Artículo

[Mathew Lambert](#) · 18 mayo, 2020 Lectura de 2 min

Cliente para WebSockets basado en Node.js

Los WebSockets, como tecnología de comunicación, están ganando una importancia cada vez mayor. En el namespace SAMPLES puedes encontrar un buen ejemplo para ejecutar un WebSocket Server. También hay un ejemplo útil para un Browser Client en el cual JavaScript hace la mayor parte del trabajo.

La cuestión es:

¿Cómo consumir la salida de un WebSocket Server en tu aplicación?

Puedes optar por el enfoque de ingeniero y usar las herramientas y clases disponibles para gestionar todo alrededor de la protocol switching, encoding, cifrado... Está bien si puedes conseguirlo y si estás dispuesto a invertir el tiempo necesario y encuentras un patrocinador para financiarlo.

Por otro lado, es reinventar la rueda. Esto ya funciona en casi cualquier navegador.

Por ejemplo, JavaScript tiene unas librerías excelentes y comprobadas, para ayudarte en lo que necesites.

Y hay un motor para ejecutarlo sin navegador: Node.js

Y Caché, IRIS, Ensemble tienen por su parte un módulo listo, con una interfaz bien establecida.

Así que lo analicé con más detalle tras lograr juntar las versiones correctas.

Una vez ensamblado y probado el WsockDemo.js, lo ejecutas sobre CPIPE o \$ZF(...)

Provees la información de control desde el servidor y revueltes el resultado.

Si estás acostumbrado a JavaScript, no habrá grandes sorpresas. La principal diferencia es aceptar y entender que estás trabajando en un entorno casi totalmente asíncrono y los métodos callback y su interacción son probablemente la diferencia más importante con la programación tradicional.

Hay cargado un ejemplo en [Open Exchange](#) con más documentos como una invitación a salir (para la mayoría de los desarrolladores) de su entorno tradicional y descubrir algo nuevo y útil.

Esto es un primer acercamiento en Node.js y está lejos de lograr la mejor solución o la más avanzada. Pero el resultado, en relación con el esfuerzo, fue razonable.

[#Interoperabilidad](#) [#JavaScript](#) [#Node.js](#) [#SOAP](#) [#Caché](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#) [#Open Exchange](#)

URL de fuente: <https://es.community.intersystems.com/post/cliente-para-websockets-basado-en-nodejs>