1. Write a blog on Difference between HTTP1.1 vs HTTP2

|  |  |
| --- | --- |
| HTTP1.1 | HTTP2 |
| HTTP 1.1 uses a simple request-response model, where a client sends a request to a server, and the server sends a response back to the client. The request and response messages are transmitted as plain text over the network, with the header and body separated by a blank line. | HTTP/2 (Hypertext Transfer Protocol version 2) is an updated version of the HTTP protocol, which was introduced in 2015. It was designed to address some of the limitations and performance issues of HTTP/1.1 and improve the speed and efficiency of web communication. |
| HTTP 1.1 only allows a single request and response to be sent over a single connection at a time, which can lead to performance issues when multiple requests are made simultaneously. | HTTP/2 allows multiple requests and responses to be sent over a single connection, which improves the performance of the protocol by reducing the overhead of establishing new connections. |
| HTTP 1.1 sends headers with each request and response, which can result in a lot of overhead and slow down performance, especially for large data transfers. | HTTP/2 uses a binary protocol instead of the text-based protocol used in HTTP/1.1. This reduces the size of the data transmitted between the client and server, improving the efficiency of the protocol. |
| HTTP 1.1 does not support server push, which means that the server cannot initiate the transfer of data to the client without first receiving a request. | HTTP/2 allows the server to send multiple responses to a single request, which can improve the speed and efficiency of web applications. |
| HTTP 1.1 does not provide a mechanism for prioritizing requests, which means that requests can be processed in any order, regardless of their importance or urgency. | HTTP/2 allows the client to specify the priority of requests, which can improve the responsiveness of the protocol and reduce latency. |
| HTTP/1.1 has larger header overhead. | HTTP/2 employs a more efficient header compression mechanism (HPACK) to reduce redundant header information, improving data transfer efficiency. |
| HTTP/1.1 does not have built-in support for stream prioritization. | HTTP/2 allows for the prioritization of streams, enabling more critical resources to be loaded first. |
| HTTP/1.1 may require opening multiple connections for parallel downloads. | HTTP/2 encourages the reuse of a single connection for multiple requests, reducing latency. |
| Existing websites can be served over HTTP/2 without modifications. | HTTP/2 is designed to be backward compatible with HTTP/1.1, allowing seamless migration. |
| HTTP/1.1 lacks these built-in flow control features. | HTTP/2 incorporates flow control mechanisms to prevent overwhelming the client with too much data. |

1. Write a blog about objects and its internal representation in Javascript
2. Write a blog about objects and its internal representation in Javascript

JavaScript, being a versatile and dynamic language, provides developers with powerful tools for creating and manipulating objects. Objects play a crucial role in JavaScript, serving as the building blocks for data structures, encapsulation, and functionality. In this blog post, we'll embark on a journey to explore the internal representation of objects in JavaScript, delving into the intricacies that make them so integral to the language.

Example:

var per ={name: “srisakthi”,

age:22,

batch: “B54”

Gender: “male”

}