HTTP Push & Pull - Introduction

In this lesson, we will have an introduction to the HTTP Push & Pull mechanism.

WE'LL COVER THE FOLLOWING

In this lesson, we will get an insight into the HTTP Push & Pull mechanism. We know that the majority of the communication on the web happens over *HTTP*, especially wherever the client-server architecture is involved.

There are two modes of data transfer between the client and the server. *HTTP PUSH* & *HTTP PULL*. Let's find out what they are & what they do.

HTTP PULL

As I stated earlier, for every response, there has to be a request first. The client sends the request & the server responds with the data. This is the default mode of HTTP communication, called the HTTP PULL mechanism.

The client pulls the data from the server whenever it requires. And it keeps doing it over and over to fetch the updated data.

An important thing to note here is that every request to the server and the response to it consumes bandwidth. Every hit on the server costs the business money & adds more load on the server.

What if there is no updated data available on the server, every time the client sends a request?

The client doesn't know that, so naturally, it would keep sending the requests to the server over and over. This is not ideal & a waste of resources. Excessive pulls by the clients have the potential to bring down the server.

HTTP PUSH

To tackle this, we have the HTTP PUSH based mechanism. In this mechanism, the client sends the request for particular information to the server, just for the first time, & after that the server keeps pushing the new updates to the client whenever they are available.

The client doesn't have to worry about sending requests to the server, for data, every now & then. This saves a lot of network bandwidth & cuts down the load on the server by notches.

This is also known as a *Callback*. Client phones the server for information. The server responds, Hey!! I don't have the information right now but I'll call you back whenever it is available.

A very common example of this is user notifications. We have them in almost every web application today. We get notified whenever an event happens on the backend.

Clients use *AJAX (Asynchronous JavaScript & XML)* to send requests to the server in the HTTP Pull based mechanism.

There are multiple technologies involved in the *HTTP Push* based mechanism such as:

- *Ajax Long polling*
- *Web Sockets*
- *HTML5 Event Source*
- *Message Queues*
- *Streaming over HTTP*

We'll go over all of them in detail up-next.