IncomingResponseDeserializer

August 19, 2026

IncomingResponseDeserializer

Kind: Class

Source: `packages/microservices/deserializers/incoming-response.deserializer.ts`

Part of: Microservices

`IncomingResponseDeserializer` normalizes responses received by microservice transports into the internal `IncomingResponse` schema. It detects whether a value is already in the expected envelope format and wraps external/raw responses when necessary, allowing downstream response handling to use a consistent structure.

Implements: `ProducerDeserializer`

Methods

Method	Signature	Returns
`deserialize`	`deserialize(value: any, options: Record<string, any>)`	`IncomingResponse`
`isExternal`	`isExternal(value: any)`	`boolean`
`mapToSchema`	`mapToSchema(value: any)`	`IncomingResponse`

Where it refuses work

- `IncomingResponseDeserializer` stops the work with an early return when `!value`.
- `IncomingResponseDeserializer` stops the work with an early return when `!isUndefined((value as IncomingResponse).err) || !isUndefined((value as IncomingResponse)⋯`.

Diagram

```
graph LR
  A[Raw transport response] --> B[deserialize]
  B --> C{isExternal?}
  C -- No --> D[Return existing IncomingResponse]
  C -- Yes --> E[mapToSchema]
  E --> F[Normalized IncomingResponse]
```

Usage

```
import { IncomingResponseDeserializer } from '@nestjs/microservices';

const deserializer = new IncomingResponseDeserializer();

const rawResponse = {
  id: 'request-42',
  data: { status: 'ok' },
};

const response = deserializer.deserialize(rawResponse);

console.log(response.id); // "request-42"
console.log(response.response); // original raw response payload
console.log(response.isDisposed); // true
```

AI Coding Instructions

- Use `deserialize()` as the public entry point; do not call `mapToSchema()` directly unless extending the deserialization behavior.
- Preserve already-normalized `IncomingResponse` objects instead of wrapping them again.
- Keep `isExternal()` aligned with the response envelope contract used by the active microservice transport.
- When changing the mapped schema, verify compatibility with downstream client response handling and request correlation via `id`.

Relationships

- IMPORTS → `isUndefined`