

IdentityDeserializer

August 18, 2026

IdentityDeserializer

Kind: Class

Source: `packages/microservices/serializers/identity.serializer.ts`

Part of: [Microservices](#)

`IdentityDeserializer` is a pass-through microservice serializer that returns incoming message data without transforming it. It is useful when a transport adapter already provides data in the format expected by the application and no custom parsing or normalization is required.

Implements: `Serializer`

Methods

Method	Signature	Returns
<code>serialize</code>	<code>serialize(value: any)</code>	<code>void</code>

Diagram

```
graph LR
  A[Microservice Transport] --> B[Incoming Message]
  B --> C[IdentityDeserializer]
  C --> D[serialize()]
  D --> E[Original Message Value]
  E --> F[Message Handler]
```

Usage

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

const serializer = new IdentityDeserializer();
```

```
const incomingMessage = {
  pattern: 'user.created',
  data: {
    id: 'user-123',
    email: 'user@example.com',
  },
};

const result = deserializer.deserialize(incomingMessage);

console.log(result === incomingMessage); // true
```

AI Coding Instructions

- Use `IdentityDeserializer` when transport payloads should be passed directly to message handlers without parsing or transformation.
- Do not add validation, serialization, or cloning logic here; use a custom deserializer when payload normalization is required.
- Ensure the configured microservice transport produces payloads compatible with the consuming handler's expected shape.
- Configure this deserializer through the microservice transport options when replacing a transport-specific or custom deserialization strategy.