**HTTP Resource Accessing**

HTTP resources are identified using URIs, which tells what the resource is, or more, specifically HTTP URL.

URL, which stands for Uniform Resource Locator, tells where in the web the resource is. A URL contains different components such as:

* Scheme

The scheme specifies the protocol to be used to access the specified resource from the internet. Different protocols are used to access specific types of resources but to get a hypertext document, HTTP or HTTPS protocols are used. HTTP is for accessing resources without SSL, or Secure Sockets Layer, while HTTPS is for accessing with SSL.

* Authority

The authority specifies the host that has the resource. It can consist of three parts:

* User information/ information credentials – it contains the information of the user, such as the username and the password.
* Host – it is the domain name of the server where the resource resides, or will be created if it does not exist yet. The domain name is resolved to an IP address using DNS, or Domain Name System.
* Port number – it identifies the port which the server listens to for the specific request. The default is 80 and is usually omitted when well-known.
* Path to resource

This simply refers to the resource to be accessed and it may refer to a static or dynamic resource. It is resolved relative to the document root.

* Query

The query is typically provided as key = value pairs, with ampersand (&) separators between key/ value pairs, and may be URL-encoded.

* Fragment identifier

The fragment identifier is a part of the query argument that serves like a bookmark for a specific resource. It usually begins with a number sign (#).

An absolute URL contains all information needed to access a resource and so, the scheme and domain name must always be specified. A relative URL uses an absolute URL as its starting point in accessing a resource. Because of this, the scheme, user information, and the domain name can be omitted.

**Response Header Fields**

Response header fields only appear on response messages sent by the server.

* Accept-Ranges

Accept-Ranges allows partial resources to be requested by the client. The header will have the value of ‘none’ if it does not allow clients to request partial resources. The server may opt not to send the header field at all if it is the case.

* Age

Age displays how long ago the response has been generated. The value of this header field are non-negative decimal integers and specifies time in seconds.

* ETag (entity tag)

ETag is a weak identifier and is used for conditional requests.

* Location

Location is used for redirection. If present, the server will fetch the latest resource from the location specified in this header field. For responses with 201 status code, the value of the header field would be the location where the new resource generated by the request can be found.

* Proxy-Authenticate

Proxy-Authenticate is a challenge imposed to the client to prove authorization to access the proxy.

* Retry-After

Retry-After is the amount of time the client has to wait before trying again. It can be used with a 503 (Service Unavailable) response and with any 3xx (Redirection) response, though 3xx responses may opt not to include this header field. The value can be an HTTP-date or an integer number indicating time in seconds.

* Server

Server contains information about the server. When a proxy is used in delivering a response, a Via header field must be included in the header. It is advised to configure this header field since too much information about the server makes it susceptible to attacks.

* Vary

Vary is used for content negotiation. It specifies which request header field values can the client change for the server to deliver a different kind of content.

**Entity Header Fields**

* Allow

Allow identifies methods that can be called to a specified resource. It must always be present in 405 (Method Not Allowed) response.

* Content-Encoding

Content-Encoding is used when the server changed the form of the entity. It is also used for the server to compress the resource for fast delivery.

* Content-Language

Content-Language is used for accessibility purposes. It identifies the languages the resource is available on.

* Content-Length

Content-Length specifies the size of the payload.

* Content-Location

Content-Location is used when the entity is from somewhere else or when it is from a location other than the specified URI. It shows the actual location of the resource.

* Content-MD5

Content-MD5 is a deprecated header field that can be used to check the integrity of an entity.

* Content-Range

Content-Range is used when a range request was generated. It specifies where in the full entity the partial entity should be applied.

* Content-Type

Content-Type specifies the MIME type of the entity.

* Expires

Expires is used for controlling cache access and is used along with the header field Cache-Control. Its value is an HTTP-date specifying until what time and date the resource is considered fresh.

* Last-Modified

Last-Modified is another header field, aside from Expires, that can be a possible reference value to check if the resource is fresh. It shows when the resource was last changed.

**World Wide Web (WWW)**

* Established in 1989
* Created by Tim Berners-Lee
* Worked with CERN
  + - * HTTP (Hypertext Transfer Protocol)
* Allows for the retrieval of linked resources from across the web.
  + - * HTML (Hyper Text Markup Language)
* The markup language for the web used for formatting.
  + - * URI (Uniform Resource Identifier)
* A kind of “address” that is unique and used to identify to each resource on the web. It is also commonly called a URL.
* Web server and web client
* Information system that allows documents to be connected to other documents
* A system of internet serves that support especially formatted documents
* An arrangement of web servers that boost particularly designed records
* Two most popular browsers people use is the Google Chrome and the Mozilla Firefox
* There are a few applications called Web programs that make it simple to get to the World Wide Web
* All significant Web locales have balanced their substance outline and improvement way to deal with oblige the quickly expanding division of the populace getting to the Web from little screen telephones rather than extensive screen desktop and smart phones

**HTTP Request Methods**

* Standard
* GET
* Transfer a current selected representation of the resource identified by the request URI
* Requests data from a specified resource
* Most commonly used method
* Must be supported by all compliant general-purpose servers
* Can be stored
* Can be bookmarked
* Must never be utilized when managing delicate information
* Are ought to be utilized just to recover information
* Asks for have length limitations
* Remains in the browser history
* HEAD
* Same as GET but the entity is not sent
* Used to retrieve metadata about the entity
* Must also be supported by all
* POST
* Performing resource-specific processing of entities enclosed in the message body
* Submits data to be processed to a specified resource
* Has a payload
* Are never cached
* Cannot be bookmarked
* Do not remain in the browser history
* No restrictions on the data length
* PUT
* Store the enclosed entity in the message body under a specified URI
* By default, the method is not allowed
* When allowed, authentication must be provided
* Uploads a representation of the specified URI
* DELETE
* Remove the resource associated with the specified URI
* Like PUT, the method is not allowed by default
* OPTIONS
* Request information on what can be done with the resource specified
* Returns the HTTP methods that the server supports
* TRACE
* Perform or request a loopback of the requested message (echo back)
* Typically used for testing/diagnosis of the request/response chain
* Allows the customer to perceive what is being gotten at the flip side of the demand chain and use the data for testing
* CONNECT
* Establishment of an encryption tunnel to communicate with https
* Converts the request connection to a transparent TCP/IP tunnel
* Extension Methods
* WebDAV
* PROPFIND - method recovers properties characterized on the asset recognized by the Request-URI
* PROPPATCH – method forms guidelines indicated in the demand body to set and additionally or expel properties characterized on the asset recognized by the Demand URI
* MKCOL – make collection (new folder)
* COPY – creates a duplicate of the source resource
* MOVE – does not guarantee the ability to move a resource to a particular destination
* LOCK – lock resources
* UNLOCK – unlock resources
* Safe Methods
* Doesn’t change the resources
* Method that can be cached
* GET, HEAD, OPTIONS, TRACE
* Idempotent Methods
* Repeated request result to the same response
* GET, HEAD, OPTIONS, TRACE, PUT, DELETE
* Cacheable Methods
* Requests that generate cache
* GET, HEAD are cacheable

Resources

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>

<https://www.ibm.com/support/knowledgecenter/en/SSGMGV_3.1.0/com.ibm.cics.ts31.doc/dfhtl/topics/dfhtl_uricomp.htm>

<https://docs.microsoft.com/en-us/sql/ado/guide/data/absolute-and-relative-urls>

<http://webfoundation.org/about/vision/history-of-the-web/>

<http://www.webdav.org/specs/rfc2518.html#METHOD_MOVE>