Contents

[HTTP Request/Response Model 1](#_Toc118391140)

[HTTP Request: 2](#_Toc118391141)

[HTTP Request Methods: 2](#_Toc118391142)

[HTTP Response: 3](#_Toc118391143)

[HTTP Request/Response best practice 3](#_Toc118391144)

## HTTP Request/Response Model

The foundation of HTTP and all extended protocols built on top of it is a very basic communications model. Client and Server are two crucial terms used in HTTP communication.

Client: The person who submits the HTTP request is known as the client. The client, for instance, is the browser. There are two types of clients.

* + Browser level client: Chrome, Firefox, Edge browser etc.
  + Application-level client: Axios, Volley, Retrofit etc.

Server: The server is responsible for both receiving requests and responding to them. In essence, the server is the piece of code in charge of accepting the request and returning the response. Typically, the term "server" refers to the computer that hosts server code.



## HTTP Request:

An HTTP client sends an HTTP request to a server in the form of a request message. The first step in a web request/response communication is an HTTP request. Each request consists of a request URL, request body, and request header.

* URL: A URL is the address of a given unique resource on the Web.
* Request Body: Data sent by the client to an API is known as a request body.
* Request Header: A request header is an HTTP header that can be used in an HTTP request to provide information about the request context to the server so that the response can be tailored.

## HTTP Request Methods:

* GET (): The GET method is used to retrieve data from a server using a given URI. GET requests should only retrieve data and have no other effect on the data.
* HEAD (): Head method is used for transferring the status line and header section.
* POST (): A POST request is used to send data to the server, such as customer information, file uploads, and so on, via HTML forms.
* PUT (): Replaces the uploaded content with all current representations of the target resource.
* DELETE (): Deletes all current representations of the resource specified by a URI.

## HTTP Response:

A server responds with an HTTP response message after receiving and interpreting a request message. Every response is made up of a response header, body, and cookies.

* Status: With every response we receive a status code like 200, 404 etc.
* Response Body: Data sent by the web server is known as a response body.
* Response Header: The response header contains a number of attributes. For example, the Content-Type attribute specifies the type of response body format.

## HTTP Request/Response best practice

* Use GET () method for retrieving data: To retrieve data, GET () requests should be used. Implementing any type of write behaviour (things that will have side effects) in an HTTP GET is generally considered bad practice ().
* Use post () method for sending sensitive information: In Post () method data is not displayed in the URL. It’s more secure compared to GET () method.
* No logging of sensitive values in application: It’s a bad practice to log sensitive values in application.
* Use PUT () to modify data: PUT () request is used for modify the underlying data. It should be used for updating existing data
* Use Delete () to remove data: Delete () verb should be used to delete the existing data from the server.
* Request rate limiting: Request should be rated limited. It helps to stop attackers from slowing down the server by DoS attacking.
* HTTP Response status message: HTTP status codes are used to indicate that any specific HTTP request has successfully completed or not. HTTP response should have proper status code.