

WsProxy

August 18, 2026

WsProxy

Kind: Class

Source: `packages/websockets/context/ws-proxy.ts`

Part of: [Websockets](#)

`WsProxy` creates an asynchronous proxy function for WebSocket-backed operations. It centralizes request forwarding and error handling so callers can invoke WebSocket actions through a Promise-based interface.

Methods

Method	Signature	Returns
<code>create</code>	<code>create(targetCallback: (...args: unknown[]) => Promise<any>, exceptionsHandler: WsExceptionsHandler, targetPattern: string)</code>	<code>(...args: unknown[]) => Promise<any></code>
<code>handleError</code>	<code>handleError(exceptionsHandler: WsExceptionsHandler, args: unknown[], error: T)</code>	<code>void</code>

When something fails

- `WsProxy` handles failure in 1 place: it logs it and continues in all 1.

Diagram

```
graph LR
  Client[Application Code] --> Proxy[Proxy[WsProxy]]
  Proxy --> Create[Create[create()]]
  Create --> Handler[Handler[Async Proxy Function]]
  Handler --> Socket[Socket[WebSocket Context]]
  Socket --> Response[Response[Promise Result]]
```

```
Socket --> Error[handleError()]
Error --> Client
```

Usage

```
import { WsProxy } from '@your-package/websockets';

// Construct the proxy with the WebSocket context/configuration required
// by your application.
const wsProxy = new WsProxy(/* websocket context */);

// Create the async function used to invoke WebSocket-backed operations.
const invoke = wsProxy.create();

try {
  const result = await invoke('users.get', { id: 'user-123' });

  console.log('WebSocket response:', result);
} catch (error) {
  console.error('WebSocket request failed:', error);
}
```

AI Coding Instructions

- Use `create()` once per configured proxy instance and retain the returned async function for WebSocket calls.
- Treat calls through the created proxy as asynchronous and always `await` them or return their Promise.
- Route transport and remote-operation failures through `handleError()` rather than duplicating WebSocket error normalization in callers.
- Ensure the WebSocket context is initialized and connected before invoking the function returned by `create()`.
- Preserve the proxy's argument forwarding behavior when extending it; proxy calls may accept arbitrary argument lists.

Relationships

- IMPORTS → `ExecutionContextHost`