address used to access Web pages and consists of:

1. **Protocol**: Method to retrieve the document (e.g., http://).
2. **Host**: The server where the document is located (e.g., www.example.com).
3. **Port**: (Optional) Server port number (e.g., :80).
4. **Path**: Location of the document on the server (e.g., /folder/page.html).

Cookies

Cookies are small data files used to maintain state in web applications. They enable features like:

1. **Access Control**: Restricting access to registered users.
2. **E-commerce**: Storing items in a shopping cart.
   - *Example*: A cookie might store { "itemID": "123", "price": "19.99" }.
3. **User Preferences**: Saving user settings for portals.
4. **Advertising**: Tracking user behavior for targeted ads.

Cookie Lifecycle:

1. **Creation**: The server generates a cookie upon a client request.
2. **Storage**: The cookie is stored in the browser’s directory.

Types of Web Documents

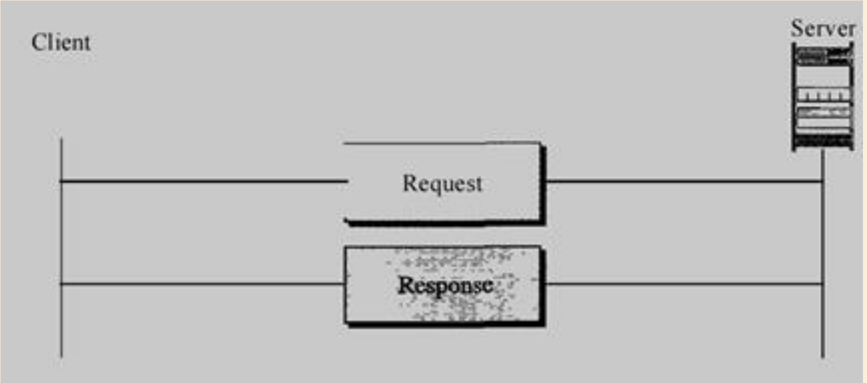
1. **Static Documents**
   - **Definition**: Fixed content that is created and stored on a server. Users can only view a copy of the document.
     - Content is determined at creation, not upon access.
     - Changes can occur on the server, but not by the user.
     - Displayed using browsers, which interpret HTML (Hypertext Markup Language).
     - HTML uses tags (e.g., <B> for bold text) for formatting.
2. **Dynamic Documents**
   - **Definition**: Created in real-time by a web server upon request.
     - Content can change with each request, like retrieving the current date and time.
     - Implemented using Common Gateway Interface (CGI) technology.
     - CGI allows for various programming languages (e.g., Perl, C, etc.) to create these documents.
     - Output includes headers (for type indication) and a body (the content).
3. **Active Documents**
   - **Definition**: Executed on the client-side (browser) and can include scripts for interaction.
     - Used for animations or user interactions.
     - Java applets and JavaScript are common technologies for creating active documents.
     - Java applets are Java programs sent from the server, while JavaScript runs directly in the browser.

HTTP

- **Definition**: Hypertext Transfer Protocol (HTTP) is a protocol used to access data on the World Wide Web.
- **Functionality**: Combines features of FTP (file transfer) and SMTP (email transfer) but is simpler since it only uses a single TCP connection (port 80).

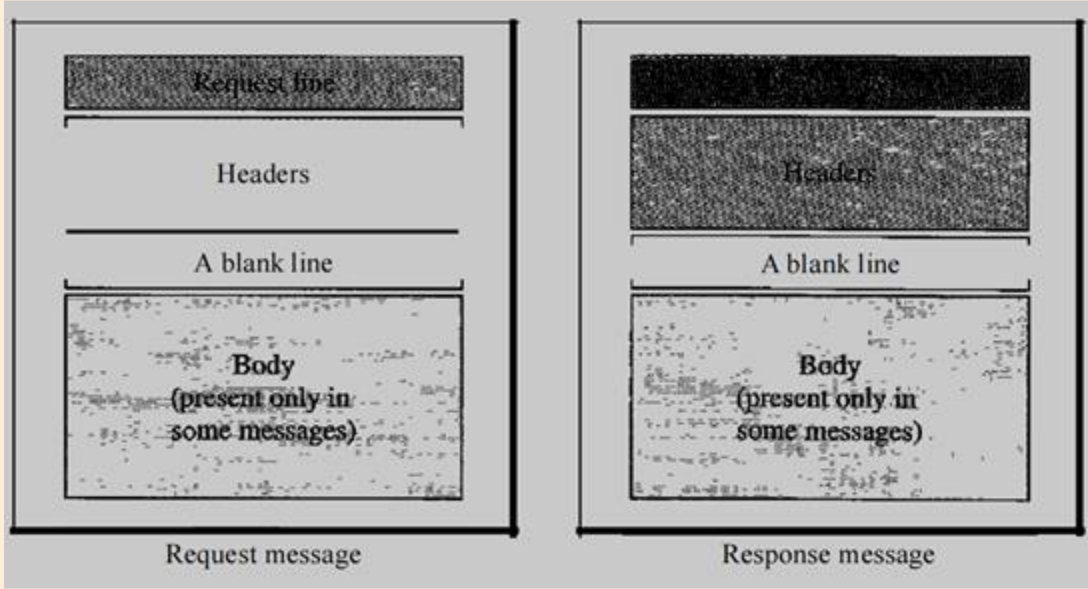
HTTP Transactions

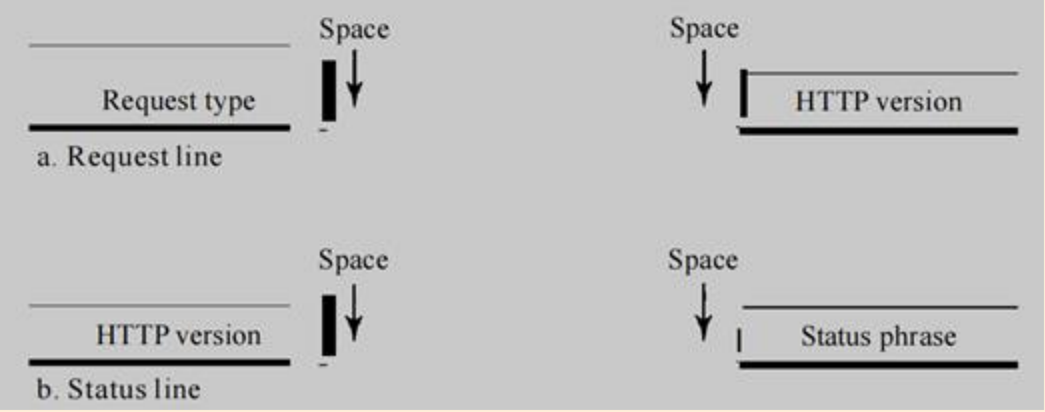
- **Stateless Protocol**: Each HTTP request is independent; the server doesn't retain any information about previous requests.
- **Request-Response Cycle**:
  - Client sends a request message.
  - Server sends a response message.



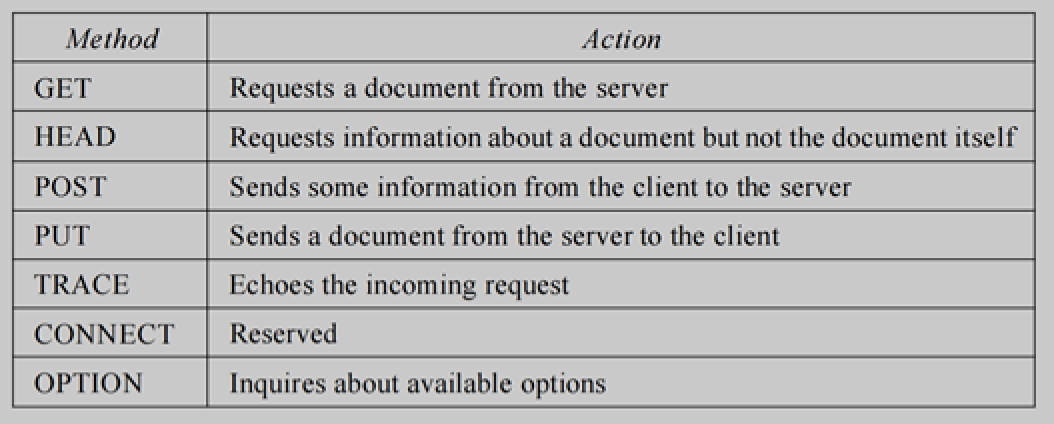
HTTP Messages

- **Request Message**: Composed of:
  - **Request Line**: Method (GET, POST, etc.), URL, and HTTP version.
  - **Headers**: Additional information (e.g., Accept, Host).
  - **Body** (optional): Contains data sent to the server.
- **Response Message**: Composed of:
  - **Status Line**: HTTP version, status code, status phrase.
  - **Headers**: Information about the response (e.g., Content-Length, Server).
  - **Body** (optional): Contains the requested content.



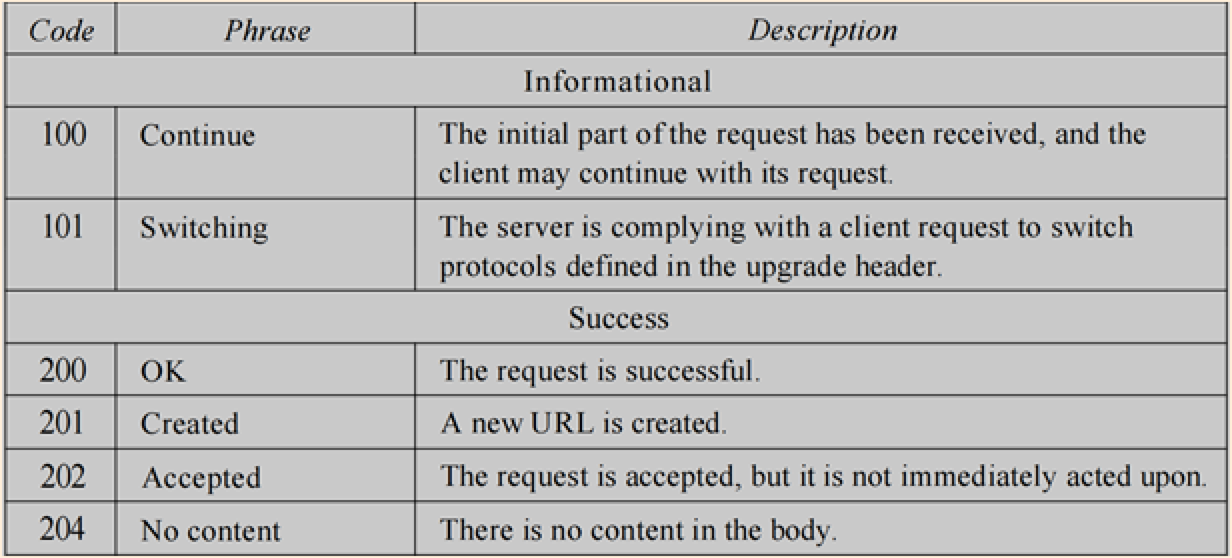


HTTP Request Methods

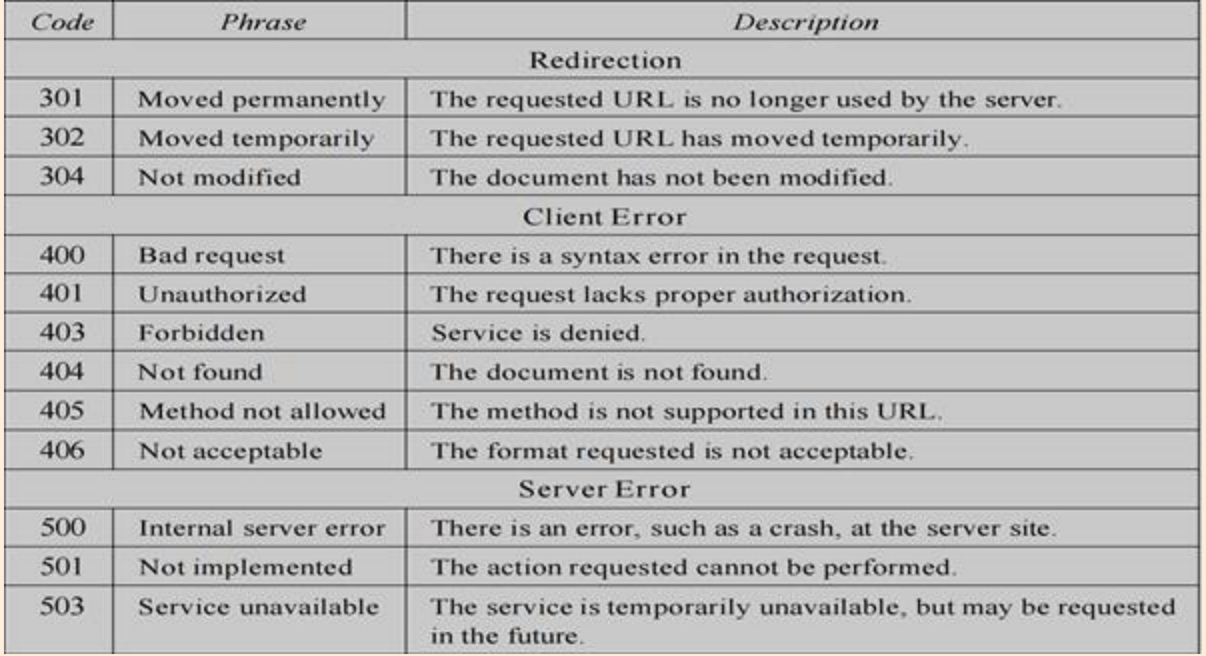


Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
OPTION	Inquires about available options

HTTP Status Codes



Code	*Phrase*	*Description*
Informational		
100	Continue	The initial part of the request has been received, and the client may continue with its request.
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header.
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.



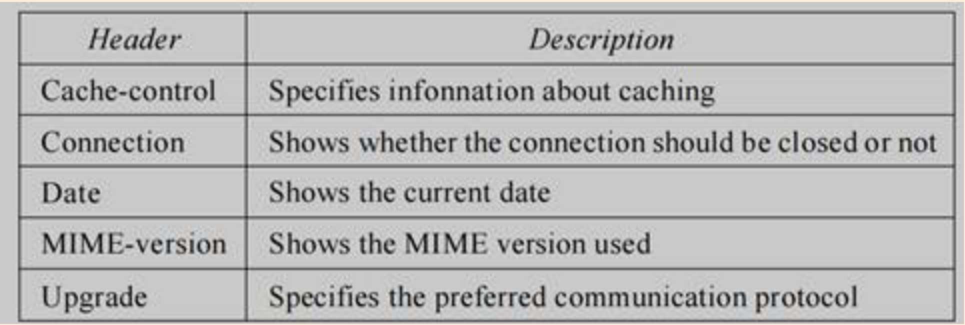
Code	*Phrase*	*Description*
		Redirection
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not been modified.
		Client Error
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
		Server Error
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future.

HTTP Headers

- **Types of Headers**:
  - **General Headers**: Used in both request and response.
  - **Request Headers**: Only in request messages.
  - **Response Headers**: Only in response messages.
  - **Entity Headers**: Provide information about the body of the document.

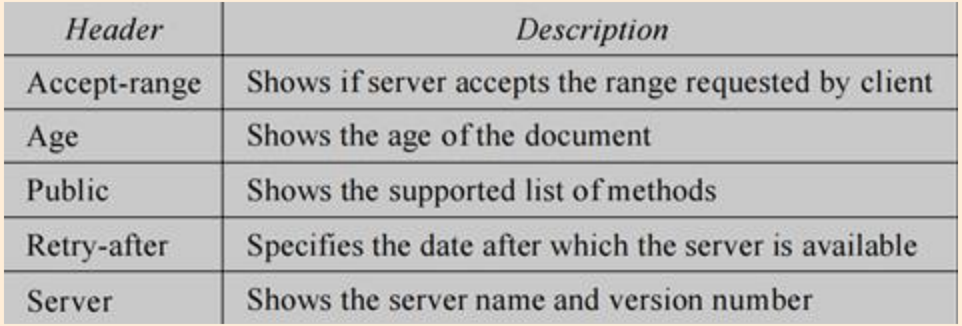
List of some common headers :

General headers



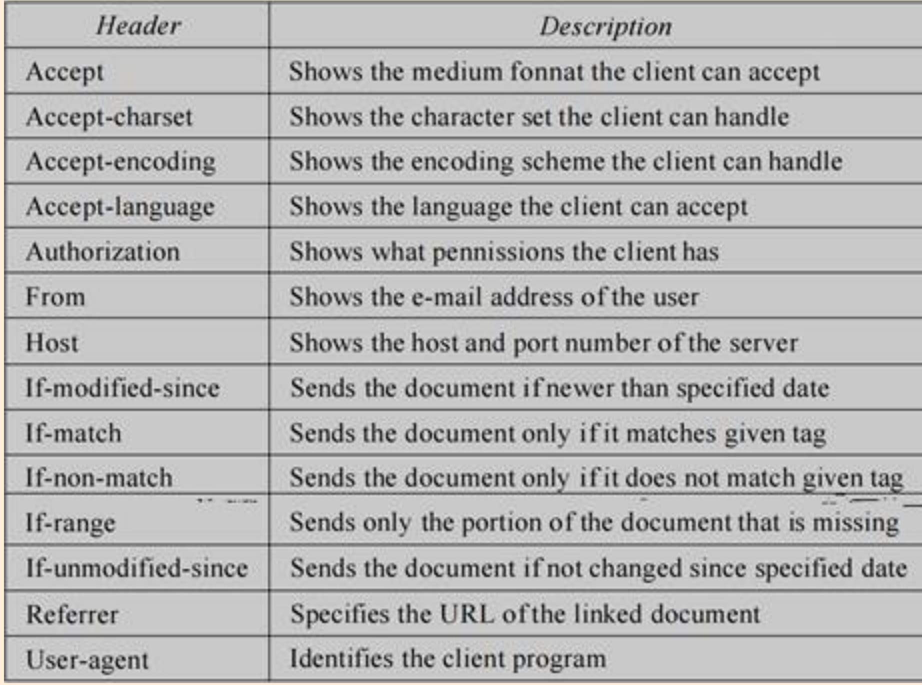
Header	*Description*
Cache-control	Specifies infonnation about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

Response headers



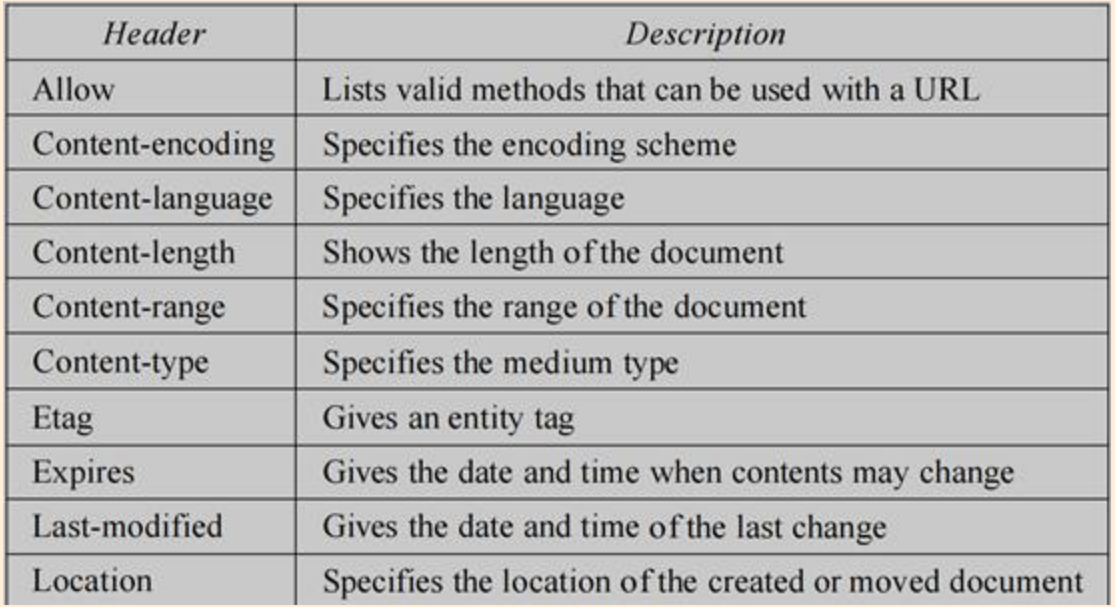
Header	*Description*
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server is available
Server	Shows the server name and version number

Request headers



Header	*Description*
Accept	Shows the medium fonnat the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what pennissions the client has
From	Shows the e-mail address of the user
Host	Shows the host and port number of the server
If-modified-since	Sends the document if newer than specified date
If-match	Sends the document only if it matches given tag
If-non-match	Sends the document only if it does not match given tag
If-range	Sends only the portion of the document that is missing
If-unmodified-since	Sends the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

Entity headers



Header	*Description*
Allow	Lists valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the medium type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created or moved document

Persistent vs Non Persistent Connections

- **Non Persistent Connection**:
  - One TCP connection per request/response.
  - High overhead due to opening and closing connections multiple times.
- **Persistent Connection (HTTP/1.1)**:
  - Keeps the connection open for additional requests.
  - Reduces latency and overhead by reusing the same connection.

Example – Persistent and Nonpersistent

- **Nonpersistent**:
  1. Client opens a TCP connection.
  2. Server sends the response and closes the connection.
  3. Connection opens and closes for each request.
- **Persistent**:
  1. TCP connection remains open for multiple requests.
  2. Server can close the connection when the client requests or after a timeout.

Proxy Servers

- **Definition**: A proxy server stores copies of responses to requests and provides them to clients without reaching the original server.
- **Advantages**:
  - Reduces the load on the original server.
  - Decreases network traffic.
  - Improves response times.

Example:
Request from client to proxy:
GET /index.html HTTP/1.1
Proxy responds with cached data.

Proxy Server Example

- **Scenario**: The proxy server caches frequently requested content and serves it without hitting the main server.
- **Request Flow**:
  1. Client sends a request to the proxy server.
  2. Proxy checks its cache.
  3. If the response is in cache, proxy returns it.
  4. If not, it forwards the request to the original server.

Thank you