

Review + Bonus Features

6 min

Nice work! In this lesson you applied your conceptual understanding of WebSockets and learned how to create a WebSocket application that utilizes popular sever- and client-side technologies.

Having completed this lesson you are now able to:

- Create a WebSocket server using the ws package
- Create a WebSocket client using the WebSocket API
- Establish and respond to WebSocket connections between your client and server
- Send messages from the client and receive them on the server
- Send messages from the server and receive them on the client
- Implement the broadcast pattern to send a single message to multiple clients
- Implement custom message types

To check your final project code, compare it to the solution code which can be downloaded by clicking the link below:

[Download the chat-solution-code](#)

Instructions

To test your understanding even further, try implementing the following bonus features for the chat application on your own:

- Allow users to specify their names, and prepend these names to messages so that all users know who is sending each message.
- When a new user joins the chat, broadcast the server time to the full chatroom.
- Instead of hard-coding your app's message types as strings, refactor them into constants that the your server- and client-side code can both reference (check out the **utils/constants.js** file).

Check out the video or download the [chat-solution-with-bonus-features zip file](#) to see how you might implement these features!

For further learning, you can also explore other WebSocket libraries such as [Socket.IO](#).