

IdentitySerializer

August 17, 2026

IdentitySerializer

Kind: Class**Source:** `packages/microservices/serializers/identity.serializer.ts`**Part of:** [Microservices](#)

`IdentitySerializer` is a microservices serializer that returns values without transforming their content. It is useful when messages are already in the required transport format or when a custom serialization step is unnecessary. Its `serialize()` method preserves the original payload for downstream transport handling.

Implements: `Serializer`

Methods

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

Diagram

```
graph LR
  A[Application Payload] --> B[IdentitySerializer]
  B -->|serialize() returns unchanged value| C[Microservice Transport]
```

Usage

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

const serializer = new IdentitySerializer();

const payload = {
  event: 'user.created',
  userId: 'user_123',
```

```
};

const serializedPayload = serializer.serialize(payload);

console.log(serializedPayload);
// { event: 'user.created', userId: 'user_123' }
```

AI Coding Instructions

- Use `IdentitySerializer` when the transport or application already accepts the payload in its original form.
- Do not expect `serialize()` to convert objects into JSON, buffers, or another wire format.
- Configure this serializer in microservice client or server transport options when no custom serialization is required.
- Prefer a custom serializer when the target transport requires encoding, schema mapping, validation, or payload normalization.