

Create a WebSocket Client

7 min

Well done! In the previous exercise, you created a WebSocket server using the ws package. In this exercise, you'll learn how to create a WebSocket client in the browser that connects to your server.

Most modern web browsers provide a native class, `WebSocket`, that provides an API for opening and closing connections to a WebSocket server, as well as the

Preview: Docs Loading link description

[methods](#)

required to send and receive data over those connections.

To create a WebSocket client, we must first call this client-side `WebSocket()` constructor function which requires one argument – the URL to which the WebSocket client should connect.

```
// index.html
const wsClient = new WebSocket('ws://some-websocket-website.com');
```

Notice that in the example above, the `ws://` protocol is used (instead of `http://`) to specify that the connection will use the WebSocket protocol.

Note: *It is easy to confuse the `WebSocket` object used in the browser with the `WebSocket` object exported by the `ws` package. Take your time to understand how each is distinct and how each object is used on the server-side or the client-side.*

Instructions

Task 1

In the WS CLIENT LOGIC section of **index.html**, we've defined a function named `init()` where you will be initializing the WebSocket client and writing the associated WebSocket logic. Find the comment labeled `// Exercise 4` indicating where you should create the WebSocket client.

Here, initialize the `wsClient` by calling the browser-native `WebSocket()` constructor to create a WebSocket client. This `wsClient` variable has already been declared for you at the top of the file so you will not need to redeclare this variable using `var`, `let`, or `const`.

Pass in the provided URL value. Since we are developing this project locally, we are using `localhost` as the domain.

Help

Your code should look like this

```
wsClient = new WebSocket(URL);
```

Task 2

The `WebSocket()` constructor requires one argument – the URL to which the `WebSocket` client should connect. We've provided this URL for you and assigned it to a variable called `URL` however the protocol is currently `?://`. Replace these question marks with the `ws://` protocol.

Then, restart your server and try sending a message. Though your messages won't be sent yet to the other connected clients, the warning message 'No WebSocket connection :(' will be gone!