**WWW (World Wide Web)**

-collection of web resources and application that gives access to use resources via the internet

-greatest information system

Sir Tim Berners-Lee

-invented WWW in 1989

* CERN (Conseil Européen pour la Recherche Nucléaire)
  + Switzerland
* Text-based resources

2 Types of Web resource

* Static-already exists
* Dynamic-produced on the fly

Application

* WWW-client-server
* Web servers
  1. Hosting/Storage (electronic)
  2. Request by the client
* Web clients
  + Browser
  + Spiders-application that crawls the web that gets information. (ex. Google spider)

Internet

-shortcut of Internetwork (inter-across boundaries, network-establishing a connection)

-global, network of networks

* hosts- devices

NETWORKS

* interconnection media
* protocols(standards)

File Transfer Protocol (FTP)-allows to access files

http://webtek.org/index.html

scheme

domain name/host name

Fiddler – web debugging proxy

**HTTP (Hypertext Transfer Protocol)**

* Application layer communications protocol used to access resources (hypertext/hypermedia) on the world wide web
* Invented by Sir Tim Berners – Lee at CERN in 1989
* Jointly developed by the W3C and the IETF

Also founded by Berners-Lee

* Used to retrieve resources

Version History

* HTTP 0.9 (1991) (GET)
* HTTP 1.0 (RFC 1945, May 1996) (GET,HEAD,POST)
* HTTP 1.1(RFC Jan 1997, RFC 2016 Jun 1999), RFC 7230-7235 (Jun 2014) (Persistent Connection)
* HTTP 2 (RFC 7540 May 2015)
  + SPDY (From google)
    - Faster
  + Server Push
    - Lessen page load times

Type Lining

* “send an html while still sending other html without getting response”
* latency
* “pipe lining”

TCP (Transmission Control Protocol)

Port (0-65535)

**HTTP Fundamentals**

* HTTP typically runs on top of TCP/IP, using TCP port(0-65535) 80 by default, or TCP port 443 for HTTPS(HTTP over SSL/TLS)
* HTTP is based on a client-server architecture
  + Servers
    - Origin servers
    - Proxy servers, gateways, tunnels (encryptions)
  + Clients a.k.a user agent(UA)
    - Web browsers, web crawlers/spiders, other end user tools and applications

TLS (Transfer Layer Security)

SSL-encrypts data

* HTTP uses a request- response standard protocol
  + The client sends an HTTP request message on the server
  + The server processes the request and replies with an HTTP response message
* HTTP is a statistics communication protocol
  + Servers do not keep information about clients in-between requests
* HTTP provides support for other functionalities such as:
  + Cache control (local storage)
  + Content media type (MULTIPURPOSE INTERNET MAIL EXTENTIONS or MIME) specification
  + Language and character set specification
  + Content/transfer codings – compression (rar,zip)
  + Content negotiation
  + Client-server protocol negotiations – switch protocols
  + Persistent connections
  + Request pipelining/multiplexing
  + Authentication/authorization
  + Etc.

**HTTP Resource Addressing**

* HTTP resources are identified using URIs(RFC (3986),or more, specifically HTTP URLs)
  + Scheme ( http or https)
  + Authority
    - User information or authentication credentials (deprecated) (usr:pwd)
    - Host
      * Domain name (resolved to an IP address using DNS) of the server where the resource resides)
    - Port number
  + Path to resource (resolved relative to the document root on the server)
    - May refer to a static or dynamic resource (as-is) – (produce)
  + Query
    - Typically provided as key=value pairs, with ampersand (&) separators between key/value pairs
    - May be URL-encoded
  + Fragment identifier
    - Starts w/ hashtag (#) – not sent to server
    - Bookmark

Example

http://usr:pwd@server.org:81/info/profile.php?id=1234#addr

**HTTP Request Message**

* Request Line (CRLF-terminated line consisting of three space-separated values)
  + Method
  + Request Target
  + Protocol Version
* Message Headers (general,request,and/or entity headers)
  + HTTP 1.1 requires at least the **HOST** request header to be provided
* Empty Line (CRLF) - new line
* Message Body a.k.a payload (Optional)

**HTTP Response Message**

* Status Line (CRLF)
  + Protocol Version
  + Status Code
  + Reason phrase
* Message Headers (general, response, and/or entity headers)
* Empty Line (CRLF)
* Message Body (CRLF)

**HTTP Request Methods**

STANDARD METHODS (8)

GET, HEAD, POST, PUT, DELETE, OPTIONS, TRACE, CONNECT

* GET (w/ payload)
  + used to retrieve (entity)
  + most commonly used HTTP method
* HEAD (w/o payload)
  + identical to GET, except that server gives response yon line and headers (no message body)
* POST
  + used in submitting HTML form data (submitted as a query)
* PUT
  + create or replace the state of the target resource
* DELETE
  + remove the association between the target resource
* OPTIONS
  + allows clients to determine the options and/or requirements
  + request target may be an asterisk (\*)
* TRACE
  + used for testing/diagnostics of the request/response chain
* CONNECT
  + requests the establishment of a tunnel to the destination origin server
  + commonly used to create an end-to-end virtual connection

METHOD PROPERTIES

Safe Methods

-GET, HEAD, OPTIONS, TRACE

-Read-only

Idempotent Methods

-GET, HEAD, OPTIONS, TRACE, PUT, DELETE

-same effect as the effect of a single such request

Cacheable Methods

-GET, HEAD, POST

-allowed to be stored for future reuse

**HTTP Message Headers (RFC 2616)**

General Header Fields (clients & servers)

* Cache-control
* Connection
* Data
* Pragma
* Trailer
* Transfer-Encoding
* Upgrade
* Via
* Warning

Request Header Fields

* Accept
* Accept-charset
* Accept-encoding
* Accept-language
* Authorization
* Expect
* From
* Host
* If-Match
* If-None-Match

Response Header Fields

* Accept-Ranges
* Age
* ETag
* Location
* Proxy-Authenticate
* Retry After
* Server
* Vary
* WWW-Authenticate

Entity Header Fields (payload)

* Allow
* Content-encoding
* Content-language
* Content-length
* Content-location
* Content-MDS
* Content-range
* Content-type
* Expires
* Last-Modifier

**HTTP Status Codes (RFC 2616)**

* Informational (1xx)
* Success (2xx)
* Redirection (3xx)
* Client Error (4xx)
* Server Error (5xx)

**HTTP Extensions**

-by redefining new request methods, message headers and/or status codes and the implementing servers and/or clients that make use of the new functionality

WebDAV (RFC 4918) (Distributed Authoring and Versioning)

* Request Methods
  + PROF FIND, PROP PATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK
* Message Headers
  + DAV, Depth, Destination, If, Lock-Token, Overwrite, Timeout
* Status Codes
  + (207) Multi Status, (422) Unprocessable Entity, (423) Locked, (424) Failed Dependency, (507) Insufficient Storage

Content Negotiation

-an HTTP client can negotiate for the resource that it prefers using the Accept, Accept-charset, Accept-encoding, and Accept-Language Headers

Range Requests