

## WebSocket

Para eliminar muchos de los problemas en comunicación que se requiere que sea instantánea, la sección de Conectividad de la especificación HTML5 incluye WebSocket. WebSocket es simplemente full-duplex, bidireccional, conexión single-socket . Con WebSocket, su pedido HTTP se vuelve un pedido único para abrir una conexión WebSocket (ya sea WebSocket o WebSocket sobre TLS (Transport Layer Security, anteriormente conocido como SSL)), y reutiliza la misma conexión desde el cliente al servidor, y el servidor al cliente.

WebSocket reduce la latencia porque una vez la conexión WebSocket es establecida, el servidor puede enviar mensajes cuando se encuentren disponibles. Por ejemplo, a diferencia de polling, WebSocket hace un pedido único. El servidor no necesita esperar por un pedido del cliente. Similarmente, el cliente puede enviar mensajes al servidor en cualquier momento. Este pedido único reduce en gran medida la latencia sobre polling, el cual envía un pedido en intervalos, sin importar de si los mensajes estén disponibles.

WebSocket realiza comunicaciones en tiempo real mucho más eficientes. Usted puede siempre usar polling (y a veces incluso streaming) sobre HTTP para recibir notificaciones sobre HTTP. Sin embargo, WebSocket ahorra ancho de banda, CPU, y latencia. WebSocket es una innovación en rendimiento.

WebSocket esta siendo usado por una amplia variedad de aplicaciones. Algunas aplicaciones fueron posibles con anteriores tecnologías de comunicación “real-time” como AJAX, pero han incrementado radicalmente su rendimiento. Divisas y aplicaciones de cotización de acciones también han sido beneficiadas por la reducción de ancho de banda y conexión full-duplex que WebSocket provee. Con el incremento en aplicaciones desplegadas en el navegador, también ha habido un boom de desarrollo de juegos HTML5. WebSocket se ajusta de manera natural para utilizar juegos en la Web, como gameplay y la interacción del juego son increíblemente reliant on responsiveness. Algunos ejemplos de juegos HTML5 que usan WebSocket son populares aplicaciones online de apuestas, aplicaciones de controlador de juego que se integran con WebGL sobre WebSocket, y chat online dentro del juego. Hay también juegos masivos multijugador (MMO) que son ampliamente usados en navegadores de todos los tipos de dispositivos móviles y de escritorio.

WebSocket consiste de el protocolo de red y una API que permite establecer una conexión entre una aplicación cliente y el servidor. La API de WebSocket es una interfaz que permite a las aplicaciones utilizar el protocolo WebSocket. Mediante el uso de la API con tus aplicaciones, puede controlar un canal de comunicación full-duplex a través del cual sus aplicaciones pueden enviar y recibir mensajes.

El constructor WebSocket toma un argumento requerido, URL (la URL que quiera conectarse) y un argumento opcional, protocolos (ya sea un nombre de protocolo o un array de nombres de protocolos que el servidor debe incluir en su respuesta para establecer la conexión). Ejemplos de protocolos que puede usar como argumento son XMPP (Extensible Messaging and Presence Protocol), SOAP (Simple Object Access Protocol), o un protocolo personalizado.

```
// Creando nueva conexión WebSocket
```

```
var ws = new WebSocket("ws://www.websocket.org");
```

Cuando se conecta a el servidor WebSocket, puede opcionalmente usar el segundo argumento para listar los protocolos que la aplicación soporte, a saber por negociación de protocolo.

```
// Conectando al servidor con un protocolo llamado myProtocol
```

```
var ws = new WebSocket("ws://echo.websocket.org", "myProtocol");
```

Programación WebSocket sigue un modelo de programación asíncrona, lo cual significa que mientras una conexión WebSocket esta abierta, su aplicación simplemente escucha por eventos. El cliente no necesita activamente sondear el servidor para más información. Para empezar a escuchar por los eventos, simplemente se agrega funciones callbacks al objeto WebSocket. Alternativamente, se puede usar el método `addEventListener()` DOM para agregar event listeners a sus objetos WebSocket.