

DOCUMENTACIÓN: SERVIDOR WEBSOCKET CON AUTENTICACIÓN JWT

DESCRIPCIÓN GENERAL

Este módulo implementa un servidor WebSocket robusto y seguro que proporciona comunicación bidireccional en tiempo real para la aplicación IoT. Su función principal es permitir que los clientes reciban actualizaciones instantáneas de datos de sensores mientras mantiene control de acceso mediante autenticación JWT.

FUNCIONALIDAD PRINCIPAL

El servidor WebSocket actúa como un sistema de publicación-suscripción en tiempo real que distribuye datos de MQTT y métricas a clientes autorizados, manejando autenticación, gestión de conexiones y keep-alive automático.

SISTEMA DE AUTENTICACIÓN

HANDSHAKE CON TOKEN JWT

Implementa un proceso de upgrade HTTP a WebSocket que valida tokens JWT en la querystring antes de permitir la conexión. Este enfoque asegura que solo usuarios autenticados puedan establecer conexiones WebSocket.

VERIFICACIÓN DE CREDENCIALES

Utiliza el servicio JWT para verificar la validez y extraer el payload del token, inyectando la información del usuario (email, rol, nombre) en la conexión WebSocket para referencia posterior.

RECHAZO DE CONEXIONES NO AUTORIZADAS

Devuelve respuestas HTTP 401 apropiadas cuando los tokens faltan o son inválidos, previniendo el acceso no autorizado al canal de comunicación en tiempo real.

SISTEMA DE PUBLICACIÓN-SUSCRIPCIÓN

GESTIÓN DE TÓPICOS

Implementa un registro eficiente de suscripciones utilizando Maps y WeakMaps para tracking bidireccional entre clientes y tópicos, optimizando el rendimiento en operaciones de broadcasting.

SUSCRIPCIONES DINÁMICAS

Permite a los clientes suscribirse y desuscribirse de tópicos específicos durante la sesión, proporcionando flexibilidad para recibir solo los datos relevantes para su contexto.

BROADCASTING EFICIENTE

Distribuye mensajes a todos los clientes suscritos a un tópico específico, verificando el estado de la conexión antes de enviar para prevenir errores.

INTEGRACIÓN CON MQTT

REGISTRO AUTOMÁTICO DE CLIENTES

Registra automáticamente cada nueva conexión WebSocket en el servicio MQTT, permitiendo que los datos en tiempo real de dispositivos IoT se distribuyan inmediatamente a los clientes conectados.

SUSCRIPCIÓN POR DEFECTO

Configura automáticamente todas las nuevas conexiones para recibir datos de temperatura, asegurando que los clientes tengan acceso inmediato a las métricas principales del sistema.

GESTIÓN DE CONEXIONES

KEEP-ALIVE AUTOMÁTICO

Implementa un sistema de ping/pong que verifica periódicamente la vitalidad de las conexiones, cerrando automáticamente las conexiones zombi para liberar recursos.

LIMPIEZA AUTOMÁTICA DE RECURSOS

Remueve todas las suscripciones y registros cuando una conexión se cierra, previniendo fugas de memoria y manteniendo el sistema de suscripciones actualizado.

PROTOCOLO DE COMUNICACIÓN

MENSAJES ESTRUCTURADOS

Utiliza un formato JSON estandarizado para todos los mensajes WebSocket, incluyendo tipo, timestamp y payload para facilitar el procesamiento en el cliente.

OPERACIONES SOPORTADAS

Suscripción: Los clientes pueden suscribirse a tópicos específicos

Desuscripción: Remoción de suscripciones a tópicos

Ping/Pong: Mantenimiento de conexión y verificación de latencia

DATOS DE DEMOSTRACIÓN

GENERACIÓN DE MÉTRICAS SINTÉTICAS

Provee un feed continuo de datos de demostración con patrones de onda senoidal y variabilidad aleatoria, útil para testing y desarrollo cuando los datos reales de MQTT no están disponibles.

COMPATIBILIDAD CON SISTEMA REAL

Mantiene la generación de datos demo mientras integra datos reales de MQTT, permitiendo transición suave entre entornos de desarrollo y producción.

SEGURIDAD Y ROBUSTEZ

MANEJO DE ERRORES GRACIOSO

Captura y maneja silenciosamente mensajes malformados, previniendo que errores de cliente afecten la estabilidad del servidor.

AISLAMIENTO DE CLIENTES

Cada cliente mantiene sus propias suscripciones sin afectar a otros, proporcionando un entorno seguro y aislado para cada sesión.

ESCALABILIDAD

GESTIÓN EFICIENTE DE MEMORIA

Utiliza WeakMaps para las suscripciones por cliente, permitiendo que el garbage collector limpie automáticamente cuando las conexiones se cierran.

BROADCASTING OPTIMIZADO

El sistema de publicación evalúa el estado de cada conexión antes de enviar, evitando intentos de envío a conexiones cerradas o inestables.

INTEGRACIÓN CON ARQUITECTURA EXISTENTE

El módulo se integra perfectamente con el servidor HTTP existente, compartiendo el mismo puerto y sistema de autenticación, mientras proporciona capacidades de comunicación en tiempo real complementarias a la API RESTful.