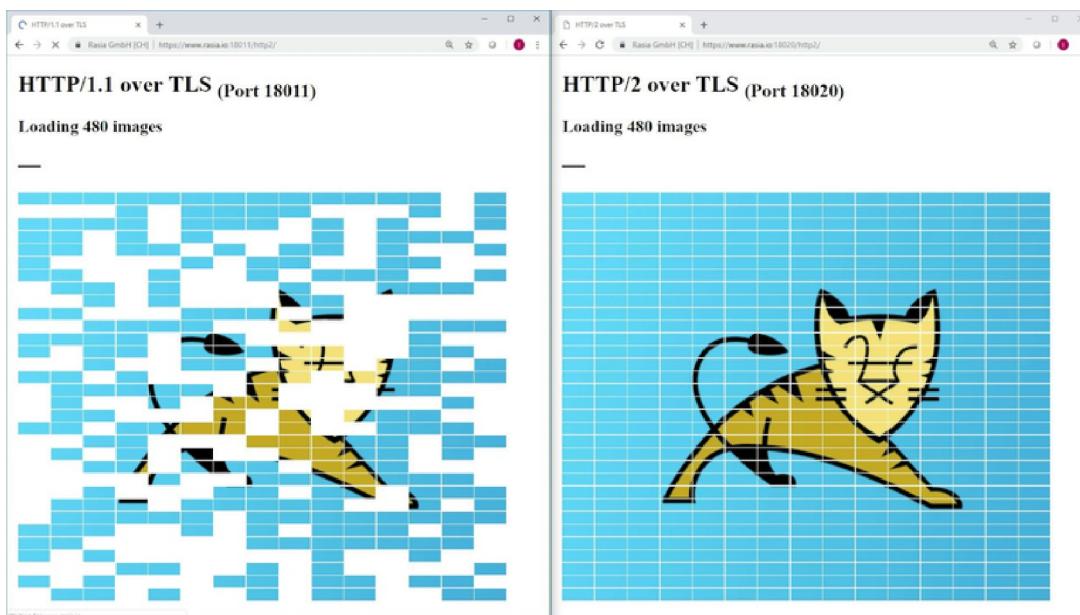


1. Write a blog on Difference between HTTP1.1 vs HTTP2 ?

Introduction :

- The evolution of the internet has led to the development of several crucial protocols, among which HTTP (Hypertext Transfer Protocol) is one of the most significant.
- HTTP defines the rules for how web browsers and servers communicate, and it has undergone several versions over the years.
- HTTP/1.1 and HTTP/2 are two major versions of this protocol, each bringing its own set of improvements.
- The key differences between HTTP/1.1 and HTTP/2 and how these changes have impacted web performance.



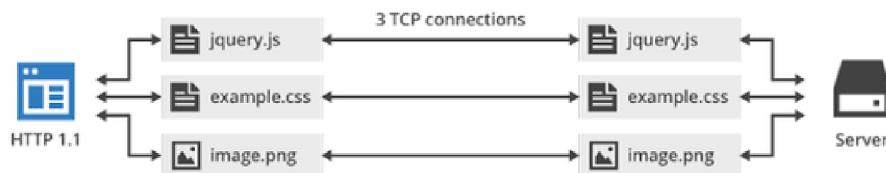
HTTP/1.1: The Old Standard :

- HTTP/1.1 has been the workhorse of the web for over a decade. It's a text-based protocol where each request/response exchange happens sequentially.

Here are some of the key characteristics of HTTP/1.1:

1. One Request at a Time:

- In HTTP/1.1, browsers are limited to making only one request at a time per connection.
- If a page has multiple resources (e.g., images, scripts, stylesheets), the browser must establish multiple connections to fetch them concurrently.
- This process can lead to inefficiencies, as browsers have to wait for responses before sending more requests.



2. Header Overhead:

- HTTP/1.1 carries a lot of redundant header information with each request and response, which increases the amount of data transferred.
- This overhead can be especially problematic for mobile devices and users with limited bandwidth.

3. No Built-in Compression:

- HTTP/1.1 doesn't natively support data compression, which means that resources like HTML, CSS, and JavaScript files are sent in their raw, uncompressed form.

HTTP/2: A Modern Approach

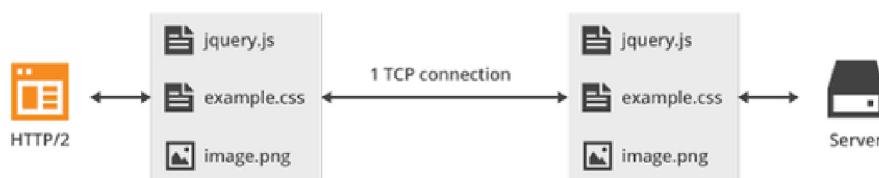
- HTTP/2, introduced in 2015, was designed to address the limitations of HTTP/1.1 and improve web performance. It brings several important changes:

Here are some of the key characteristics of HTTP/2:

1. Multiplexing:

- HTTP/2 allows multiple requests and responses to be multiplexed over a single connection.
- This means that browsers can fetch resources concurrently without the overhead of opening multiple connections.
- It significantly reduces latency and speeds up page loading.

Multiplexing



2. Header Compression:

- HTTP/2 uses header compression techniques like HPACK, which significantly reduce the size of headers sent with each request and response.
- This reduces the overall data transfer and speeds up page loading times

3. Server Push :

- One of the most exciting features of HTTP/2 is server push.
- It enables the server to proactively send resources to the client before the client requests them.
- For example, if the server knows that a client will need a specific stylesheet after receiving an HTML file, it can push that stylesheet to the client. This reduces round-trip times and improves page load times.

4 . Binary Protocol:

- Unlike HTTP/1.1's plain text format, HTTP/2 uses a binary protocol, which is more efficient to parse for both browsers and servers.

Conclusion

- HTTP/2 represents a significant improvement over HTTP/1.1 in terms of web performance, thanks to features like multiplexing, header compression, and server push.
- However, it's important to note that HTTP/1.1 is still widely used on the internet, and not all websites have migrated to HTTP/2.
- Compatibility and support for HTTP/1.1 are still essential.

2.

Write a blog about objects and its internal representation in JavaScript ?

Introduction :

- In the world of JavaScript, objects are foundational. They are versatile and powerful data structures that allow you to store and manipulate data in a structured manner.
- To work effectively with JavaScript, it's essential to understand not only how to create and use objects but also how they are internally represented.

Objects in JavaScript :

- In JavaScript, objects are collections of key-value pairs.
- They can hold various data types, including other objects and functions. Objects serve as the building blocks for constructing complex data structures and organizing code.

Internal Representation of Objects :

- JavaScript engines, responsible for executing JavaScript code, have an internal structure to represent objects efficiently. While this representation can vary between engines,

here are some common concepts to understand :

```
const person = {  
    firstName: 'John',  
    lastName: 'Doe',  
    age: 30,  
};
```

1. Properties :

- Objects consist of properties, which are stored as key-value pairs.
- In our person object, firstName, lastName, and age are properties, and their corresponding values are stored within the object.

2. Property Attributes :

- Each property in JavaScript has associated attributes that determine its behavior. These attributes include:
- **Value:** The actual data stored in the property.
- **Writable:** A Boolean indicating whether the property's value can be changed.
- **Enumerable:** A Boolean indicating whether the property is listed during iteration
- **Configurable:** A Boolean indicating whether the property can be deleted or its attributes modified.
- These attributes play a crucial role in how properties behave and can be accessed or modified.

3.

Read about IP address, port, HTTP methods, MAC address ?

IP Address (Internet Protocol Address):

- An IP address is a unique numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.
- It serves two primary functions: host or network interface identification and location addressing.
- IP addresses are divided into two major versions: IPv4 (32-bit) and IPv6 (128-bit). IPv4 addresses are more common but limited in number, while IPv6 addresses were introduced to accommodate the growing number of devices on the internet.
- IPv4 addresses are typically written in decimal format, such as "192.168.1.1," while
- IPv6 addresses are written in hexadecimal format.

Port :

- A port is a logical endpoint for network connections, used to distinguish multiple services or processes on a single device or server.
- Ports are identified by port numbers ranging from 0 to 65535.
- Ports below 1024 are considered "well-known ports" and are reserved for standard services (e.g., HTTP uses port 80, HTTPS uses port 443).
- Ports above 1024 are available for applications and services to use.

HTTP Methods (Hypertext Transfer Protocol Methods):

- HTTP methods, also known as HTTP verbs, are used to indicate the desired action to be performed on a resource identified by a URL (Uniform Resource Locator).
- Common HTTP methods include:
- GET: Retrieve data from the server.
- POST: Submit data to be processed by the server.
- PUT: Update a resource on the server.
- DELETE: Remove a resource from the server.
- HEAD: Retrieve only the headers of a response (no body).
- PATCH: Apply partial modifications to a resource.
- These methods allow clients (e.g., web browsers) and servers to interact and perform various operations on web resources.

MAC Address (Media Access Control Address) :

- A MAC address is a hardware address assigned to a network interface controller (NIC) or network adapter. It is unique for every NIC globally.
- MAC addresses are used at the data link layer of the OSI model to identify devices on a local network, such as an Ethernet LAN.
- The MAC address is typically a 48-bit (or 6-byte) address represented as a series of hexadecimal numbers, often separated by colons or hyphens (e.g., "00:1A:2B:3C:4D:5E").
- Unlike IP addresses, which can change, a device's MAC address is usually fixed and does not change unless the hardware itself is replaced.

