**HTTP and TCP, IP**

**The TCP/IP model consists of four layers:**

1. Process/Application Layer e.g. HTTP
2. Host-to-Host/Transport Layer e.g. TCP
3. Internet Layer e.g. IP
4. Network Access/Link Layer e.g. IEEE 802.2

**HTTP: Hyper Text Transfer Protocol**

HTTP is an application protocol running at the application layer. There are several protocols layered on top of each other. For example, your application will send of an Http request, or when you enter a website address. There is always a single request and a response. There are different types of http requests such as POST, PUT, GET, PATCH and DELETE.

**Below is a GET request**:

GET /about.html HTTP/1.1

Host: www.objc.io

Accept-Encoding: gzip, deflate

Connection: keep-alive

If-None-Match: "a54907f38b306fe3ae4f32c003ddd507"

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

If-Modified-Since: Mon, 10 Feb 2014 18:08:48 GMT

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_9\_2) AppleWebKit/537.74.9 (KHTML, like Gecko) Version/7.0.2 Safari/537.74.9

Referer: http://www.objc.io/

DNT: 1

Accept-Language: en-us

The first line is the request line. It has three parts: the action, the resource and the HTTP version. The next eleven lines which consist of 11 Http headers which convey different meaning to the user.

**Below is a Response:**

HTTP/1.1 304 Not Modified

Connection: keep-alive

Date: Mon, 03 Mar 2014 21:09:45 GMT

Cache-Control: max-age=3600

ETag: "a54907f38b306fe3ae4f32c003ddd507"

Last-Modified: Mon, 10 Feb 2014 18:08:48 GMT

Age: 6

X-Cache: Hit from cloudfront

Via: 1.1 eb67cb25620df959ba21a943fbc49ef6.cloudfront.net (CloudFront)

X-Amz-Cf-Id: dDSBgR86EKBemW6el-pBI9kAnuYJEaPQYEqGmBnilD12CbixCuZYVQ==

The first line is known as the status line. It contains the HTTP version followed by a status code (304) and a status message.

**TCP: Transmission Control Protocol**

The transmission control protocol is built on top of IP (internet protocol).

The Transmission Control Protocol provides reliable, ordered, error-checked delivery of a stream of data between programs. With TCP, an application running on one device can send data to an application on another device and be sure that the data arrives there in the same way that it was sent.

A TCP connection is duplex and allows data to flow in both directions. The applications on either end do not have to worry about the data being split up into packets, or the fact that the packet transport is best effort*.* A use case for TCP is when a client connects to a webserver and the client can send a message to the server and the server can send a response back. TCP uses concept of ports. A web server will listen on a port and the browser will use a so-called ephemeral port number.

**IP: Internet Protocol**

This protocol specifies how datagrams (packets) are sent between their hosts. The packet is a chunk of binary data that has a source host and a destination host. An IP network will then transmit the packet from the source to the destination. Each host in an IP network has an address, the IP address. The IP is responsible for routing datagrams as the packet travels through the network. The packet travels through the network, each node that it travels through looks at the destination address in the packet to determine where the packet should be forwarded. Most packages are IPv4, however IPv6 is slowly gaining traction.

**References to where I got all of this information**

<https://www.geeksforgeeks.org/tcp-ip-model/>

<https://www.objc.io/issues/10-syncing-data/ip-tcp-http/>