What is HTTP?

In this lesson, we will learn the basics of HTTP.

We'll cover the following

The basics of HTTP

HTTP (`Hypertext Transfer Protocol`) is a client-server protocol that enables a client and server to communicate over a network using request/response. It's an application layer protocol that relies on TCP/IP for its services.

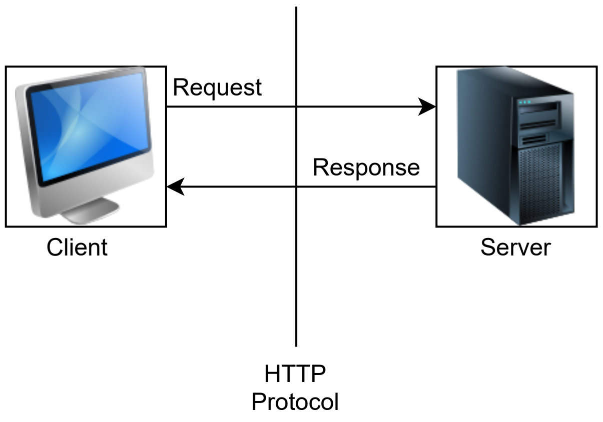
> Note: TCP (Transmission Control Protocol) is a transport layer protocol that ensures messages get delivered in the order in which they are sent, either from client to server or server to client.

What is HTTPS?

HTTPS is an extension of the `Hypertext Transfer Protocol`. It is used for secure communication over a computer network. In *HTTPS*, the communication protocol is encrypted using a TLS-encrypted TCP connection.

The architecture of HTTP

Here is the basic block diagram of a web application architecture using `HTTP`.



In the diagram above,

Client: the client sends a request to the server via the `HTTP` protocol

Server: here, the server processes the request and sends the response for the requested information to the client

The basic aspects of HTTP

- `HTTP` *is connectionless*: each client request is treated as a unique and independent connection.
- `HTTP` *is stateless but not sessionless*. By stateless we mean that, HTTP does not keep any information about the client. Therefore, there is no relation between any requests that are being carried out consecutively on the same connection.
- `HTTP` *is extensible/customized*. It can be integrated with any new functionality just by setting up or providing an agreement between a *client* and a *server*.

A sample HTTP request

Let's make a simple `HTTP` request using the curl command and analyze the output.

```
curl -iX GET https://reqres.in/api/users/1
```

Please click on the terminal to run this command and then see the response of the server.

Terminal

The server response has HTTP headers and a message body.

Response headers as key/value pairs

```
HTTP/2 200
date: Fri, 27 Mar 2020 05:04:49 GMT
content-type: application/json; charset=utf-8
content-length: 370
set-cookie: __cfduid=da1cd20e010c26e64c1cabad561df2ba21585285489;
...............................
.................
```

- The first line is the `HTTP` status code which is `200`.
- One of the **keys** is a `content-type` and has a corresponding **value** of *application/json; charset=utf-8*.
- One of the **keys** is a `set-cookie` and has a corresponding **value** of *da1cd20e010c26e64c1cabad561df2ba21585285489*.

> Note: Cookies are a unique identification number that help us keep track of our activity on different web pages. For example, Amazon keeps track of items in your cart using cookies. When you first visit the website, the server assigns the unique identification number to the client. After the cookie is set, the client will always request the server using the cookie. In this way, the server will know who is making the request so it can respond accordingly.

We will learn more about `HTTP` headers later in another section, HTTP headers.

Response message body

```
{"data":{"id":1,"email":"george.bluth@reqres.in","first_name":"George","last_n
ame":"Bluth","avatar":"https://s3.amazonaws.com/uifaces/faces/twitter/calebogd
en/128.jpg"},"ad":{"company":"StatusCode Weekly","url":"http://statuscode.or
g/","text":"A weekly newsletter focusing on software development, infrastructu
re, the server, performance, and the stack end of things."}}
```

The second part of the `HTTP` response is the *message body*, which is in the `JSON` format. It is the information returned by the server for the requested resource and in this case, we have requested the details of user-1 (`/api/users/1`).

Now that you are familiar with the HTTP, in the next lesson, we will learn about various HTTP methods.