HTTP Fundamentals

➢ What are the basic Features of HTTP?

There are three basic features that make HTTP a simple but powerful protocol:

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

➢ What are request methods in HTTP?

HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as *HTTP verbs*. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

GET

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

HEAD

The HEAD method asks for a response identical to that of a GET request, but without the response body.

POST

The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.

PUT

The PUT method replaces all current representations of the target resource with the request payload.

DELETE

The DELETE method deletes the specified resource.

PATCH

The PATCH method is used to apply partial modifications to a resource

HTTP Fundamentals

- **What are the differences between GET and POST methods?**

GET	POST
In GET method, values are visible in the URL.	In POST method, values are not visible in the URL.
GET has a limitation on the length of the values, generally 255 characters.	POST has no limitation on the length of the values since they are submitted via the body of HTTP.
GET performs are better compared to POST because of the simple nature of appending the values in the URL.	It has lower performance as compared to GET method because of time spent in including POST values in the HTTP body.
This method supports only string data types.	This method supports different data types, such as string, numeric, binary, etc.
GET results can be bookmarked.	POST results cannot be bookmarked.
GET request is often cacheable.	The POST request is hardly cacheable.
GET Parameters remain in web browser history.	Parameters are not saved in web browser history.

- **What is status code in HTTP?**

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

- Informational responses (100–199)
- Successful responses (200–299)
- Redirects (300–399)
- Client errors (400–499)
- Server errors (500–599)

- **What are the header fields in HTTP?**

HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers:

- **General-header:** These header fields have general applicability for both request and response messages.

- **Client Request-header:** These header fields have applicability only for request messages.
- **Server Response-header:** These header fields have applicability only for response messages.
- **Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request.

What is URI?

A URI is a string containing characters that identify a physical or logical resource. URI follows syntax rules to ensure uniformity. Moreover, it also maintains extensibility via a hierarchical naming scheme. The full form of URI is Uniform Resource Identifier.

What are Idempotent methods and why do we call them?

A request method is considered "idempotent" if the intended effect on the server of multiple identical requests with that method is the same as the effect for a single such request. Of the request methods defined by this specification, PUT, DELETE, and safe request methods are idempotent.

Explain HTTP Request & Response Messages

HTTP messages are how data is exchanged between a server and a client. There are two types of messages: requests sent by the client to trigger an action on the server, and responses, the answer from the server.

What is Session State in HTTP?

Session state, in the context of .NET, is a method keep track of the a user session during a series of HTTP requests.

What is HTTPS?

HTTPS (Hypertext Transfer Protocol Secure) is a secure version of the HTTP protocol that uses the SSL/TLS protocol for encryption and authentication. HTTPS is specified by RFC 2818 (May 2000) and uses port 443 by default instead of HTTP’s port 80.

The HTTPS protocol makes it possible for website users to transmit sensitive data such as credit card numbers, banking information, and login credentials securely over the internet. For this reason, HTTPS is especially important for securing online activities such as shopping, banking, and remote work. However, HTTPS is quickly becoming the standard protocol for *all* websites, whether or not they exchange sensitive data with users.