

KafkaRequest

August 18, 2026

KafkaRequest

Kind: Interface

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

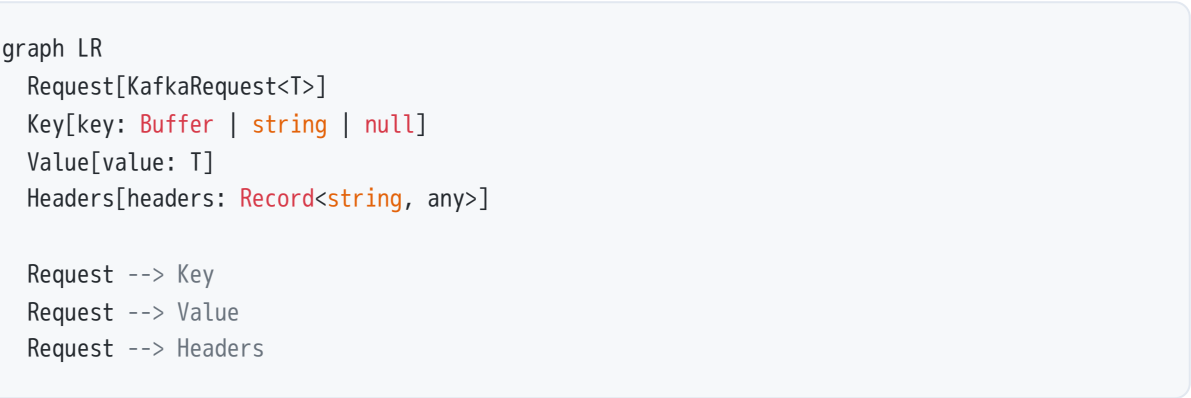
Part of: [Microservices](#)

`KafkaRequest` defines the normalized message shape consumed by the Kafka request serializer. It separates the Kafka message key, payload value, and headers so transport-specific metadata can be preserved when publishing or handling messages.

Properties

Property	Type
<code>key</code>	<code>Buffer</code>
<code>value</code>	<code>T</code>
<code>headers</code>	<code>Record<string, any></code>

Diagram



Usage

```
import { KafkaRequest } from './kafka-request.serializer';

interface UserCreatedEvent {
  userId: string;
  email: string;
}

const request: KafkaRequest<UserCreatedEvent> = {
  key: 'user-123',
  value: {
    userId: 'user-123',
    email: 'user@example.com',
  },
  headers: {
    correlationId: 'req-8f2a',
    eventType: 'user.created',
  },
};

// Pass `request` to the Kafka serializer or client publish operation.
```

AI Coding Instructions

- Use `key` for Kafka partitioning; provide a stable string or `Buffer` when related messages must remain ordered.
- Set `key` to `null` when no partitioning key is required; do not use `undefined` unless the surrounding API explicitly supports it.
- Keep `value` strongly typed with `KafkaRequest<T>` so event payload contracts are preserved.
- Store transport metadata such as correlation IDs, tracing values, and event types in `headers`.
- Ensure header values are compatible with the configured Kafka client serialization behavior before publishing.

Relationships

- IMPORTS → `isNil`
- IMPORTS → `isObject`
- IMPORTS → `isPlainObject`
- IMPORTS → `isString`
- IMPORTS → `isUndefined`

