

KafkaRequestSerializer

August 18, 2026

KafkaRequestSerializer

Kind: Class

Source: `packages/microservices/serializers/kafka-request.serializer.ts`

Part of: [Microservices](#)

`KafkaRequestSerializer` prepares outbound Kafka messages for transport by normalizing payloads into Kafka-compatible message objects. It JSON-encodes object and array values while preserving strings, `Buffer` instances, and `null`, and applies the same encoding rules to message keys and headers.

Implements: `Serializer`

Methods

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

Where it refuses work

- `KafkaRequestSerializer` stops the work with an early return when `isUndefined(value)`.

Diagram

```
graph LR
  A[Application payload] --> B[KafkaRequestSerializer.serialize]
  B --> C[{Kafka message shape?}]
  C -- No --> D[Wrap as { value: payload }]
  C -- Yes --> E[Use key, value, and headers]
  D --> F[encode value]
  E --> F
  E --> G[encode key]
```

```
E --> H[encode headers]
F --> I[Kafka-compatible message]
G --> I
H --> I
```

Usage

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

const serializer = new KafkaRequestSerializer();

const message = serializer.serialize({
  key: { tenantId: 'tenant-42' },
  value: {
    event: 'order.created',
    orderId: 'order-123',
  },
  headers: {
    correlationId: 'request-abc',
    metadata: { source: 'checkout' },
  },
});

console.log(message);
// {
//   key: '{"tenantId":"tenant-42"}',
//   value: '{"event":"order.created","orderId":"order-123"}',
//   headers: {
//     correlationId: 'request-abc',
//     metadata: '{"source":"checkout"}'
//   }
// }
```

AI Coding Instructions

- Pass plain payloads or Kafka message-shaped objects containing `value` , optionally with `key` and `headers` .
- Keep string values and `Buffer` values as-is; objects and arrays are automatically serialized with `JSON.stringify` .
- Ensure objects placed in `key` , `value` , or `headers` are JSON-serializable to avoid runtime serialization failures.
- Use `Buffer` explicitly when producing binary Kafka payloads rather than relying on JSON encoding.

- Preserve Kafka message fields when extending this serializer, especially `key` , `value` , and `headers` .