**Hypertext Transfer Protocol (HTTP): Fundamentals**

**Group 4**

**Flores, Franco**

**Javierto, Javar**

**Mangati, Steven**

**Natividad, Ernallyn**

**Pelera, Bethany**

**Tagudar, Vince**

**Web Systems and Technology**

**9461A**

**3:00-4:00 TF**

**Hypertext Transfer Protocol (HTTP): Fundamentals**

**History**

* Tim Berners-Lee invented the HTTP along with his team in CERN.
* The first version of HTTP is HTTP V0.9 (The One-Line Protocol) in 1991.
* The main motive of HTTP is to do a raw data transfer.
* On its first version it has only one method, the GET method.

**Definition**

* A protocol used by the World Wide Web (WWW), which defines how the messages transferred are formatted and transmitted. It also determines the actions that browsers and web servers take upon command.
* Hypertext Transfer Protocol or simply HTTP is an application protocol on the Internet which is a common language between users and servers which allows the sending and receiving of data.

**HTTP is a stateless Protocol**

* HTTP is called a stateless protocol because its commands execute independently, without considering the previous actions or commands that were done.
* The server and client are aware of one another during a current command then afterwards forgets about each other.

**HTTP is an** **asymmetric request-response client-serverprotocol**

* The HTTP client pulls data or information from servers instead of the server pushing data to client. A client request is needed to feed information to the client itself. Hence calling HTTP a “pull protocol”.

**HTTP is media independent**

* Any type of resource or data can be transmitted by HTTP as long as both the client/user and server know how to grasp the content.

**HTTP is connectionless**

* The client initiates a request and after the request, it disconnects from the server and waits for its response, the request is then processed and re-establishes the connection with the client for it to send a response back.

**HTTP is a Text Based**

* http is a plain text protocol that’s why it is easy to read and write
* HTTP information don’t need to translate and encode
* it uses descriptive text

**HTTP is extensible**

* this protocol is easy to extend and experiment because of HTTP Headers
* by a simple agreement between client and server about a new header’s semantics, new functionality can be introduced

**HTTP Flow**

1. Open a TCP connection

2. Send an HTTP message

3. Read and response sent by the server

4. Close or reuse the connection for the further requests.

**HTTP Request and Response**

* A client has the ability to send a message to a server that is called HTTP request
* A server has the ability to response by sending a message that is called HTTP response

**HTTP Request Message**

* It consists of:
* **[**method**][**URL**][**version**]**
* **[**headers**]**
* **[**body**]**

**HTTP** **Request Method**

* HTTP GET – it is used to retrieve data from web server by specifying parameters in URL portion of the request.
* HTTP HEAD – similar to the GET method but the server will reply with a response line and headers (no-entity body).
* HTTP POST – it is used when you want to send some data to the server.
* HTTP PUT – to request the server to store the included entity-body at a location specified by the given URL.
* HTTP DELETE – to request the server to delete a file at a location specified by the given URL.

**HTTP Request Header**

* Referer
* User-Agent
* Accept
* Accept-Language
* Cookie
* If-Modified-Since

**HTTP Response Message**

* It consists of:
* **[**version**][**status**][**reason**]**
* **[**headers**]**
* **[**body**]**

**HTTP Response Status Code**

|  |  |
| --- | --- |
| **Range** | **Category** |
| 100-199 | Informational |
| 200-299 | Successful |
| 300-399 | Redirection |
| 400-499 | Client Error |
| 500-599 | Server Error |

**HTTP Response Header**

* Connection
* Content-Encoding
* Content-Length
* Content-Type
* Date
* Expires
* Location
* Server

**URL**

* Uniform Resource Locator
* Uniquely defines sources in the web.
* Specifies addresses on the web
* Parts;

-Protocol

-Hostname

-Port

-Path and file name

-Query String

-Fragment

## HTTP Syntax

## It uses (ABNF) Augmented Backus-Naur Form.

## It uses Request Line that is in the first line of the header, and followed by optional request headers.

## Uses "<major>, <minor>" numbering scheme to indicate the protocol version.

## HTTP uses three Data/Time Format.

## It uses character set to refer to a method used with one or more tables to convert a sequence of octets into a sequence of characters.

## Content coding values indicate an encoding transformation that has been or can be applied.

## Uses Internet Media Types to provide open and extensible data typing and type negotiation.

## Product tokens are used to identify themselves by software name and version.

## It uses Quality Values that use floating points to indicate the importance of various negotiable parameters.

## Parameters/Syntax

## HTTP Version

## HTTP uses <major>.<minor> numbering scheme to mark or show the protocol version

* Uniform Resource Identifiers (URI)

This is used by HTTP to identify a resource like a website

* Date/Time Format

All of HTTP date/time stamps are represented in GMT (Greenwich Mean Time)

* Characters Sets

These are used to specify the character sets the client wants to use

* Content Encoding

These indicate that an encoding algorithm has been used to encode the content before transmitting it over the network.

* Media Types

HTTP uses Internet Media Types to provide open and extensible data typing and type negotiation.

* Language Tabs

References:

An overview of HTTP. (n.d.). Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

Beal, V. (n.d.). HTTP - HyperText Transfer Protocol. Retrieved from https://www.webopedia.com/TERM/H/HTTP.html

Cardinal, M. (2014, October 12). HTTP fundamentals for developers. Retrieved from https://www.slideshare.net/mario\_cardinal/http-fundamentals-for-developers

Grigorik, I. (2015, November 30). HTTP: Brief History of HTTP - High Performance Browser Networking (O'Reilly). Retrieved from https://hpbn.co/brief-history-of-http/

HTTP Fundamentals. (n.d.). Retrieved from ftp://cssgroup.net/Documentation/Active4D/HTTPFundamentals.pdf

HTTP (HyperText Transfer Protocol). (n.d.). Retrieved from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http\_basics.html

HTTP/1.1: Protocol Parameters. (n.d.). Retrieved from https://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.2.1

R., & F., J. (n.d.). Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing. Retrieved from https://httpwg.org/specs/rfc7230.html#notation

Tutorials Point. (2018, August 12). HTTP Overview. Retrieved from https://www.tutorialspoint.com/http/http\_overview.htm

Tutorials Point. (2018, August 12). HTTP Parameters. Retrieved from https://www.tutorialspoint.com/http/http\_parameters.htm

Tutorials Point. (2018, August 12). HTTP Methods. Retrieved from https://www.tutorialspoint.com/http/http\_methods.htm