HTTP Requests and HTTP Response

In this lesson, we will walk you through the concepts of HTTP request and response.

We'll cover the following

What is an `HTTP` *request*?

`HTTP` *requests* are the messages sent by the client to initiate an action on a resource identified by a given URL over the network. There are various `HTTP` *request* methods for specific purposes.

The components of an `HTTP` *request* are as below:

- **The method to be applied to the resource** – an `HTTP` method (like `GET`, `PUT`, `POST` `HEAD` or `OPTIONS`), that describes the action to be performed. We have already discussed the various `HTTP` methods in the previous lesson.
- **The resource identifier** – resource identified by a given request URL.
- **Headers (optional)** – headers are the information sent to the server in the form of key/value pairs, which contains the details of what the browser wants and will accept from the server.
- **Params (optional)** – used for sending additional data to the server. For example, a query parameter.
- **Message body (optional)** – These are additional information required by the server to process current requests properly. For example, a file type of JSON or XML sent in the case of a `POST` or `PUT` method request.

What is an `HTTP` *response*?

After receiving and processing an `HTTP` *request* message, a server responds with an `HTTP` *response* message.

`HTTP` request and response example

The example below will help you to understand more about `HTTP` *request* and *response*.

Request – The request below contains the `HTTP` `Method`, the target `URL`, and the query `Param` in the URL (post id = 1).

```
curl -iX GET  https://jsonplaceholder.typicode.com/posts/1
```

Terminal

Response – The *response* returned from the server for the above *Request* contains:

- A status-line (contains `HTTP` version and `HTTP` Status code) – *HTTP/2 200*
- *Headers* – these are general, response, or entity fields. We will discuss more about it in the HTTP headers lesson
- *Response message body* – a message in the form of a `JSON`, which returns the data associated with the post (id =1)


```
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio repre
henderit",
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum
\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem e
veniet architecto"
}
```


Try it yourself

You can make an HTTP request for the URLs below or any other URL/target of your choice and analyze the HTTP response in the terminal below.

```
curl -I https://www.facebook.com/
```

```
curl -I GET  https://reqres.in/api/users/2


curl -X GET  https://reqres.in/api/users/2

curl -iX GET https://reqres.in/api/unknown/2

curl -iX POST -d {"name":"john"} https://reqres.in/api/users
```

Terminal

Let’s dive right in and learn about various HTTP status codes in the next lesson.