Difference between HTTP1.1 and HTTP2

HTTP (hypertext markup language) is an application protocol (a communication standard for communication with servers and the web) that was first made in 1989 by Timothy Berners-Lee. HTTP 1.1 was first created in 1997 with very few changes being made to it over the years. However, in 2015, a transformatively different version, HTTP2, was created and brought into use.

The main purpose of this massive upgrade was to reduce latency, and make it more usable with mobile platforms and server-intensive graphics and videos.

A breakdown of how these two are different is given below:

|  |  |  |
| --- | --- | --- |
| No. | HTTP1.1 | HTTP2 |
| 1. Purpose | Created by Timothy Berners-Lee as a way for devices to communicate with the internet and the server on which the desired website is stored. | Created by the Internet Engineering Task Force at Google to reduce webpage load latency using methods like compression, multiplexing, and prioritization. |
| 2. Binary Framing | Uses plain text format for all requests and responses | Uses binary framing layer to encapsulate message sin binary format while still using HTTP semantics. By converting the underlying application message to binary, it ensures that web apps created in HTTP2 function as normal even when interacting with a new protocol. |
| 3. Connect | Uses persistent connections and pipelining, The HTTP keeps the connection open unless directly told to close. The disadvantage of this is that it might result in a bottleneck (head-of-line blocking), reducing the speed and preventing data from transferring. | The binary framing layer fixes this by breaking down requests into smaller packets of info, making the data transfer more flexible. It further uses steam prioritization to avoid multiple requests wanting the same resource degrading performance. |
| 4. Flow Control | Uses TCP connections for flow control of excess incoming data. The client and server establish buffer sizes for this. With each new TCP connection, a new flow mechanism is implemented using the transport layer. | A single TCP connection is used by HTTP2 to multiplex streams of data. For flow control, it lets the client and server set their own flow controls rather than relying on the transport layer. This means the flow control mechanism doesn’t have to wait for the signal to reach its ultimate destination before adjusting the receive window. |
| 5. Compress | Uses compression programs like gzip to optimize applications. However, the header data is sent as plain text, making the data transmitted heavier with each request. | Uses HPACK compression to shrink headers as well. It uses binary framing layer to do so. It splits headers from their data, creating a separate header frame and data frame. HPACK then compresses the header frame. |

Objects and Their Internal Representation in Javascript

Javascript uses an object-based paradigm. This means it contains objects, which are identified through a collection of properties. Properties, in turn, are associations between a key and a value. If the value happens to be a function, the property is referred to as a method.

These properties are essentially variables that are attached to the object. These properties define the characteristics of the object. You can find out the properties of an object using the following dot-notation:

objectName.propertyName

*Note: The object name and the property name are case sensitive.*

For example, you can create an object called myPassport and give it properties named number, name, year.

var myPassport = new Object ();

myPassport.numer = A1234567

myPassport.name = Nandita

myPassport.year = 2022

Alternatively, you could write the object and its properties using an object initializer:

var myPassport = {

number: ‘A1234567’,

name: ‘Nandita’,

year: ‘2022’

};

Now that we understand what an object is, let’s look at the internal representation of objects.

Properties can be access using one of two expressions:

**Obj.prop** or **obj[“prop”]**

All property names are strings, even if they aren’t written as one (a number, for example), they get converted to strings. So, the object is basically mapping a string to a value. This is how they are linked internally in Javascript.