WritePacket

August 17, 2026

WritePacket

Kind: Interface

Source: `packages/microservices/interfaces/packet.interface.ts`

Part of: Microservices

`WritePacket<T>` represents the result of writing or handling a microservice packet response. It carries either a successful typed `response` or an `err`, along with lifecycle and status information used to determine whether the underlying request has completed or been disposed.

Properties

Property	Type
`err`	`any`
`response`	`T`
`isDisposed`	`boolean`
`status`	`string`

Diagram

```
graph LR
  Request[Microservice Request] --> Handler[Packet Handler]
  Handler --> Packet[WritePacket<T>]

  Packet --> Response[response: T]
  Packet --> Error[err: any]
  Packet --> Status[status: string]
  Packet --> Disposed[isDisposed: boolean]
```

Usage

```ts
import type { WritePacket } from './interfaces/packet.interface';

interface UserProfile {
  id: string;
  name: string;
}

const packet: WritePacket<UserProfile> = {
  err: null,
  response: {
    id: 'user_123',
    name: 'Ada Lovelace',
  },
  isDisposed: false,
  status: 'success',
};

if (packet.err) {
  console.error('Microservice request failed:', packet.err);
} else if (!packet.isDisposed) {
  console.log(`Received ${packet.status} response:`, packet.response);
}
```

AI Coding Instructions

- Use `WritePacket<T>` with a concrete response type so consumers can safely access `response`.
- Check `err` before processing `response`, since error packets may not contain meaningful response data.
- Respect `isDisposed` when handling asynchronous responses; avoid writing to or consuming packets after disposal.
- Keep `status` values consistent with the surrounding microservice transport or packet-handling conventions.
- Preserve packet metadata when forwarding or transforming responses between microservice layers.