

KafkaJSRequestTimeoutErrorMetadata

August 18, 2026

KafkaJSRequestTimeoutErrorMetadata

Kind: Interface

Source: `packages/microservices/external/kafka.interface.ts`

Part of: [Microservices](#)

`KafkaJSRequestTimeoutErrorMetadata` describes diagnostic timing and routing details for a KafkaJS request that exceeded its timeout. It captures the target broker, client identity, correlation ID, and timestamps needed to calculate how long the request remained pending.

Properties

Property	Type
<code>broker</code>	<code>string</code>
<code>clientId</code>	<code>string</code>
<code>correlationId</code>	<code>number</code>
<code>createdAt</code>	<code>number</code>
<code>sentAt</code>	<code>number</code>
<code>pendingDuration</code>	<code>number</code>

Diagram

```
graph LR
  Client[Kafka Client] -->|request| Broker[Kafka Broker]
  Client --> Metadata[KafkaJSRequestTimeoutErrorMetadata]
  Metadata --> B["B[broker: string]"]
  Metadata --> C["C[clientId: string]"]
  Metadata --> CID["CID[correlationId: number]"]
  Metadata --> CA["CA[createdAt: number]"]
  Metadata --> SA["SA[sentAt: number]"]
```

```
Metadata --> PD[pendingDuration: number]
Broker -->|timeout| Error[Request Timeout Error]
Metadata --> Error
```

Usage

```
import type { KafkaJSRequestTimeoutErrorMetadata } from './kafka.interface';

function logRequestTimeout(
  metadata: KafkaJSRequestTimeoutErrorMetadata,
): void {
  console.error('Kafka request timed out', {
    broker: metadata.broker,
    clientId: metadata.clientId,
    correlationId: metadata.correlationId,
    pendingDurationMs: metadata.pendingDuration,
    createdAt: new Date(metadata.createdAt).toISOString(),
    sentAt: new Date(metadata.sentAt).toISOString(),
  });
}

const timeoutMetadata: KafkaJSRequestTimeoutErrorMetadata = {
  broker: 'localhost:9092',
  clientId: 'orders-service',
  correlationId: 42,
  createdAt: Date.now() - 10_000,
  sentAt: Date.now() - 9_500,
  pendingDuration: 9_500,
};

logRequestTimeout(timeoutMetadata);
```

AI Coding Instructions

- Treat `createdAt`, `sentAt`, and `pendingDuration` as millisecond-based numeric timestamps/durations.
- Preserve the original `broker`, `clientId`, and `correlationId` values when wrapping or rethrowing Kafka timeout errors.
- Use this metadata for structured logging, tracing attributes, and timeout diagnostics rather than user-facing error messages.
- Do not assume `pendingDuration` is equivalent to total application processing time; it represents the Kafka request's pending duration.