Literature Review

**HTTP**

Hypertext Transfer Protocol is a stateless application layer protocol for communicating between distributed systems.(Podila 2013). HTTP is specified by RFC 2616 (Http 2015).

HTTP is connectionless, stateless and media independent protocol. A client sends a request and server answers with a response. After that the connection is closed and communication is forgotten on both sides. Current version of HTTP is 1.1.

HTTP is using Uniform Resource Identifiers (URI 2015) for resource requests.

**Generic URI format (HTTP Tutorial. 2015)**

URI = "http:" "//" host [ ":" port ] [ abs\_path [ "?" query ]]

Example:

https://www.youtube.com/watch?v=BNgU-ZaF06w

**HTTP request (HTTP Tutorial. 2015)**

* A Request-line
* Zero of more header fields ending with CRLF
* An empty line ending with CRLF
* A message body (optional)

A Request Line generic syntax:

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Request methods:

* GET: asking server for a resource located at given URI
* HEAD: similar to GET request method, but only header is returned from server
* POST: asking to send a data in request body to server
* PUT: asking to replace a resource located at given URI with new data in request body
* DELETE: asking to delete a resource located at given URI
* CONNECT: asking to establish a tunnel to server identified by given URI
* OPTIONS: describes the communication options for resource at given URI
* TRACE: test the resource availability

**HTTP Request Simple exampl**e:

GET <http://www.itb.ie/> HTTP/1.1

User-Agent: Fiddler

Host: <www.itb.ie>

User-Agent and Host are some of many predefined header fields. Custom fields can be introduced as well. This request message doesn’t contain any message body.

**HTTP response (HTTP Tutorial. 2015)**

* A Status-line
* Zero of more header fields ending with CRLF
* An empty line ending with CRLF
* A message body (optional)

Status Line generic syntax:

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

Status Code:

It’s 3 digit integer. First digit defines the class of the status.

* 1xx : Informational
* 2xx : Success
* 3xx : Redirection
* 4xx : Client Error
* 5xx : Server Error
* **HTTP Response exampl**e:

HTTP/1.1 200 OK

Cache-Control: private

Content-Length: 14887

Content-Type: text/html

Server: Microsoft-IIS/7.5

Set-Cookie: ASPSESSIONIDQAADTRCR=KIIFFBMDCAOAICBNIDKKACIH; path=/

X-Powered-By: ASP.NET

Date: Sun, 05 Apr 2015 09:36:25 GMT

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

Rest of the html omitted…

Example above is the response to <http://www.itb.ie> request. Response returned with status code 200 which indicates the success. After Status Line the various header fields follows. Then there is an empty line followed by message body. The message body contains actual html document of requested web page. Most of the html document is omitted in this example.

List of References

Podila. (2013). *HTTP: The Protocol Every Web Developer Must Know.* Available: http://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177. Last accessed 2nd April 2015

HTTP/1.1. (2015) *Hypertext Transfer Protocol -- HTTP/1.1*. [ONLINE] Available at:<http://www.w3.org/Protocols/rfc2616/rfc2616>. [Accessed 04 April 2015].

URI (2015). Uniform Resource Identifiers. [ONLINE] Available at: <https://www.ietf.org/rfc/rfc2396.txt>. [Accessed 05 April 2015].

HTTP Tutorial. (2015). HTTP Tutorial. [ONLINE] Available at:<http://www.tutorialspoint.com/http>. [Accessed 05 April 2015].