

WebSocket Communication Patterns

2 min

The main feature of a WebSocket connection is that they allow continuous and bidirectional communication between a server and a client. In addition to the client sending data to the server, the server can also send data directly to the client – and this can happen without interruption!

For example, in the stock tracking application from the previous exercise, the server can continuously send the client the most up-to-date stock prices without the client requesting them.

In addition to enabling uninterrupted communication between a client and a server, WebSocket connections also enable a communication pattern called *broadcasting*. Broadcasting is when a server sends the same message to many connected clients at once.

In some cases, the server may send a broadcast message to all connected users but often the server will choose only a subset of clients to broadcast a message to, such as in a chatroom application. For example, when a client's message is sent to the server, the server can broadcast the message to all other connected clients. If the application supports smaller chat rooms, the server can determine which room the "sender" is in, and broadcast their message only to the other clients in the same room.

To summarize, WebSocket connections enable the three following communication patterns:

- client to server
- server to client
- server to many clients (broadcasting)

Instructions

Take a look at how a social-media news feed application might utilize each of these three communication patterns. Can you recognize which communication pattern is being used at each step?

Hint

The client can send information to the server by writing a new post or liking an existing post. The server can broadcast new posts to the other clients. When a client's own post receives a "like", the server can send that client a notification.

