

WsArgumentsHost

August 17, 2026

WsArgumentsHost

Kind: Interface

Source: `packages/common/interfaces/features/arguments-host.interface.ts`

Part of: [Common](#)

Methods to obtain WebSocket data and client objects.

`WsArgumentsHost` provides typed access to the arguments associated with a WebSocket handler invocation. It exposes the incoming message payload, connected client instance, and event pattern after an `ArgumentsHost` or `ExecutionContext` has been switched to the WebSocket context.

Diagram

```
graph LR
  A[WebSocket Event] --> B[ExecutionContext]
  B --> C[switchToWs()]
  C --> D[WsArgumentsHost]
  D --> E[getClient()]
  D --> F[getData()]
  D --> G[getPattern()]
```

Usage

```
import {
  CallHandler,
  ExecutionContext,
  Injectable,
  NestInterceptor,
  WsArgumentsHost,
} from '@nestjs/common';
import { Observable } from 'rxjs';
import { Socket } from 'socket.io';
```

```

@Inject()
export class WsLoggingInterceptor implements NestInterceptor {
  intercept(context: ExecutionContext, next: CallHandler): Observable<unknown> {
    const wsContext: WsArgumentsHost = context.switchToWs();

    const client = wsContext.getClient<Socket>();
    const data = wsContext.getData<{ message: string }>();
    const pattern = wsContext.getPattern();

    console.log({
      clientId: client.id,
      pattern,
      message: data.message,
    });

    return next.handle();
  }
}

```

AI Coding Instructions

- Obtain this interface through `context.switchToWs()` when working in guards, interceptors, filters, or pipes that support multiple transport types.
- Use generic type parameters with `getData<T>()` and `getClient<T>()` to avoid untyped message payloads and client objects.
- Do not assume the client is always a Socket.IO `Socket` ; its type depends on the configured WebSocket adapter.
- Use `getPattern()` when logging, authorizing, or routing behavior based on the WebSocket event name.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `ExecutionContextHost` — `packages/core/helpers/execution-context-host.ts` :10