WSContextInit

August 18, 2026

WSContextInit

Kind: Interface

Source: `src/helper/websocket/index.ts`

Part of: Helper

An argument for WSContext class

`WSContextInit` is the initialization contract passed to `WSContext`. It groups the underlying WebSocket transport with connection state, endpoint metadata, and no-argument `send` and `close` callbacks.

Properties

Property	Type
`raw`	`T`
`readyState`	`WSReadyState`
`url`	`string
`protocol`	`string

Diagram

```
graph LR
  Init[WSContextInit] --> Raw[raw: T]
  Init --> State[readyState: WSReadyState]
  Init --> URL[url: string | URL | null]
  Init --> Protocol[protocol: string | null]
  Init --> Send[send(): void]
  Init --> Close[close(): void]
  Init --> Context[WSContext]
```

Usage

```ts
import { WSContext } from "./helper/websocket";
import type { WSContextInit, WSReadyState } from "./helper/websocket";

const socket = new WebSocket("wss://example.test");

const init: WSContextInit<WebSocket> = {
  raw: socket,
  readyState: socket.readyState as WSReadyState,
  url: socket.url,
  protocol: socket.protocol || null,
  send: () => socket.send("ping"),
  close: () => socket.close(),
};

const context = new WSContext(init);

context.send();
context.close();
```

AI Coding Instructions

- Keep `raw` typed as the underlying WebSocket or WebSocket-compatible transport.
- Implement `send` and `close` as no-argument callbacks that match the interface signature.
- Preserve `null` for `url` and `protocol` when connection metadata is unavailable.
- Keep `readyState` aligned with the transport state when creating or updating the context.