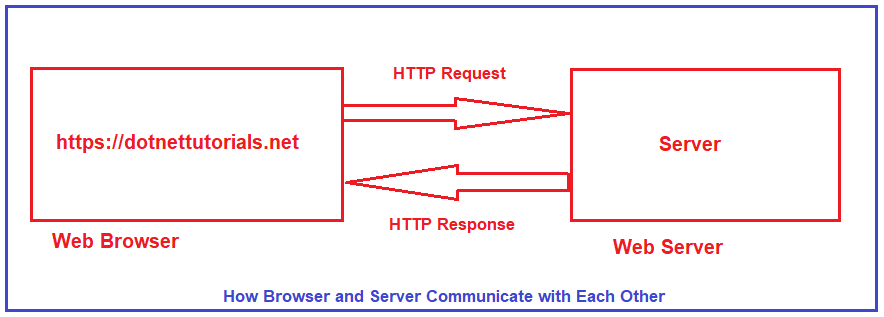
**How browser and Server Communicate with Each Other?**

HTTP stands for HyperText Transport Protocol. HTTP is used for communication between the client and server. Let us understand what does it mean by client and server with an example.

Suppose, you open the web browser, type the URL in the browser and press the enter button. As soon as you press the enter, a request is going to the server (called a web server). Whatever data you are sending from the web browser to the webserver is called a Request and whatever data we are receiving from the webserver is called a Response. This is how the browser and web server communicate with each other in the form of Request and Response. And this type of communication is only possible by using HTTP Protocol. So, the request can be termed as HTTP Request and the response can be called HTTP Response. For better understanding, please have a look at the following image.



Browser is not the only client. For example, if you are using a mobile application, then your mobile is a client. If you are calling APIs using tools like Postman and Fiddler, then Postman and Fiddlers are also the clients.

**What is HTTP?**

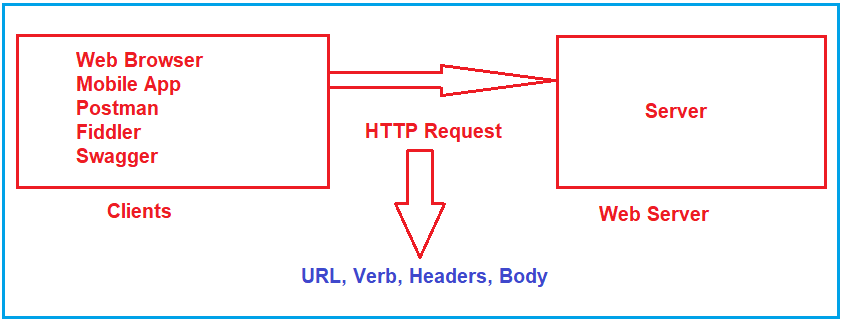
I have got the following two definitions of HTTP (HyperText Transport Protocol) from Wikipedia,

1. Hypertext Transfer Protocol (often abbreviated to HTTP) is a communications protocol. It is used to send and receive web pages and files on the internet. It was developed by Tim Berners-Lee and is now coordinated by the W3C. HTTP version 1.1 is the most commonly used version today. It is defined in RFC 2616.
2. The Hypertext Transfer Protocol (HTTP) is an application layer protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

**HTTP Request Components:**

When we send something from the client (browser, mobile, postman, fiddler, etc.) to the server (webserver) is called a Request. The request is formed with a couple of components. They are as follows:

1. **URL**: Each Request must have a unique URL
2. **Verb (Method):** Each Request must have an HTTP Verb.
3. **Header(s):** Each Request can contain one or more Headers.
4. **Body**: Each request can have a body. The body contains the data that we want to send to the server.

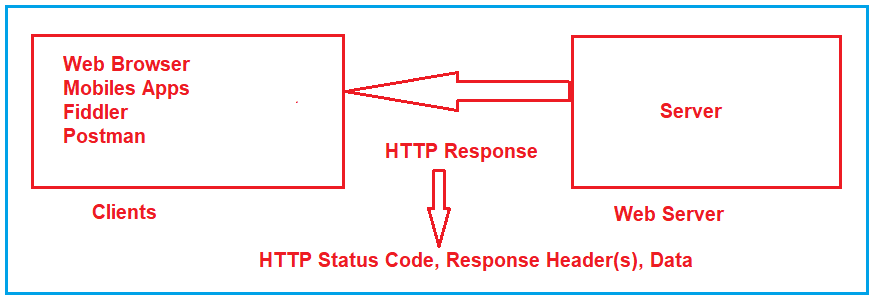


**HTTP Response Components:**

Whatever we get from the webserver to the clients called HTTP Response. The HTTP response contains the following components.

1. **HTTP Status Code:** It must have a Status Code.
2. **Response Headers:** It can have one or more response headers.
3. **Data**: Response can have data i.e. return to the client.

Along with the above three components, other components are also there. But the above three components are the important components in an HTTP Response.



**HTTP Verbs or HTTP Methods:**

We have already discussed that each HTTP Request must have an HTTP Verb. Let us understand what all HTTP Methods or Verbs are available.

**GET Method:**

The GET HTTP Method is used to Retrieve the Data. The HTTP GET method requests a representation of the specified resource. Requests using GET should only be used to request data (they shouldn’t include data). For example, you want to search something like you want to get the list of employees, list of products, you want to retrieve a book by id, etc. So, whenever you are expecting some data from the server, you need to use GET HTTP Verb.

In the case of the HTTP Get Method, the Web API may be expecting some data from the client, but the main purpose of the GET method is to fetch or retrieve data from the server. If you want to implement some kind of search functionality then the Web API may expect some data to filter out the results. In this case, the clients need to send the data.

**POST Method:**

The POST request is used to make a new entry in the database. It is not only specific to a database, whenever you want to create a new resource in your application, then you need to use the POST method.

What does it mean by adding a new Resource? It means if you want to add a new Employee, or you want to add a new product. So, here the main concept is not the database, the main concept is adding a new resource.

**PUT Method:**

PUT is also one of the HTTP Verb. The PUT method is used to update all the properties of the current resource in the database. What does it mean? For Example, we have a table called Product in our database. If we want to update all properties of a particular product (i.e. current resource) then we need to use PUT HTTP Request. So, whenever you want to update all the properties (column) of a resource (existing record in the database), then you need to use PUT Method. You cannot add a new resource using the PUT method.

**PATCH Method:**

There are some situations where you don’t want to update all the properties of an existing resource instead you want to update a few of the properties, then you need to use the PATCH method. So, the PATCH method is similar to the PUT method, but it is used to update few properties of the current resource in the database. For example, if you want to update few properties (columns) of an existing product then you need to use the PATCH method. That means if your Product table contains 10 columns, and you want to update only four columns of an existing product, then you need to use the PATCH method.

**DELETE Method:**

The DELETE method is used to delete the resource from the database. That means you are removing or deleting an existing entity from your database. In modern applications, we use two concepts for delete. One is Soft Delete and another one is Hard Delete.

Soft Delete: In your table, if you have some column like IsDeleted or IsActive, or something similar to this and you just want to update that column, then you cannot use Delete Method. In that case, you need to use the PATCH method. This is because you are not deleting the record from the database, you just update the record.

Hard Delete: If you want to remove the existing entity from the table, then you need to use the DELETE method. For example, Delete an existing product from the Product table in the database, etc.

There are other HTTP Methods available, but these are the most used HTTP Methods, we generally used in our applications.

**HTTP Status Codes:**

The HyperText Transport Protocol status code is also one of the important components of HTTP Response. The Status code is issued from the server and they give information about the response. Whenever we get any response from the server to the client, in that HTTP Response, we must have one HTTP Status code. All the HTTP Status codes are divided into five categories. They are as follows. Here, XX will represent the actual number.

1. **1XX**: Informational Response (Ex: 100, 101, 102, etc.)
2. **2XX**: Successful, whenever you get 2XX as the response code, it means the request is successful. (Ex. 200, 201, 203, etc.)
3. **3XX**: Redirection, whenever you get 3XX as the response code, it means it is re-directional i.e. some re-directional is happening on the server. (Ex. 300, 301, etc.)
4. **4XX**: Client Error, whenever you get 4XX as the response code, it means there is some problem with your request. (Ex: 400, 404, etc.)
5. **5XX**: Server Error. Whenever you get 5XX as the response code, it means there is some problem in the server. (Ex: 500, etc.).

**Frequently used HTTP Status Codes:**

The following are some of the frequently used Status codes.

1. **100: 100 means Continue.** The HTTP 100 Continue informational status response code indicates that everything so far is OK and that the client should continue with the request or ignore it if it is already finished.
2. **200: 200 means OK.** The HTTP 200 OK success status response code indicates that the request has succeeded. If you are searching for some data and you got the data properly. That means the request is successful and in that case, you will get 200 OK as the HTTP status code.
3. **201: 201 means a new resource created.** The HTTP 201 Created success status response code indicates that the request has succeeded and has led to the creation of a resource. The new resource is effectively created before this response is sent back and the new resource is returned in the body of the message, its location being either the URL of the request or the content of the Location header. If you are adding successfully a new resource by using the HTTP Post method, then in that case you will get 201 as the Status code.
4. **204: 204 means No Content.** The HTTP 204 No Content success status response code indicates that a request has succeeded, but that the client doesn’t need to navigate away from its current page. If the server processed the request successfully and it is not returning any content, then in that case you will get a 204 response status code.
5. **301: 301 means Moved Permanently.** If you are getting 301 as a status code from the server, it means the resource you are looking for is moved permanently to the URL given by the Location headers.
6. **302: 302 means Found.** If you are getting 302 as a status code from the server, it means the resource you are looking for is moved temporarily to the URL given by the Location headers.
7. **400: 400 means Bad Request.** If you are getting 400 as the status code from the server, then the issue is with the client request. If the request contains some wrong data such as malformed request syntax, invalid request message framing, or deceptive request routing, then we will get this 400 Bad Request status code.
8. **401: 401 means Unauthorized.** If you are trying to access the resource for which you don’t have access (Invalid authentication credentials), then you will get a 401 unauthorized status code from the server.
9. **404: 404 means Not Found**. If you are looking for a resource that does not exist, then you will get this 404 Not Found status code from the server. Links that lead to a 404 page are often called broken or dead links.
10. **405: 405 means Method Not Allowed.** The 405 Method Not Allowed response status code indicates that the request method is known by the server but is not supported by the target resource. For example, we have one method which is a POST method in the server and we trying to access that method from the client using GET Verb, then, in that case, you will get a 405 status code.
11. **500: 500 means Internal Server Error**. If there is some error in the server, then you will get a 500 Internal Server Error status code.
12. **503: 503 means Service Unavailable.** The 503 Service Unavailable server error response code indicates that the server is not ready to handle the request. If the server is down for maintenance or the server is overloaded then in that case, you will get the 503 Service Unavailable Status code.
13. **504: 504 means Gateway Timeout.** The 504 Gateway Timeout server error response code indicates that the server while acting as a gateway or proxy, did not get a response in time from the upstream server that is needed in order to complete the request.