

Módulo de Comunicación en Tiempo Real con WebSocket

Propósito General

Este módulo implementa un servidor WebSocket completo que proporciona comunicación bidireccional en tiempo real entre el servidor y los clientes, integrando autenticación JWT, sistema de publicación/suscripción por tópicos, y conexión directa con el servicio MQTT para datos de sensores en tiempo real. El sistema está diseñado para soportar aplicaciones que requieren actualizaciones instantáneas de datos.

Arquitectura de WebSocket Server

El servidor WebSocket se construye sobre el módulo ws y se integra con un servidor HTTP existente mediante el manejo de solicitudes de upgrade. Esta arquitectura permite que el mismo puerto maneje tanto tráfico HTTP tradicional como conexiones WebSocket, simplificando la configuración de red y firewall.

Sistema de Autenticación

El proceso de autenticación se realiza durante el handshake de upgrade HTTP a WebSocket. El sistema espera un token JWT en el parámetro de consulta token (por ejemplo: /ws?token=eyJhbGci...). Cuando se recibe una solicitud de upgrade, el servidor verifica la validez del token utilizando el servicio verifyAccessToken.

Si la autenticación falla por token ausente o inválido, el servidor responde con un código HTTP 401 Unauthorized y cierra la conexión. Si la autenticación es exitosa, el payload del token (que contiene email, rol y nombre del usuario) se adjunta al objeto WebSocket, haciendo esta información disponible para toda la duración de la conexión.

Sistema de Publicación/Suscripción (Pub/Sub)

El módulo implementa un sistema Pub/Sub completo que permite a los clientes suscribirse a tópicos específicos y recibir mensajes publicados en esos tópicos. La estructura utiliza dos mapas principales: topics que mantiene el conjunto de clientes suscritos a cada tópico, y subsByClient que realiza el seguimiento inverso de los tópicos a los que está suscrito cada cliente.

Las funciones sub, unsub y unsubAll gestionan las suscripciones de manera eficiente, utilizando WeakMap para las suscripciones por cliente, lo que permite la recolección automática de garbage collection cuando los clientes se desconectan.

Integración con Servicio MQTT

Una característica clave es la integración directa con el servicio MQTT. Cuando un cliente WebSocket se conecta, se registra automáticamente en el servicio MQTT como suscriptor para recibir datos de temperatura en tiempo real. Esto permite que los datos

provenientes de sensores MQTT se distribuyan inmediatamente a todos los clientes WebSocket conectados.

La integración es bidireccional: cuando un cliente WebSocket se desconecta, se elimina automáticamente del servicio MQTT para optimizar recursos.

Protocolo de Comunicación WebSocket

El protocolo implementado soporta varios tipos de mensajes:

Suscripción: { "type": "sub", "topic": "nombre_topico" } - Suscribe el cliente a un tópico específico

Cancelación de suscripción: { "type": "unsub", "topic": "nombre_topico" } - Elimina la suscripción a un tópico

Ping/Pong: { "type": "ping" } / { "type": "pong", "ts": timestamp } - Mecanismo de keepalive

Los mensajes publicados siguen el formato: { "topic": "nombre_topico", "ts": timestamp, "payload": datos }

Sistema de Keepalive y Monitoreo de Conexión

Para detectar y limpiar conexiones zombi, el servidor implementa un mecanismo de keepalive que envía pings cada 30 segundos a todos los clientes conectados. Cada cliente WebSocket tiene una propiedad isAlive que se establece en false al enviar un ping y se restablece a true cuando se recibe el pong correspondiente.

Los clientes que no responden a los pings dentro del intervalo establecido son terminados automáticamente, liberando recursos del servidor.

Suscripción Automática y Demo

Al conectarse, cada cliente se suscribe automáticamente al tópico "temperature" para recibir datos de temperatura en tiempo real. Adicionalmente, el módulo incluye un feed de demostración que publica valores de métricas simuladas (una función seno con ruido aleatorio) en el tópico "metricas/demo" cada segundo.

Este feed de demo es útil para desarrollo y testing, permitiendo verificar la funcionalidad WebSocket incluso cuando no hay datos MQTT reales disponibles.

Gestión de Ciclo de Vida

El servidor maneja adecuadamente el ciclo de vida de las conexiones:

Conexión: Autenticación, suscripción automática, registro en MQTT

Comunicación: Procesamiento de mensajes, keepalive, Pub/Sub

Desconexión: Limpieza de suscripciones, eliminación del registro MQTT

Consideraciones de Seguridad

El uso de autenticación JWT asegura que solo usuarios válidos puedan establecer conexiones WebSocket. Aunque el token se pasa por query string (adecuado para prototipos), en producción podría considerarse el uso de headers de autenticación más seguros.

El sistema no implementa autorización a nivel de tópicos, permitiendo a cualquier usuario autenticado suscribirse a cualquier tópico. Para aplicaciones sensibles, podría añadirse control de acceso basado en roles para diferentes tópicos.

Escalabilidad y Rendimiento

La implementación utiliza estructuras de datos eficientes (Map, Set, WeakMap) que permiten un rendimiento óptimo incluso con muchos clientes conectados. El sistema de Pub/Sub está optimizado para operaciones rápidas de suscripción y publicación.

Para entornos de alta escalabilidad, podría considerarse la implementación de clusters de WebSocket o el uso de adaptadores para distribuir el estado entre múltiples instancias.

Integración con el Ecosistema de la Aplicación

Este módulo se integra con múltiples componentes del sistema:

Servicio JWT para autenticación

Servicio MQTT para datos en tiempo real

Servidor HTTP principal para el handshake

Sistema de roles para futuras extensiones de autorización

La arquitectura modular permite fácil extensión para soportar nuevos tipos de mensajes, tópicos, o integraciones con otros servicios en tiempo real.

Este servidor WebSocket proporciona así una base sólida para aplicaciones que requieren comunicación en tiempo real, combinando seguridad, eficiencia y integración con sistemas existentes de una manera cohesiva y mantenible.