Web technology Guide

- *Meenakshi M*

If you are working on web application testing then you should be aware of different web terminologies. This page will help you to learn all basic and advanced web terminologies that will definitely help you to test your web projects.

Web terminologies covered in this page are:

What is internet, www, TCP/IP, HTTP protocol, SSL (Secure socket layer), HTTPS, HTML, Web server, Web client, Proxy server, Caching, Cookies, Application server, Thin client, Daemon, Client side scripting, Server side scripting, CGI, Dynamic web pages, Digital certificates and list of HTTP status codes.

- ***Internet***
  - A global network connecting millions of computers.

- ***World Wide Web (the Web)***
  - An information sharing model that is built on top of the Internet, utilizes HTTP protocol and browsers (such as Internet Explorer) to access Web pages formatted in HTML that are linked via hyperlinks and the Web is only a subset of the Internet (other uses of the Internet include email (via SMTP), Usenet, instant messaging and file transfer (via FTP)

- ***URL (Uniform Resource Locator)***
  - The address of documents and other content on the Web. It is consisting of protocol, domain and the file. Protocol can be either HTTP, FTP, Telnet, News etc., domain name is the DNS name of the server and file can be Static HTML, DOC, Jpeg, etc., . In other words URLs are strings that uniquely identify resources on internet.

- ***TCP/IP***
  - **TCP/IP** protocol suite used to send data over the Internet. TCP/IP consists of only 4 layers - Application layer, Transport layer, Network layer & Link layer

Internet Protocols:
Application Layer - DNS, TLS/SSL, TFTP, FTP, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, TELNET, BitTorrent, RTP, rlogin.
Transport Layer- TCP, UDP, DCCP, SCTP, IL, RUDP,
Network Layer - IP (IPv4, IPv6), ICMP, IGMP, ARP, RARP, ...
Link Ethernet Layer- Wi-Fi, Token ring, PPP, SLIP, FDDI, ATM, DTM, Frame Relay, SMDS,

- ***TCP (Transmission Control Protocol)***
  - Enables two devices to establish a connection and exchange data.
  - In the Internet protocol suite, TCP is the intermediate layer between the Internet Protocol below it, and an application above it. Applications often need reliable pipe-like connections to each other, whereas the Internet Protocol does not provide such streams, but rather only unreliable packets. TCP does the task of the transport layer in the simplified OSI model of computer networks.
  - It is one of the core protocols of the Internet protocol suite. Using TCP, applications on networked hosts can create connections to one another, over which they can exchange data or packets. The protocol guarantees reliable and in-order delivery of sender to receiver data. TCP also distinguishes data for multiple, concurrent applications (e.g. Web server and e-mail server) running on the same host.

- ***IP***
  - Specifies the format of data packets and the addressing protocol.
    The Internet Protocol (IP) is a data-oriented protocol used for communicating data across a packet-switched internet work. IP is a network layer protocol in the internet protocol suite. Aspects of IP are IP addressing and routing. Addressing refers to how end hosts become assigned IP addresses. IP routing is performed by all hosts, but most importantly by internetwork routers

- ***IP Address***
  - A unique number assigned to each connected device, often assigned dynamically to users by an ISP on a session-by-session basis – dynamic IP address. Increasingly becoming dedicated, particularly with always-on broadband connections – static IP address.

- ***Packet***
  - A portion of a message sent over a TCP/IP Network. It contains content and destination

- ***HTTP (Hypertext Transfer Protocol)***
  - Underlying protocol of the World Wide Web. Defines how messages are formatted and transmitted over a TCP/IP network for Web sites. Defines what actions Web servers and Web browsers take in response to various commands.
  - HTTP is stateless. The advantage of a stateless protocol is that hosts don't need to retain information about users between requests, but this forces the use of alternative methods for maintaining users' state, for example, when a host would like to customize content for a user who has visited before. The common method for solving this problem involves the use of sending and requesting cookies. Other methods are session control, hidden variables, etc
  - example: when you enter a URL in your browser, an HTTP command is sent to the Web server telling to fetch and transmit the requested Web page

    - HEAD: Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.
    - GET : Requests a representation of the specified resource. By far the most common method used on the Web today.
    - POST : Submits user data (e.g. from a HTML form) to the identified resource. The data is included in the body of the request.
    - PUT: Uploads a representation of the specified resource.
    - DELETE: Deletes the specified resource (rarely implemented).
    - TRACE: Echoes back the received request, so that a client can see what intermediate servers are adding or changing in the request.
    - OPTIONS:
    - Returns the HTTP methods that the server supports. This can be used to check the functionality of a web server.
    - CONNECT: For use with a proxy that can change to being an SSL tunnel.

- ***HTTP pipelining***
  - appeared in HTTP/1.1. It allows clients to send multiple requests at once, without waiting for an answer. Servers can also send multiple answers without closing their socket. This results in fewer roundtrips and faster load times. This is particularly useful for satellite Internet connections and other connections with high latency as separate requests need not be made for each file. Since it is possible to fit several HTTP requests in the same TCP packet, HTTP pipelining allows fewer TCP packets to be sent over the network, reducing network load. HTTP pipelining requires both the client and the server to support it. Servers are required to support it in order to be HTTP/1.1 compliant, although they are not required to pipeline responses, just to accept pipelined requests.

- ***HTTP-Tunnel***
  - technology allows users to perform various Internet tasks despite the restrictions imposed by firewalls. This is made possible by sending data through HTTP (port 80). Additionally, HTTP-Tunnel technology is very secure, making it indispensable for both average and business communications. The HTTP-Tunnel client is an application that runs in your system tray acting as a SOCKS server, managing all data transmissions between the computer and the network.

- ***HTTP streaming***
  - It is a mechanism for sending data from a Web server to a Web browser in response to an event. HTTP Streaming is achieved through several common mechanisms. In one such mechanism the web server does not terminate the response to the client after data has been served. This differs from the typical HTTP cycle in which the response is closed immediately following data transmission. The web server leaves the response open such that if an event is received, it can immediately be sent to the client. Otherwise the data would have to be queued until the client's next request is made to the web server. The act of repeatedly queing and re-requesting information is known as a Polling mechanism. Typical uses for HTTP Streaming include market data distribution (stock tickers), live chat/messaging systems, online betting and gaming, sport results, monitoring consoles and Sensor network monitoring.

- ***HTTP referrer***
  - It signifies the webpage which linked to a new page on the Internet. By checking the referer, the new page can see where the request came from. Referer logging is used to allow websites and web servers to identify where people are visiting them from, for promotional or security purposes. Since the referer can easily be spoofed (faked), however, it is of limited use in this regard except on a casual basis. A dereferer is a means to strip the details of the referring website from a link request so that the target website cannot identify the page which was clicked on to originate a request. Referer is a common misspelling of the word referrer. It is so common, in fact that it made it into the official specification of HTTP – the communication protocol of the World Wide Web – and has therefore become the standard industry spelling when discussing HTTP referers.

- ***SSL (Secure Sockets Layer)***
  - Protocol for establishing a secure connection for transmission, it uses the HTTPS convention
  - SSL provides endpoint authentication and communications privacy over the Internet using cryptography. In typical use, only the server is authenticated (i.e. its identity is ensured) while the client remains unauthenticated; mutual authentication requires public key infrastructure (PKI) deployment to clients. The protocols allow client/server applications to communicate in a way designed to prevent eavesdropping, tampering, and message forgery.

  - SSL involves a number of basic phases:
    - Peer negotiation for algorithm support
    - Public key encryption-based key exchange and certificate-based authentication
    - Symmetric cipher-based traffic encryption
    - During the first phase, the client and server negotiate which cryptographic algorithms will be used. Current implementations support the following choices:

    - for public-key cryptography: RSA, Diffie-Hellman, DSA or Fortezza;
    - for symmetric ciphers: RC2, RC4, IDEA, DES, Triple DES or AES;
    - For one-way hash functions: MD5 or SHA.

- ***HTTPS***
  - is a URI scheme which is syntactically identical to the http: scheme normally used for accessing resources using HTTP. Using an https: URL indicates that HTTP is to be used, but with a different default port and an additional encryption/authentication layer between HTTP and TCP. This system was invented by Netscape Communications Corporation to provide authentication and encrypted communication and is widely used on the Web for security-sensitive communication, such as payment transactions.

- ***HTML (Hypertext Markup Language)***
  - The authoring language used to create documents on the World Wide Web
  - Hundreds of tags can be used to format and layout a Web page's content and to hyperlink to other Web content.
- ***Hyperlink***
  - Used to connect a user to other parts of a web site and to other web sites and web-enabled services.

- ***Web server***
  - A computer that is connected to the Internet. Hosts Web content and is configured to share that content.
  - Webserver is responsible for accepting HTTP requests from clients, which are known as Web browsers, and serving them Web pages, which are usually HTML documents and linked objects (images, etc.).
    - Examples:
      - Apache HTTP Server from the Apache Software Foundation.
      - Internet Information Services (IIS) from Microsoft.
      - Sun Java System Web Server from Sun Microsystems, formerly Sun ONE Web Server, iPlanet Web Server, and Netscape Enterprise Server.
      - Zeus Web Server from Zeus Technology

- ***Web client***
  - Most commonly in the form of Web browser software such as Internet Explorer or Netscape
  - Used to navigate the Web and retrieve Web content from Web servers for viewing.

- ***Proxy server***
  - An intermediary server that provides a gateway to the Web (e.g., employee access to the Web most often goes through a proxy)
  - Improves performance through caching and filters the Web
  - The proxy server will also log each user interaction.

- ***Caching***
  - Web browsers and proxy servers save a local copy of the downloaded content – pages that display personal information should be set to prohibit caching.

- ***Web form***
  - A portion of a Web page containing blank fields that users can fill in with data (including personal info) and submits for Web server to process it.

- ***Web server log***
  - Every time a Web page is requested, the Web server may automatically logs the following information:
    - the IP address of the visitor
    - date and time of the request
    - the URL of the requested file
    - the URL the visitor came from immediately before (referrer URL)
    - the visitor's Web browser type and operating system

- ***Cookies***
  - A small text file provided by a Web server and stored on a users PC the text can be sent back to the server every time the browser requests a page from the server. Cookies are used to identify a user as they navigate through a Web site and/or return at a later time. Cookies enable a range of functions including personalization of content.

- ***Session vs. persistent cookies***
  - A Session is a unique ID assigned to the client browser by a web server to identify the state of the client because web servers are stateless.
  - A session cookie is stored only while the user is connected to the particular Web server – the cookie is deleted when the user disconnects
  - Persistent cookies are set to expire at some point in the future – many are set to expire a number of years forward

- ***Socket***

  - A *socket* is a network communications endpoint.

- ***Application Server***

  - An application server is a server computer in a computer network dedicated to running certain software applications. The term also refers to the software installed on such a computer to facilitate the serving of other applications. Application server products typically bundle middleware to enable applications to intercommunicate with various qualities of service — reliability, security, non-repudiation, and so on. Application servers also provide an API to programmers, so that they don't have to be concerned with the operating system or the huge array of interfaces required of a modern web-based application. Communication occurs through the web in the form of HTML and XML, as a link to various databases, and, quite often, as a link to systems and devices ranging from huge legacy applications to small information devices, such as an atomic clock or a home appliance.
  - An application server exposes business logic to client applications through various protocols, possibly including HTTP. the server exposes this business logic through a component API, such as the EJB (Enterprise JavaBean) component model found on J2EE (Java 2 Platform, Enterprise Edition) application servers. Moreover, the application server manages its own resources. Such gate-keeping duties include security, transaction processing, resource pooling, and messaging
  - Ex: JBoss (Red Hat), WebSphere (IBM), Oracle Application Server 10g (Oracle Corporation) and WebLogic (BEA)

- ***Thin Client***
  - A thin client is a computer (client) in client-server architecture networks which has little or no application logic, so it has to depend primarily on the central server for processing activities. It is designed to be especially small so that the bulk of the data processing occurs on the server.

- ***Thick client***
  - It is a client that performs the bulk of any data processing operations itself, and relies on the server it is associated with primarily for data storage.

- ***Daemon***
  - It is a computer program that runs in the background, rather than under the direct control of a user; they are usually instantiated as processes. Typically daemons have names that end with the letter "d"; for example, syslogd is the daemon which handles the system log. Daemons typically do not have any existing parent process, but reside directly under init in the process hierarchy. Daemons usually become daemons by forking a child process and then making the parent process kill itself, thus making init adopt the child. This practice is commonly known as "fork off and die." Systems often start (or "launch") daemons at boot time: they often serve the function of responding to network requests, hardware activity, or other programs by performing some task. Daemons can also configure hardware (like devfsd on some Linux systems), run scheduled tasks (like cron), and perform a variety of other tasks.

- ***Client-side scripting***
  - Generally refers to the class of computer programs on the web that are executed client-side, by the user's web browser, instead of server-side (on the web server). This type of computer programming is an important part of the Dynamic HTML (DHTML) concept, enabling web pages to be scripted; that is, to have different and changing content depending on user input, environmental conditions (such as the time of day), or other variables.

  - Web authors write client-side scripts in languages such as JavaScript (Client-side JavaScript) or VBScript, which are based on several standards:
    - HTML scripting
    - HTTP

- Document Object Model
  - Client-side scripts are often embedded within an HTML document, but they may also be contained in a separate file, which is referenced by the document (or documents) that use it. Upon request, the necessary files are sent to the user's computer by the web server (or servers) on which they reside. The user's web browser executes the script, then displays the document, including any visible output from the script. Client-side scripts may also contain instructions for the browser to follow if the user interacts with the document in a certain way, e.g., clicks a certain button. These instructions can be followed without further communication with the server, though they may require such communication.

- ***Server-side Scripting***
  - It is a web server technology in which a user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. It is usually used to provide interactive web sites that interface to databases or other data stores. This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.

    - ASP: Microsoft designed solution allowing various languages (though generally VBscript is used) inside a HTML-like outer page, mainly used on Windows but with limited support on other platforms.
    - ColdFusion: Cross platform tag based commercial server side scripting system.
    - JSP: A Java-based system for embedding code in HTML pages.
    - Lasso: A Datasource neutral interpreted programming language and cross platform server.
    - SSI: A fairly basic system which is part of the common apache web server. Not a full programming environment by far but still handy for simple things like including a common menu.

- PHP : Common opensource solution based on including code in its own language into an HTML page.
- Server-side JavaScript: A language generally used on the client side but also occasionally on the server side.
- SMX : Lisplike opensource language designed to be embedded into an HTML page.

- ***Common Gateway Interface (CGI)***
  - is a standard protocol for interfacing external application software with an information server, commonly a web server. This allows the server to pass requests from a client web browser to the external application. The web server can then return the output from the application to the web browser.

- ***Dynamic Web pages:***

  - can be defined as: (1) Web pages containing dynamic content (e.g., images, text, form fields, etc.) that can change/move without the Web page being reloaded or (2) Web pages that are produced on-the-fly by server-side programs, frequently based on parameters in the URL or from an HTML form. Web pages that adhere to the first definition are often called Dynamic HTML or DHTML pages. Client-side languages like JavaScript are frequently used to produce these types of dynamic web pages. Web pages that adhere to the second definition are often created with the help of server-side languages such as PHP, Perl, ASP/.NET, JSP, and languages. These server-side languages typically use the Common Gateway Interface (CGI) to produce dynamic web pages.

- ***Digital Certificates***

In cryptography, a public key certificate (or identity certificate) is a certificate which uses a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust s
"endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together.

Certificates can be used for the large-scale use of public-key cryptography. Securely exchanging secret keys amongst users becomes impractical to the point of effective impossibility for anything other than quite small networks. Public key cryptography provides a way to avoid this problem. In principle, if Alice wants others to be able to send her secret messages, she need only publish her public key. Anyone possessing it can then send her secure information. Unfortunately, David could publish a different public key (for which he knows the related private key) claiming that it is Alice's public key. In so doing, David could intercept and read at least some of the messages meant for Alice. But if Alice builds her public key into a certificate and has it digitally signed by a trusted third party (Trent), anyone who trusts Trent can merely check the certificate to see whether Trent thinks the embedded public key is Alice's. In typical Public-key Infrastructures (PKIs), Trent will be a CA, who is trusted by all participants. In a web of trust, Trent can be any user, and whether to trust that user's attestation that a particular public key belongs to Alice will be up to the person wishing to send a message to Alice.

In large-scale deployments, Alice may not be familiar with Bob's certificate authority (perhaps they each have a different CA — if both use employer CAs, different employers would produce this result), so Bob's certificate may also include his CA's public key signed by a "higher level" CA2, which might be recognized by Alice. This process leads in general to a hierarchy of certificates, and to even more complex trust relationships. Public key infrastructure refers, mostly, to the software that manages certificates in a large-scale setting. In X.509 PKI systems, the hierarchy of certificates is always a top-down tree, with a root certificate at the top, representing a CA that is 'so central' to the scheme that it does not need to be authenticated by some trusted third party.

A certificate may be revoked if it is discovered that its related private key has been compromised, or if the relationship (between an entity and a public key) embedded in the certificate is discovered to be incorrect or has changed; this might occur, for example, if a person changes jobs or names. A revocation will likely be a rare occurrence, but the possibility means that when a certificate is trusted, the user should always check its validity. This can be done by comparing it against a certificate revocation list (CRL) — a list of revoked or cancelled certificates. Ensuring that such a list is up-to-date and accurate is a core function in a centralized PKI, one which requires both staff and budget and one which is therefore sometimes not properly done. To be effective, it must be readily available to any who needs it whenever it is needed and must be updated frequently. The other way to check a certificate validity is to query the certificate authority using the Online Certificate Status Protocol (OCSP) to know the status of a specific certificate.

Both of these methods appear to be on the verge of being supplanted by XKMS. This new standard, however, is yet to see widespread implementation.

A certificate typically includes:

The public key being signed.
A name, which can refer to a person, a computer or an organization.
A validity period.
The location (URL) of a revocation center.
The most common certificate standard is the ITU-T X.509. X.509 is being adapted to the Internet by the IETF PKIX working group.

Classes
Verisign introduced the concept of three classes of digital certificates:

Class 1 for individuals, intended for email;
Class 2 for organizations, for which proof of identity is required; and
Class 3 for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authority (CA)

- ***List of HTTP status codes***

1xx Informational
Request received, continuing process.

100: Continue
101: Switching Protocols

2xx Success
The action was successfully received, understood, and accepted.
200: OK
201: Created
202: Accepted
203: Non-Authoritative Information
204: No Content
205: Reset Content
206: Partial Content

3xx Redirection
The client must take additional action to complete the request.
300: Multiple Choices
301: Moved Permanently
302: Moved Temporarily (HTTP/1.0)
302: Found (HTTP/1.1)
see 302 Google Jacking
303: See Other (HTTP/1.1)
304: Not Modified
305: Use Proxy

Many HTTP clients (such as Mozilla and Internet Explorer) don't correctly handle responses with this status code.
306: (no longer used, but reserved)
307: Temporary Redirect

4xx Client Error
The request contains bad syntax or cannot be fulfilled.
400: Bad Request
401: Unauthorized
Similar to 403/Forbidden, but specifically for use when authentication is possible but has failed or not yet been provided. See basic authentication scheme and digest access authentication.
402: Payment Required
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: Request Entity Too Large
414: Request-URI Too Long
415: Unsupported Media Type
416: Requested Range Not Satisfiable
417: Expectation Failed

5xx Server Error
The server failed to fulfill an apparently valid request.
500: Internal Server Error
501: Not Implemented
502: Bad Gateway
503: Service Unavailable
504: Gateway Timeout
505: HTTP Version Not Supported
509: Bandwidth Limit Exceeded

For more details Visit:

- **Software Testing Resources: http://www.softwaretestinghelp.com/resources/**
- **Software Testing Articles: http://www.softwaretestinghelp.com/sitemap/**
- **Software Testing Jobs: http://www.softwaretestinghelp.com/jobs/**