Hypertext Transfer Protocol, is a foundational protocol for transferring data over the internet. It's the standard that allows web browsers to request and receive information from web servers. Essentially, it's the language used for communication between a client and a server when accessing the World Wide Web.

**HTTP STATUS CODES :**

HTTP status codes are grouped into five classes, determined by their first digit:

**1xx Informational responses:** The request has been received, and the server is continuing to process it. These are interim responses. Example: 100 Continue, 101 Switching Protocols, 102 Processing, 103 Early Hints.

**2xx Successful responses:** The request was successfully received, understood, and accepted. Example: 200 OK, 201 Created, 202 Accepted, 203 Non-Authoritative Information, 204 No Content, 205 Reset Content, 206 Partial Content, 207 Multi-Status, 208 Already Reported, 226 IM Used.

**3xx Redirection messages:** Further action needs to be taken by the client to complete the request, often involving a change in URL. Example: 300 Multiple Choices, 301 Moved Permanently, 303 See Other, 302 Found, 304 Not Modified, 305 Use Proxy, 306 unused, 307 Temporary Redirect, 308 Permanent Redirect.

**4xx Client error responses:** The request contains bad syntax or cannot be fulfilled, indicating a problem on the client's end. Example: 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 405 Method Not Allowed, 406 Not Acceptable, 407 Proxy Authentication Required, 408 Request Timeout, 409 Conflict, 410 Gone, 411 Length Required, 412 Precondition Failed, 413 Content Too Large, 414 URI Too Long, 415 Unsupported Media Type, 416 Range Not Satisfiable, 417 Expectation Failed, 418 I'm a teapot, 421 Misdirected Request, 422 Unprocessable Content, 423 Locked, 424 Failed Dependency, 425 Too Early, 426 Upgrade Required, 428 Precondition Required, 429 Too Many Requests, 431 Request Header Fields Too Large, 451 Unavailable For Legal Reasons.

**5xx Server error responses:** The server failed to fulfill an apparently valid request, indicating a problem on the server's end. Example: 500 Internal Server Error, 501 Not Implemented, 502 Bad Gateway, 503 Service Unavailable, 504 Gateway Timeout, 505 HTTP Version Not Supported, 506 Variant Also Negotiates, 507 Insufficient Storage, 508 Loop Detected, 510 Not Extended, 511 Network Authentication Required.

**HTTP HEADERS :**

HTTP headers are key-value pairs that provide additional information about a client's request or a server's response in the HTTP protocol.These headers help clients and servers understand how to handle the request or response, including details like content type, caching instructions, and authentication information.

**Request Headers:** Sent by the client to the server, providing information about the client, the requested resource, or the desired behavior.

* User-Agent: Identifies the client application (e.g., browser).
* Accept: Specifies the media types the client can handle.
* Authorization: Contains credentials for accessing protected resources.
* Content-Type: Indicates the type of data being sent in the request body.

**Response Headers:** Sent by the server to the client, providing information about the server, the response itself, or the resource being returned.

* Content-Type: Specifies the type of data being sent in the response body.
* Cache-Control: Controls caching behavior.
* Server: Indicates the server software.

**HTTP REQUEST METHODS :** Specify the action to be performed on a particular resource. The most common methods are GET, POST, PUT and DELETE, which correspond to the CRUD (Create, Read, Update, Delete) operations.

**GET:** Used to retrieve data from a server. It is idempotent, meaning multiple identical requests will have the same effect as a single request. For example, fetching a webpage's content.

**POST:** Used to send data to the server to create or update a resource. It is not idempotent. For example, submitting a form to create a new user.

**PUT:** Used to replace an existing resource with the data provided in the request. If the resource doesn't exist, it may create it. It is idempotent. For example, updating a user's profile information.

**DELETE:**Used to remove a resource from the server. It is idempotent. For example, deleting a user account.

***Additional commonly used methods:***

**OPTIONS:** Used to describe the communication options for the target resource.

**HEAD:** Similar to GET, but only returns the header information, not the actual resource content.

**TRACE:** Used to test the path between the client and the server, returning the request as it was received.

**CONNECT:** Used to establish a tunnel through the proxy server.

**PATCH:** Used to apply partial modifications to a resource. It's similar to PUT but only updates the specified fields. It is generally considered idempotent, though it depends on the implementation.