**Note- To communicate between the clients and the server on web we need a protocol. HTTP is just one of them which has specific rules to communicate.**

**HTTP** stands for **H**yper **T**ext **T**ransfer **P**rotocol. **WWW** is about communication between web **clients** and **servers**. Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**

HTTP Request / Response

Communication between clients and servers is done by **requests** and **responses**:

1. A client (Here we are the client) sends an **HTTP request** to the web server
2. An web server receives the request(**Servers** are most often computers in the cloud)
3. The server runs an application to process the request
4. The server returns an **HTTP response** (output) to the browser
5. The client (the browser) receives the response

## The HTTP Request Circle

A typical HTTP request / response circle:

1. The browser requests an HTML page. The server returns an HTML file.
2. The browser requests a style sheet. The server returns a CSS file.
3. The browser requests an JPG image. The server returns a JPG file.
4. The browser requests JavaScript code. The server returns a JS file
5. The browser requests data. The server returns data (in XML or JSON).

//Here it is the 5TH one which we are requesting for. Since we will request for the data. Therefore it would send the data either in the xml format or json format depending on the type of data on that site

Note - Here we are going to perform HTTP request for the data. Therefore we would perform this task of fetching data from the web on a separate thread. So that our Main thread doesn’t get blocked(Otherwise too perform all the network operations on a separate thread only as we sould not load a huge task on the main Thread)

Steps to handle this ->

1. We will set up an HTTP protocol client to send HTTP requests.
2. Now the data is received through streams(we require streams to read and write data to the files from your program which read the data line by line). Whenever our program has to get data from an external source it is Input Stream. Whenever our program has to write data, it is output stream. We use streams because we can’t read all the data at once. It is because it may happen that your OS doesn’t have enough space to store the output. Therefore we do it in chunks using streams
3. We also need a buffer to store the file in a local buffer, which when is full or whole data is read, data is read from it to the main Storage.

***NOTE-***

***Generally we fetch data from the online web sites which are API’s. API’s are just a normal website which provide data to the HTTP request .***

***In the basic 2 Networking projects. We extracted the data from the net online in a very long way. It is very time consuming process to write this code. Therefore now we will use third part libriary to achieve this task.(don’t forget to add the dependency for any external third party libriary you use)***

1. ***okHTTP libriary - It is used to fetch data from the web easily i.e does the work of HTTP client***
2. ***GSON libriary – Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object.***

***Therefore first we will try to implement them. Note – generally we use retrogit libriary for all network works. But when we need to use the header in the url, we should use the okHTTP. okHTTP only can handle that HEADER***

***RETROFIT LIBRIARY***

***3rd libriary is the retrofit. We should always prefer to use this. To use this we need to add 2 libriaries one is the retrofit and the other is of gson ( gson specific to the retrofit ). It is the link to access it.***

[***https://square.github.io/retrofit/***](https://square.github.io/retrofit/)