**Basic Term to know before learning Servlet**

**1. HTTP (Hypertext Transfer Protocol)**

**HTTP** is a protocol used for transmitting hypertext requests and information on the internet. It's the foundation of any data exchange on the Web, and it is a protocol used by the World Wide Web.

* **Methods of HTTP:**
  + **GET**: Requests data from a specified resource. Example: Fetching a webpage.
  + **POST**: Submits data to be processed to a specified resource. Example: Submitting a form.
  + **PUT**: Updates a current resource with new data.
  + **DELETE**: Deletes the specified resource.
  + **HEAD**: Similar to GET, but it only requests the headers without the body.
  + **OPTIONS**: Returns the HTTP methods that the server supports for the specified URL.
  + **PATCH**: Applies partial modifications to a resource.

**2. Client and Server**

* **Client**: The client is the end user’s device (like a computer or smartphone) that sends requests to a server over the network. It could be a web browser, a mobile app, or any software capable of making network requests.
* **Server**: The server is a powerful computer that provides resources, data, services, or programs to the client over a network. When a client sends a request, the server processes it and sends back the appropriate response.

**Example**: When you type a URL in your browser (client), it sends an HTTP request to the server where the website is hosted. The server processes this request and sends back the HTML, CSS, and JavaScript files to your browser, which then renders the webpage.

**3. Static Web Page vs. Dynamic Web Page**

* **Static Web Page**: A static web page is a simple HTML page that doesn't change content or layout with every request. The content is fixed and the same for all users.

**Example**: An "About Us" page of a website with just text and images that doesn’t change until someone manually updates it.

* **Dynamic Web Page**: A dynamic web page is generated on-the-fly and can show different content and layout based on user interaction, database queries, or other factors. It usually involves server-side scripting.

**Example**: A user’s profile page on a social media site, where the content changes based on the logged-in user’s data.

**4. URL (Uniform Resource Locator)**

A **URL** is the address used to access resources on the internet. It specifies the location of a resource and the protocol to retrieve it.

**Example**:

arduino

Copy code

https://www.example.com/index.html

* https: The protocol.
* www.example.com: The domain name or IP address of the server.
* /index.html: The specific resource on the server.

**5. URI (Uniform Resource Identifier)**

A **URI** is a string of characters that unambiguously identifies a particular resource. It can be further classified into:

* **URL**: Identifies a resource by its location.
* **URN**: Identifies a resource by its name within a given namespace.

**Example**:

makefile

Copy code

URN: urn:isbn:0451450523

In this example, urn:isbn:0451450523 is a URI that refers to a book by its ISBN.

**6. Authentication and Authorization**

* **Authentication**: The process of verifying the identity of a user or entity. It's the act of confirming whether someone is who they claim to be.

**Example**: Entering a username and password to log into a website. The system checks if the credentials match an account in its database.

* **Authorization**: The process of determining what an authenticated user is allowed to do. It controls access to resources based on the user's permissions.

**Example**: After logging in, a user might be authorized to view their profile, but not to access the admin panel of a website.

**Example Scenario:**

When you log into an online banking application:

1. **Authentication** verifies that you are the account holder by checking your credentials (e.g., username and password).
2. **Authorization** determines what actions you can perform (e.g., viewing your balance, transferring money).