

What Are WebSockets?

Intro: What Are WebSockets?

2 min

Imagine that your favorite basketball team is playing but you can't watch the game live. You're trying to follow the score on your phone but you can only get the live score by manually pressing a button to request new data. Needless to say, this isn't a great user experience. Not only is pressing a button to get the latest score annoying, each request will take some time to process meaning that the score may have already changed by the time you get a response!

This is how an application using an HTTP connection might work and demonstrates the problem that *WebSocket connections* are here to solve. While plain old HTTP connections require the client to make a new request to the server in order to get new data, WebSocket connections require just a single request that establishes continuous, bidirectional communication enabling real-time updates of data shared between a client and a server.

A WebSocket connection is the solution that allows you to track the score of a basketball game in real-time, send and receive instant messages in an online chatroom, and play fast-paced multiplayer video games in your browser.

This lesson covers the core concepts of WebSockets while the next lesson will cover building a WebSockets application. In this lesson, you will learn how WebSockets:

- Benefit specific kinds of applications
- Have various modes of communication
- Are built upon the foundation of an HTTP request
- Create a persistent connection
- Allow for bidirectional communication
- Are superior to earlier attempts to mimic real-time updates
- Are initiated with something called a "handshake"
- Can be made more secure using the wss:// protocol and HTTPS

Before taking this course, make sure you are familiar with Node.js and the basics of HTTP connections between clients and servers. If you need to review Node.js, check out the corresponding [Learn Node.js course](#).

Instructions

Take a look at these two versions of the same application. They both allow a user to track the current stock price for a company, however one requires you to manually fetch the current price by pressing a button while the other is continuously updating with the real-time price.

Which of these applications uses a WebSocket connection and which uses a traditional HTTP connection? Which of these versions offers the superior user experience?

The diagram on the left shows the HTTP version of the application. Notice how data can only come in once the button is pressed and a new data request is made.

The diagram on the right shows the WebSocket version of the application in which data is continuously streamed from the server to the client.

