Apps that Benefit from WebSockets

2 min

So far we've learned about the communication capabilities of WebSocket connections, the benefits of persistent connections, some details about how an HTTP connection can be upgraded to a WebSocket connection during the handshake process, and about secure WebSocket connections using wss://. As a final step, let's discuss the applications for which WebSockets can be most beneficial.

WebSockets are the go-to approach for applications that need real-time data. Before WebSockets, many web developers had to rely on abusing the HTTP protocol by repeatedly making HTTP

Preview: Docs Loading link description

requests

in quick succession to simulate continuous streams of data. This resulted in excessive headers, latency buildup, and difficulty tracking each client's current state.

However, WebSockets aren't always the superior solution compared to basic HTTP connections so you should consider carefully if WebSockets are needed for your application. To help, you can ask yourself these two questions:

- Does the application involve multiple users communicating with each other?
- Is the application a

Preview: Docs Loading link description

window

into server-side data that's constantly changing?

Answering "yes" to either of these questions indicates the need for WebSockets technology.

Instructions

Take a look at the four examples of applications that would benefit from WebSockets. For each, consider which of the two questions you might answer "yes" to as well as the ways in which HTTP connections would struggle to perform.

Multiplayer games

- WebSockets enable fast-paced multiplayer games where each action of a player can be instantly and continuously communicated to every other player.
- With HTTP, new actions of the other players may take place before a response reaches the client.

Financial tickers / Data trackers

- WebSockets enable the streaming of live data so that stock traders can be confident they have accurate prices before trading.
- With HTTP, you can only see the data reflected in your last HTTP request which, in a fast-paced environment, might be too late.

Social Feeds & Chat Rooms

- WebSockets enable social media users to see communicate in real-time.
- With HTTP, users would have to make requests to see new messages and comments.

Collaborative Documents

- WebSockets enable multiple users to simultaneously collaborate on online documents.
- With HTTP, contradictory changes to a document would require users to manually resolve conflicting changes.